



Kostenloses eBook

LERNEN

underscore.js

Free unaffiliated eBook created from
Stack Overflow contributors.

#underscor

e.js

Inhaltsverzeichnis

Über	1
Kapitel 1: Erste Schritte mit underscore.js	2
Bemerkungen.....	2
Examples.....	2
Installation oder Setup.....	2
Kapitel 2: Sammlungen	4
Examples.....	4
Karte.....	4
jeder.....	4
Credits	6



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

Kapitel 1: Erste Schritte mit underscore.js

Bemerkungen

In diesem Abschnitt erhalten Sie einen Überblick darüber, was underscore.js ist und warum ein Entwickler sie verwenden möchte.

Es sollte auch alle großen Themen in underscore.js erwähnen und auf die verwandten Themen verweisen. Da die Dokumentation für underscore.js neu ist, müssen Sie möglicherweise erste Versionen dieser verwandten Themen erstellen.

Examples

Installation oder Setup

Underscore ist eine Open Source-Bibliothek für funktionales Programmierprogramm für JavaScript. Unterstrich bietet viele nützliche Funktionen zum Arbeiten mit Arrays oder Sammlungen von JavaScript-Objekten, einschließlich Filtern, Sortieren und Abfragen.

Node.js

Stellen Sie sicher, dass Sie node und npm installiert haben, und geben Sie den folgenden Befehl ein

```
npm install underscore
```

Laube

Stellen Sie sicher, dass Knoten, Npm und Bower installiert sind, und geben Sie den folgenden Befehl ein

```
bower install underscore
```

NuGet

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

Browser

Laden Sie die [Entwicklungsversion](#) oder die [Produktionsversion](#) herunter, und verknüpfen Sie das Skript in Ihrer HTML-Datei mithilfe des `script` .

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

Erste Schritte mit underscore.js online lesen: <https://riptutorial.com/de/underscore->

Kapitel 2: Sammlungen

Examples

Karte

Die `.map` Funktion akzeptiert ein Array und eine Iterate-Funktion. Der Iteratee erstellt eine transformierte Kopie jedes Array-Objekts.

Die Iterate-Funktion stellt 3 Argumente bereit

1. `item` - Das aktuelle iterierte Objekt
2. `i` - Der Index des iterierten Objekts
3. `list` - Ein Verweis auf das ursprüngliche Array / die Liste

Das neue Array hat die gleiche Länge wie das alte Array, enthält jedoch die transformierten Objekte

Beispiel:

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

Eine prägnantere Möglichkeit, das obige Beispiel mit ES6 zu schreiben, wäre

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

oder mit einem Inline-Lambda-Ausdruck

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

Map ist auch nützlich, wenn Sie Eigenschaften aus Objekten ziehen und ein Array daraus erstellen möchten

Beispiel:

```
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']
```

jeder

Die `_.each` Funktion akzeptiert einen Array oder ein Objekt, eine iteratee Funktion und einen optionalen `context` wird die Funktion die iteratee iteratee Funktion liefert 3 Argumente für jedes Array - Element einmal und in Reihenfolge aufgerufen

```
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)
```

Beispiel 1:

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

Das obige Beispiel zeigt 2 Warnungen, die erste mit den Worten "Hallo" und die Zweite für "Welt".

Beispiel 2

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

In diesem Beispiel wird dreimal eine stringifizierte Version des Objekts protokolliert.

`_.each` ist eine Terminaloperation, und im Gegensatz zu anderen Zwischenfunktionen (`map`, `pucken`, `Werte` usw.) müssen Sie nicht in den iteratee-Funktionskörper zurückkehren.

Sammlungen online lesen: <https://riptutorial.com/de/underscore-js/topic/7695/sammlungen>

Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit underscore.js	Ashiquzzaman , Community , svarog
2	Sammlungen	svarog