MISSION READY

DARE TO
DEVELOP

Array and (more) String Methods

Ewan Zhang

Arrays Recap

- An array is a data structure which can hold more than one value at a time in an ordered fashion, like a list of items
- Array used to store multiple values in a single variable.

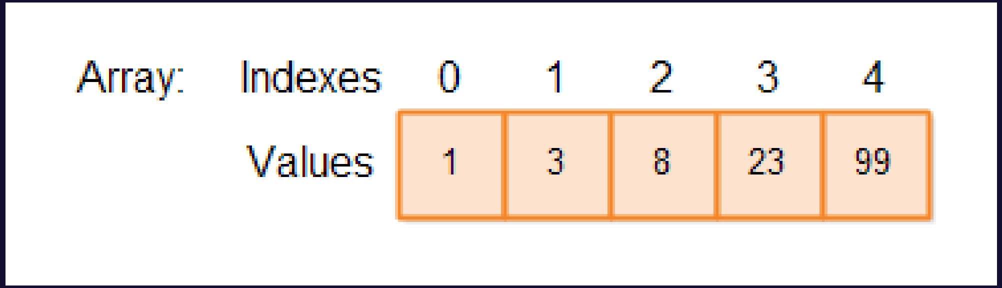
```
const arrayName = [item1, item2, ...];

const fruits = ["apples", "oranges", "peaches", "grapes", "lemons", "mango", 
"banana"];
```

Access the Elements (values) of an Array

- JavaScript arrays are zero-based indexed.
  - The first element of an array starts at index 0, the second element starts at index 1, and so on.

To access an element in an array, you specify an index in the square brackets [ ]

```
arrayName[index]
```



Array:	Indexes	0	1	2	3	4
	Values	1	3	8	23	99

Array properties

- The length property of an array returns the length of an array (the number of array elements).

```
const names = ["Grace", "Ollie", "Lee"];
names.length; // the length of names is 3
```

Array Methods

```
let pushMethod = iceCreams.push("rainbow");
//adds the item to the end, returns new length.
//pushMethod === 4;
//["vanilla", "chocolate", "blueberry", "rainbow"]

let popMethod = iceCreams.pop();
//removes and returns last item.
//popMethod === "rainbow".
// ["vanilla", "chocolate", "blueberry"]

let unshiftMethod = iceCreams.unshift("tiramisu");
//adds the item at the beginning, returns new length.
//unshiftMethod === 4;
//["tiramisu", "vanilla", "chocolate", "blueberry"]

let shiftMethod = iceCreams.shift();
//removes and returns first item.
//shiftMethod === "tiramisu".
//["vanilla", "chocolate", "blueberry"]
```

Some more methods

- String.split()
  - Takes a pattern and divides a String into an ordered list of substrings by searching for the pattern, puts these substrings into an array, and returns the array
  - Limit number of elements to return

```
const icecreamSplit = "vanilla chocolate matcha";
const resultsIcecreamSplit = icecreamSplit.split(" ", 2);
console.log(resultsIcecreamSplit); // vanilla chocolate
```

- Array.join()
  - Returns a new string by concatenating all of the elements in the array (or an array-like object), separated by commas or a specified separator string.

```
const icecreamsJoin = ["vanilla", "chocolate", "matcha"];
const resultsIcecreamsJoin = icecreamsJoin.join(" ");
console.log(resultsIcecreamsJoin); // vanilla chocolate
matcha
```

Array.slice()

slice() can also be used on strings!

- Returns a copy of a portion of an array into a new array selected from **start** to **end** (***end** not included*) where **start** and **end** represent the index of items in that array. The original array will not be modified.

```
const numberSlice = [0, 1, 2, 3, 4, 5];
const resultsNumberSlice = numberSlice.slice(1);//slice(1,3)

console.log(resultsNumberSlice); //return a new sliced array
console.log(numberSlice);//not affect origin array
```

```
//exercise:
const array = ["html","css","JS","react"]
//How can I get ["JS"] by slice() method?
```

Array.splice()

- Used to modify an array by adding, removing, or replacing elements at specific positions.
- It allows you to make changes to the array in place, rather than creating a new array.
- Return an array of delete elements
- Modifies the original array directly. Use [...arr] for keep the original value in array

```javascript
const numberSplice = [0, 1, 2, 3, 4, 5];
const resultsNumberSplice = numberSplice.splice(1,1,"pen");

console.log(resultsNumberSplice); //return a new array with delete element
console.log(numberSplice); //Modifies the original array directly.
```

```javascript
//splice() exercsise:
const array1 = ["html","css","react"]
//How can I put "JS" back in array1 after "css" by slice() method?
```

Array.forEach()

