Array

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array

Array Methods

- push
- pop
- shift
- unshift
- slice
- splice
- sort
- reverse
- map
- filter
- some
- every
- ..

sort

```
var fruit = ['cherries', 'apples', 'bananas'];
fruit.sort();

// fruits -> ['apples', 'bananas', 'cherries']

var scores = [1, 10, 2, 21];
scores.sort();

// scores -> [1, 10, 2, 21]
```

comparator

```
function compare(a, b) {
  if (a is less than b by some ordering criterion) {
    return -1;
  }
  if (a is greater than b by the ordering criterion) {
    return 1;
  }
  // a must be equal to b
  return 0;
}

[...].sort(compare);
```

Sorting helper

https://www.npmjs.com/package/sort-by

Included in exercises bundle

slice (copying)

```
var items = [1, 2, 3];

var itemsCopy = items.slice();

// items -> [1, 2, 3];

// itemsCopy -> [1, 2, 3];

itemsCopy[0] = 100;

// itemsCopy -> [100, 2, 3];

// items -> [1, 2, 3];
```

gotcha with slice

And Arrays in general..

```
var items = [{ x: 1 }, { x: 2 }, { x: 3 }];
var itemsCopy = items.slice();


// items -> [{ x: 1 }, { x: 2 }, { x: 3 }];
// itemsCopy -> [{ x: 1 }, { x: 2 }, { x: 3 }];

itemsCopy[0].x = 100;

// itemsCopy -> [{ x: 100 }, { x: 2 }, { x: 3 }];
// items -> [{ x: 100 }, { x: 2 }, { x: 3 }];
```

forEach

- Executes a function once for each element in an array
- Returns nothing

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/forEach

forEach

```
var numbers = [1, 2, 3];

var doubled = [];

numbers.forEach(function(item) {
  doubled.push(item * 2);
});

// doubled -> [2, 4, 6]

// numbers -> [1, 2, 3];



[...].forEach(function(currentItem, currentIndex, theWholeArray) {
  // perform side-effect
});
```

forEach

```
var fruits = ['Banana', 'Apple', 'Mango'];

fruits.forEach(function(fruit) {
  console.log(fruit);
});

// Banana

// Apple

// Mango


// fruits -> ['Banana', 'Apple', 'Mango'];

[...].forEach(function(currentItem, currentIndex, theWholeArray) {
  // perform side-effect
});
```

map

- Transform each element in the array
- Returns a new array with the transformed elements

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/map

map

```
var numbers = [1, 2, 3];

var doubled = numbers.map(function(num) {
  return num * 2;
});


// numbers -> [1, 2, 3];

// doubled -> [2, 4, 6];


[...].map(function(currentItem, currentIndex, theWholeArray) {
  // return the new transformed value
});
```

map

```
var numbers = [1, 2, 3];

var double = function (num) {
  return num * 2;
}


var doubled = numbers.map(double);


// numbers -> [1, 2, 3];

// doubled -> [2, 4, 6];
```

Much less messy loops and temporary variables!

map

```
var fruits = ['Banana', 'Apple', 'Mango'];

var upperCased = fruits.map(function(fruit) {
  return fruit.toUpperCase();
});


// fruits -> ['Banana', 'Apple', 'Mango'];

// upperCased -> ['BANANA', 'APPLE', 'MANGO'];


[...].map(function(currentItem, currentIndex, theWholeArray) {
  // return the new transformed value
});
```

filter

- Run a condition on each element
- Return a filtered array where we drop the ones that returned false

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter

filter

```
var numbers = [1, 2, 3, 4];

var odds = numbers.filter(function(num) {
  return (num % 2) !== 0; // true or false
});


var evens = numbers.filter(function(num) {
  return (num % 2) === 0; // true or false
});


// odds -> [1, 3]

// evens -> [2, 4]

// numbers -> [1, 2, 3, 4]
```

Array Method Exercises

Canvas: exercises-array-methods.zip