



EBook Gratuito

APPENDIMENTO

underscore.js

Free unaffiliated eBook created from
Stack Overflow contributors.

#underscor

e.js

Sommario

Di.....	1
Capitolo 1: Iniziare con underscore.js.....	2
Osservazioni.....	2
Examples.....	2
Installazione o configurazione.....	2
Capitolo 2: collezioni.....	3
Examples.....	3
carta geografica.....	3
ogni.....	3
Titoli di coda.....	5

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

Capitolo 1: Iniziare con underscore.js

Osservazioni

Questa sezione fornisce una panoramica di cosa è underscore.js e perché uno sviluppatore potrebbe volerlo utilizzare.

Dovrebbe anche menzionare qualsiasi argomento di grandi dimensioni all'interno di underscore.js e collegarsi agli argomenti correlati. Poiché la Documentazione per underscore.js è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

Examples

Installazione o configurazione

Underscore è una libreria di utilità di programmazione funzionale open source per JavaScript. Underscore offre molte funzioni utili per lavorare con array o raccolte di oggetti JavaScript, inclusi filtri, ordinamenti e query.

Node.js

Assicurati di aver installato node e npm, quindi digita il seguente comando

```
npm install underscore
```

pergolato

Assicurati di aver installato node, npm e bower, quindi digita il seguente comando

```
bower install underscore
```

NuGet

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

Browser

Scarica la [versione di sviluppo](#) o la [versione di produzione](#) e collega lo script all'interno del file HTML utilizzando l'elemento di `script`.

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

Leggi [Iniziare con underscore.js online](https://riptutorial.com/it/underscore-js/topic/7688/iniziare-con-underscore-js): <https://riptutorial.com/it/underscore-js/topic/7688/iniziare-con-underscore-js>

Capitolo 2: collezioni

Examples

carta geografica

La funzione `.map` accetta un array e una funzione iteratee, l'iteratee produce una copia trasformata di ogni oggetto array.

La funzione iteratee fornisce 3 argomenti

1. `item` - L'oggetto iterato corrente
2. `i` - L'indice dell'oggetto iterato
3. `list` - Un riferimento alla matrice / lista originale

Il nuovo array avrà la stessa lunghezza del vecchio array ma manterrà gli oggetti trasformati

Esempio:

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

Sarebbe un modo più conciso di scrivere l'esempio sopra usando ES6

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

o usando un'espressione lambda in linea

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

La mappa è anche utile quando vuoi cogliere le proprietà dagli oggetti e realizzarne una serie

Esempio:

```
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']
```

ogni

La funzione `_.each` accetta un array o un oggetto, una funzione iteratee e un oggetto `context`

opzionale, la funzione iteratee viene richiamata una volta e in ordine per ogni elemento dell'array
La funzione iteratee fornisce 3 argomenti

```
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)
```

Esempio 1:

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

L'esempio sopra mostrerà 2 avvisi, il primo con le parole "ciao" e il secondo per "mondo".

Esempio 2:

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

Questo esempio registrerà una versione con stringa dell'oggetto 3 volte.

`.each` è un'operazione terminale e, a differenza di altre funzioni intermedie (map, pluck, valori ecc.) non è necessario tornare all'interno del corpo della funzione iteratee.

Leggi collezioni online: <https://riptutorial.com/it/underscore-js/topic/7695/collezioni>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con underscore.js	Ashiquzzaman , Community , svarog
2	collezioni	svarog