



Бесплатная электронная книга

УЧУСЬ

underscore.js

Free unaffiliated eBook created from
Stack Overflow contributors.

#underscor

e.js

.....	1
1: underscore.js	2
.....	2
Examples.....	2
.....	2
2:	4
Examples.....	4
.....	4
.....	5
.....	6

Около

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

It is an unofficial and free underscore.js 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 underscore.js.

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

глава 1: Начало работы с underscore.js

замечания

В этом разделе представлен обзор того, что такое underscore.js, и почему разработчик может захотеть его использовать.

Следует также упомянуть о любых крупных предметах в underscore.js и ссылаться на связанные темы. Поскольку Documentation for underscore.js является новым, вам может потребоваться создать начальные версии этих связанных тем.

Examples

Установка или настройка

Underscore - это библиотека функций функционального программирования с открытым исходным кодом для JavaScript. Underscore предоставляет множество полезных функций для работы с массивами или наборами объектов JavaScript, включая фильтрацию, сортировку и запрос.

Node.js

Убедитесь, что у вас установлен узел и npm, затем введите следующую команду

```
npm install underscore
```

Беседка

Убедитесь, что у вас установлен узел, npm и bower, затем введите следующую команду

```
bower install underscore
```

NuGet

```
Install-Package underscore.js
```

браузер

Загрузите [версию разработки](#) или [версию Production](#) и свяжите скрипт внутри HTML-файла с помощью элемента `script`.

```
<script src="js/underscore/underscore.js"></script>
```

Прочитайте [Начало работы с underscore.js](https://riptutorial.com/ru/underscore-) онлайн: <https://riptutorial.com/ru/underscore->

[js/topic/7688/начало-работы-с-underscore-js](#)

глава 2: Коллекции

Examples

карта

Функция `.map` принимает массив и функцию `iteratee`, итератор производит преобразованную копию каждого объекта массива.

Функция `iteratee` предоставляет 3 аргумента

1. `item` - текущий итерированный объект
2. `i` - индекс итерированного объекта
3. `list` - ссылка на исходный массив / список

Новый массив будет иметь такую же длину, как старый массив, но будет удерживать преобразованные объекты

Пример:

```
_.map([1, 2, 3, 4], function(item, i, list) {  
  return (item*item);  
});  
// [1, 4, 9, 16]
```

Более краткий способ написать приведенный выше пример с использованием ES6 будет

```
_.map([1, 2, 3, 4], (item, i, list) => {  
  return (item*item);  
});
```

или с использованием встроенного лямбда-выражения

```
_.map([1, 2, 3, 4], (item, i, list) => (item*item));
```

Карта также полезна, когда вы хотите выщипывать свойства объектов и создавать массив из них

Пример:

```
let people = [{name: 'he-man', age: 22}, {name: 'man-at-arms', age: 44}];  
  
_.map(people, function(item) {  
  return item.name;  
});  
// ['he-man', 'man-at-arms']
```

каждый

Функция `_.each` принимает массив или объект, функцию `iteratee` и необязательный объект `context`, функция `iteratee` вызывается один раз и для каждого элемента массива. Функция `iteratee` предоставляет 3 аргумента

```
item - The current iterated object (or value if an object was passed)
i - The index of the iterated object (or key if an object was passed)
list - A reference to the original array/list (the object that was passed)
```

Пример 1:

```
_.each(["hello", "underscore"], function(item, i, list) {
  alert(item)
});
```

В приведенном выше примере будут показаны 2 предупреждения, первое со словами «привет» и второе для «мира».

Пример 2:

```
_.each({one: 1, two: 2, three: 3}, (value, key, object) =>
  console.log(JSON.stringify(object)));
);
```

Этот пример будет записывать стробированную версию объекта 3 раза.

`.each` - это операция терминала, и в отличие от других промежуточных функций (`map`, `pluck`, `values` и т. д.) вам не нужно возвращаться внутри тела функции `iteratee`.

Прочитайте Коллекции онлайн: <https://riptutorial.com/ru/underscore-js/topic/7695/коллекции>

кредиты

S. No	Главы	Contributors
1	Начало работы с underscore.js	Ashiquzzaman , Community , svarog
2	Коллекции	svarog