



Hands-on JavaScript Iterators

Purpose of the this hands-on training is to teach the students JavaScript Iterators.

Outline

- Part 1 - What is Iterators?
- Part 2 - JavaScript Array `forEach()` Method
- Part 3 - JavaScript Array `map()` Method
- Part 4 - JavaScript Array `filter()` Method
- Part 5 - Part 5 - JavaScript Array `reduce()` Method
- Part 6 - JavaScript Array `every()` Method
- Part 7 - JavaScript Array `some()` Method
- Part 8 - JavaScript Array `find()` Method

Part 1 - What is Iterators?

- ES6 (aka ES2015) has introduced new concept in javascript: iterators.
- The Iterator pattern allows clients to effectively loop over a collection of objects.

Part 2 - JavaScript Array `forEach()` Method

- The `forEach()` method executes a provided function once for each array element.

Syntax :

```
nameOfArray.forEach(callback(element, index, arr), thisValue);
```

Part 3 - JavaScript Array `map()` Method

- The `map()` method creates a new array populated with the results of calling a provided function on every element in the calling array.

Syntax :

```
nameOfArray.map(function(currentValue, index, arr), thisValue)
```

Part 4 - JavaScript Array `filter()` Method

- The `filter()` method creates a new array with elements that fall under a given criteria from an existing array;

```
nameOfArray.filter(callback(element, index, arr), thisValue);
```

Part 5 - JavaScript Array `reduce()` Method

- **Reduce** is a method that can be difficult to understand especially with all the vague explanations that can be found on the web. There are a lot of benefits to understanding reduce as it is often used in state management (think **Redux**).

```
nameOfArray.reduce(callback, initialValue);
```

Part 6 - JavaScript Array `every()` Method

- The JavaScript array `every()` method checks whether all the given elements in an array are satisfying the provided condition. It returns true when each given array element satisfying the condition otherwise false.

Syntax :

```
array.every(callback(currentvalue, index, arr), thisArg);
```

Part 7 - JavaScript Array `some()` Method

- The `some()` method tests whether at least one element in the array passes the test implemented by the provided function. It returns a Boolean value.

```
array.some(callback[, thisObject]);
```

Part 8 - JavaScript Array `find()` Method

- The JavaScript `find()` method is a convenient way to find and return the first occurrence of an element in an array, under a defined testing function.

```
array.find(function(currentValue, index, arr),thisValue)
```