Telerik Academy

Dive into Underscore.js

Dive into the query-based JavaScript

Telerik Software Academy
Learning & Development Team
http://academy.telerik.com

Underscore.js

- Underscore.js is a JavaScript library, that extends regular JavaScript functionality
  - Provides extensions to object, arrays, selection, etc..
  - Usable is client JavaScript (web and mobile) and server JavaScript (Node.js)

Telerik Academy

Underscore.js Functionality

- Underscore extends the functionality for:
  - Collections
    - each, map, find, filter, where, some, countBy
  - Arrays
    - first, last, union, uniq, range
  - Functions
    - Bind, delay, defer, once, after
  - Objects
    - Keys, values, extend, functions, clone
- Templates and Chaining

Telerik Academy

Extensions for Collections

Collections

- Collections means both arrays and objects
  - All underscore methods work both on arrays and objects (associative arrays)
- Collection extensions:
  - _.each() - iterates over a collection
  - _.map(), _.pluck() - a projection of a collection
  - _.filter(), _.reject() and _.where - filter elements
  - _.all() and _.any() - evaluate a collection
  - _.sortBy(), _.groupBy() - sorts and groups

Telerik Academy

Collections: each()

- _.each() iterates over a list of elements, yielding each in turn to an iterator function
  - Just like for-in
    - Delegates to the native forEach function if supported

```
var numbers = [1, 2, 3, 4, 5, 6, 7, 8];
_.each(numbers, console.log);
_.each(numbers, function(item) { console.log(item); }
//log all the numbers
```

- Can be used with objects as well:

```
_.each(console, console.log);
//logs all members of the console
var person = new Person("Doncho","Minkov");
_.each(person, console.log);
//logs Doncho and Minkov
```

_.each()

Live Demo

Telerik Academy

Collections: map()

- _.map() produces a new array of elements, after the values are computed
  - Delegates to the native map, if supported

```
var numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8];
var numberNames = ["zero", "one", "two", "three", …]
function numbersToNames(item) { return numberNames[item];}
_.map(numbers, numbersToNames);e
//log the number names
```

  - Can be used with objects as well:

```
_.map(console, function (item) {
                return item.toString();
               });
```

_.map()

Live Demo

Telerik Academy

Collections: filter() and reject()

- Filter and reject return a subset of the original collection, based on an boolean expression
  - Filter returns all items matching the condition
  - Reject returns all items that don't fulfill the condition

```
var numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8];
function isEven(number){ return number%2 === 0;}
var even = _.filter(numbers, isEven);
var odd = _.reject(numbers, isEven);
```

Telerik Academy

_.filter() and _.reject()

Live Demo

Telerik Academy

Collections: where()

- _.where() filters a collection based on a property value
  - Returns a subcollection of the original collection

```
var people = [{name: "Ivan Todorov", age: 21},
              {name: "Todor Ivanov", age: 11},
              {name: "Petra Georgieva", age: 14},
              {name: "Georgi Petrov", age: 11},
              {name: "Stamina Staminova", age: 19}];
var elevenYearsOld = _.where(people, {age: 11});
//returns Todor Ivanov and Georgi Petrov
```

_.where()

Live Demo

Telerik Academy

Collections: all() and any()

- _.all() returns true if ALL of the elements that meet a boolean expression
- _.any() returns true if ANY of the elements fulfill a boolean condition
  - And false if none if the elements fulfull the condition

```
var numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
var anyEven = _.any(numbers,
                    function(el){ return el%2==0;});
//anyEven = true
var allEven = _.all(numbers,
                    function(el){ return el%2==0;});
//allEven = false;
```

_.all() and _.any()

Live Demo

Telerik Academy

Collections: pluck()

- _.pluck() returns a projection of a collection
  - Select all elements, but only part of them
  - A simplified version of _.map()

```
var people = […];
Var lnames = _.pluck(people, 'lname');
//lnames = ['Minkov', 'Kenov', 'Georgiev',
            'Kostov', 'Stoyanov' ]
```

