

 免费电子书

学习

Web Component

Free unaffiliated eBook created from
Stack Overflow contributors.

#web-

component

.....	1
1: Web	2
.....	2
.....	2
Examples.....	2
.....	2
HTML - Hello World.....	3
- Hello World.....	3
Shadow DOM - Hello World.....	3
HTML - Hello World.....	3
Hello World.....	4
2: Web	5
.....	5
Examples.....	5
WebpackJest.....	5
.....	7

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [web-component](#)

It is an unofficial and free Web Component 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 Web Component.

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: Web

Web

Web Components Web HTML JavaScript CSS Web

Web Components

-
- HTML
- DOM
- HTML

◦

HTML	W3C HTML5	20141028
	W3C WHATWG HTML DOM	20161013
DOM	W3C WHATWG HTML DOM	20171116
HTML	W3C	2016225

Examples

<template>

-
-
-
-
-
-
- ...

`customElements.define()` Shadow DOM `attachShadow()` HTML Imports `<link rel="import">` Chrome Opera

Polyfills

polyfill

- [WebReflectionWebcomponents.org](#)
- [Shadow DOMWebcomponents.org](#)
- [Neovov](#)
- [HTMLWebcomponents.org](#)

HTML - Hello World

<template>HTML◦

```
<template id="Template1">
  Hello, World !
</template>

<div id="Target1"></div>

<script>
  Target1.appendChild( Template1.content.cloneNode( true ) )
</script>
```

#Target1 div◦

- Hello World

<hello-world>HTML“HelloWorld”

```
<script>
//define a class extending HTMLElement
class HelloWorld extends HTMLElement {
  connectedCallback () {
    this.innerHTML = 'Hello, World!'
  }
}

//register the new custom element
customElements.define( 'hello-world', HelloWorld )
</script>

<!-- make use the custom element -->
<hello-world></hello-world>
```

Shadow DOM - Hello World

DOM“HelloWorld”div◦

```
<div id="Div1">intial content</div>

<script>
  var shadow = Div1.attachShadow( { mode: 'open' } )
  shadow.innerHTML = "Hello, World!"
</script>
```

HTML - Hello World

HTMLdiv“HelloWorld”DOM◦

hello.html

```
<script>
  var div = document.createElement( 'div' )
  div.innerHTML = 'Hello, World!'
  document.body.appendChild( div )
</script>
```

index.html

```
<html>
  <link rel="import" href="hello.html">
```

Hello World

Shadow DOMHTML“HelloWorld” HTML。

hello-world.html

```
<!-- 1. Define the template -->
<template>
  Hello, World!
</template>

<script>
  var template = document.currentScript.ownerDocument.querySelector( 'template' )

  //2. Define the custom element

  customElements.define( 'hello-world', class extends HTMLElement
  {
    constructor()
    {
      //3. Create a Shadow DOM
      var sh = this.attachShadow( { mode: 'open' } )
      sh.appendChild( document.importNode( template.content, true ) )
    }
  } )
</script>
```

index.html

```
<html>
<head>
  <!-- 4. Import the HTML component -->
  <link rel="import" href="hello-world.html">
</head>
<body>
  <hello-world></hello-world>
</body>
</html>
```

Web <https://riptutorial.com/zh-CN/web-component/topic/8239/web>

2: Web

◦

Examples

WebpackJest

Facebook [Jest](#) JavaScript React ◦ Jest "" ◦

GitHub [web-components-webpack-es6-boilerplate](#)

Jest [jsdom](#) NodeJS ◦ [webpack](#)

```
-src
  --client
  --server
-webpack
  --config.js
package.json
```

server render ◦ **Webpack** config.js

```
resolve: {
  modules: ["node_modules"],
  alias: {
    client: path.join(__dirname, "../src/client"),
    server: path.join(__dirname, "../src/server")
  },
  extensions: [".js", ".json", ".scss"]
},
```

JestWebpack ◦

```
module.exports = {
  setupTestFrameworkScriptFile: "<rootDir>/bin/jest.js",
  mapCoverage: true,
  moduleFileExtensions: ["js", "scss", "html"],
  moduleDirectories: ["node_modules"],
  moduleNameMapper: {
    "src/(.*)$": "<rootDir>/src/$1"
  },
  transform: {
    "^.+\\. (js|html|scss)$": "<rootDir>/bin/preprocessor.js"
  },
  testMatch: ["<rootDir>/test/**/?(*.) (spec|test).js"],
  testPathIgnorePatterns: ["<rootDir>/ (node_modules|bin|build)"]
};
```

jest **key** package.json jest.config.js jest.config.js ◦

html◦ preprocessorjsbabel-jest◦

setupTestFrameworkScriptFilecustom elements jsdomjsdom◦ preprocessor.js

```
const babelJest = require("babel-jest");

const STYLE_URLS_REGEX = /styles:\s*\[\s*((?:'|").*\s*(?:'|").*\s*.*\s*\]/g;
const ESCAPE_TEMPLATE_REGEX = /(\${|\`)/g;

module.exports.process = (src, path, config) => {
  if (path.endsWith(".html")) {
    src = src.replace(ESCAPE_TEMPLATE_REGEX, "\\$1");
    src = "module.exports=`" + src + "`";
  }
  src = src.replace(STYLE_URLS_REGEX, "styles: []");

  return babelJest.process(src, path, config);
};
```

/require('template.html')◦ babel◦

web components polyfills◦ jsdom◦ setupTestFrameworkScriptFile jest.js

```
require("document-register-element/pony")(window);
```

jsdomweb components **API**◦

```
-bin
  --jest.js
  --preprocessor.js
-src
  --client
  --server
-webpack
  --config.js
-test
package.json
jest.config.js
```

testyarn run jest --no-cache --config \$(node jest.config.js) yarn run jest --no-cache --config \$(node jest.config.js)◦

Web <https://riptutorial.com/zh-CN/web-component/topic/10057/web>

S. No		Contributors
1	Web	Community , Mike , Supersharp
2	Web	Vardius