_.pluck()

Live Demo

Telerik Academy

Collections: sortBy() and groupBy()

- _.sortBy() sorts the elements of a collection
  - Much like the native sort function
  - Sort by a property
  - Sort by iterator function

```
var people = […];
people = _.sortBy(people, 'fnames');
```

- _.groupBy() groups the elements of a collection
  - Group by a property
  - Group by with iterator function

```
var people = […];
people = _.groupBy(people, 'age');
```

Telerik Academy

_.sortBy() and _.groupBy()

Live Demo

Telerik Academy

Array Extensions

Telerik Academy

Array Extensions

- Array extensions work only on array objects
  - Does not work on associative arrays or objects
- Array extensions:
  - _.first() and _.initial() selects the first n items
  - _.last() and _.rest() selects the last n items
  - _.compact() - removes all false values
  - _.union() and _.intersect() - unites or intersects two or more arrays

Telerik Academy

Arrays: first() and initial()

- _.first() returns the first element in an array
  - Can be used with a parameter to return the first N elements

```
var numbers = [1, 2, 3, 4, 5];
var first = _.first(numbers); //1
Var firstTwo = _.first(numbers, 2); //[1, 2]
```

- _.initial() returns all elements except the last one
  - Can be used with a parameter to all the elements except the last N

```
var numbers = [1, 2, 3, 4, 5];
var initial = _.initial(numbers); //[1, 2, 3, 4]
var initialTwo = _.initial(numbers, 2); // [1, 2, 3]
```

Telerik Academy

_.first() and _. initial()

Live Demo

Telerik Academy

Arrays: last() and rest()

- _.last() returns the last element in an array
  - Can be used with a parameter to return the last N elements

```
var numbers = [1, 2, 3, 4, 5];
var first = _.last(numbers); //5
Var firstTwo = _.last(numbers, 2); //[4, 5]
```

- _.rest() returns all elements except the first one
  - Can be used with a parameter to all the elements except the first N

```
var numbers = [1, 2, 3, 4, 5];
var initial = _.rest(numbers); //[2, 3, 4, 5]
var initialTwo = _.rest(numbers, 2); //[3, 4, 5]
```

Telerik Academy

last() and rest()

Live Demo

Telerik Academy

Function Extensions

Telerik Academy

Function Extensions

- Function extensions provide some additional functionality to regular functions
- Function extensions:
  - _.memorize(func)
    - Memorizes the invocation of a function
    - If the function is called again with the same parameters, the memorized result is returned
  - _.compose(func1, func2, func3)
    - Returns a composition list of function
    - The same as func1(func2(func3 ()));

Telerik Academy

Function Extensions

Live Demo

Telerik Academy

Object Extensions

Telerik Academy

Object Extensions

- Object extensions provide some additional functionality to regular objects
- Object extensions:
  - _.keys(obj) – list of all the keys of an object
  - _.values(obj) – list of the values of an object
  - _.invert(obj) – inverts the keys and the values
  - _.extend(obj, properties) – performs prototypal inheritance

Telerik Academy

Object Extensions

Live Demo

Telerik Academy

Underscore.js

http://academy.telerik.com

Homework

1. Write a method that from a given array of students finds all students whose first name is before its last name alphabetically. Print the students in descending order by full name. Use Underscore.js
2. Write function that finds the first name and last name of all students with age between 18 and 24. Use Underscore.js
3. Write a function that by a given array of students finds the student with highest marks
4. Write a function that by a given array of animals, groups them by species and sorts them by number of legs

Telerik Academy

Homework (2)

5. By a given array of animals, find the total number of legs
   - Each animal can have 2, 4, 6, 8 or 100 legs
6. By a given collection of books, find the most popular author (the author with the highest number of books)
7. By an array of people find the most common first and last name. Use underscore.