- apply a function to all elements in an array
- Need to pass in a function
- The function takes a value argument
- Optionally the function takes index and array as arguments

```javascript
const icecreamsForEach = ["vanilla", "chocolate", "blueberry"];

icecreamsForEach.forEach(function (icecream, index) {
  console.log(icecream + " ice-cream");
});

// "vanilla ice-cream"  "chocolate ice-cream"  "blueberry ice-cream"
```

Anonymous function

- This is we normally do:

```
icecreamsForEach.forEach(addString);

function addString(icecream, index) {
  console.log(icecream + " ice-cream");
}
```

- Often when you need to pass a function as a parameter of another function, you can do it without set up function names.

```
icecreamsForEach.forEach(function (icecream, index) {
  console.log(icecream + " ice-cream");
});
```

Anonymous arrow function

- Function without names (from previous page):

```
icecreamsForEach.forEach(function (icecream, index) {
  console.log(icecream + " ice-cream");
});
```

- Can be written in arrow function syntax:

```
icecreamsForEach.forEach((icecream, index)=> console.log(icecream + " ice-
cream"));
```

- Can be written in an even more compact syntax (if only one param):

```
icecreamsForEach.forEach(icecream => console.log(icecream + " ice-cream"));
```

Array.map() ★ ★ ★ ★ ★

- apply a function to all non-null elements in an array
- Returns a new array as a result. Does not change original array
- Like forEach(), you need to pass in function

```
const icecreamsMap = ["vanilla", "chocolate", "blueberry"];
const orderedIcecreams = icecreamsMap.map(function (icecream, index) {
  return `Flavour ${index}: ${icecream}`;
});

console.log(orderedIcecreams);
 // ['Flavour 0: vanilla', 'Flavour 1: chocolate', 'Flavour 2: blueberry']
```

```
//map exercise
const array3 = [1,2,3,4,5]
/*Use map() method to get this array [2,4,6,8,10]
```

Array.sort()

- Used to sort the elements of an array in place.
- It rearranges the elements of the array in ascending order by default
- The sorting is performed based on the values returned by the comparison function.
- Modifies the original array directly. Use [...arr] for keep the original value in array

```
const numbers = [5, 3, 8, 1, 2];
const a = numbers.sort();

console.log(a); // Output: [1, 2, 3, 5, 8]
console.log(numbers)// Output: [1, 2, 3, 5, 8]
```

```
//Sort() exercise:
const icecreams1 = ["vanilla", "rainbow", "chocolate", "matcha"];
//Using sort() method to get this array below:
//[ 'chocolate', 'matcha', 'rainbow', 'vanilla' ]
```

More Array Methods

- Array.filter()
  - Returns a new array with all elements in the original array that passed a test
  - Do not change the original array
  - Optionally the function takes index and array as arguments

```
const icecreamCosts = [2.95, 4, 5];
const expensiveIcecreamCosts =
icecreamCosts.filter(function (cost, index, array) {
  return cost > 3;
});
console.log(expensiveIcecreamCosts); //results == [4, 5]
```

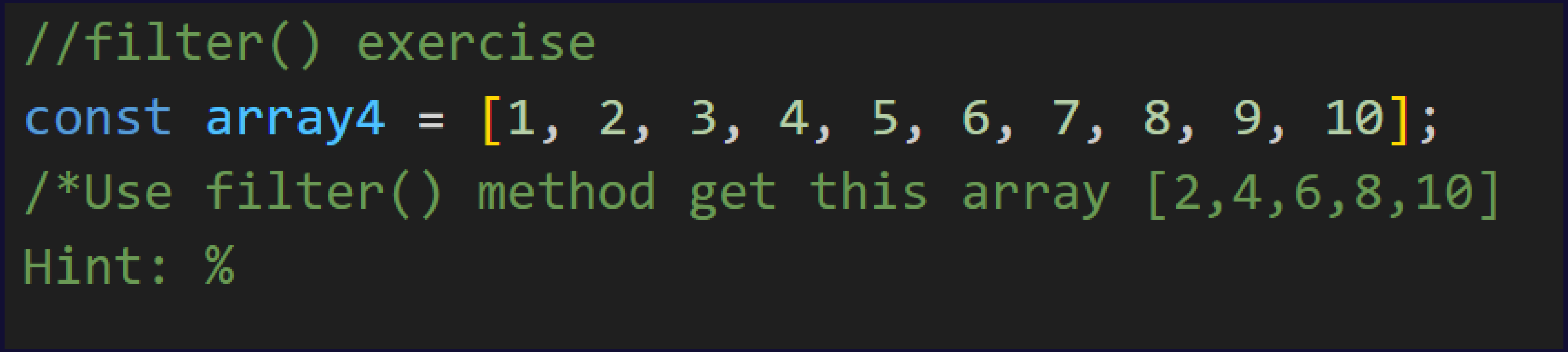
- Array.indexOf()
  - Returns a number, which is the position of the FIRST element in an array that passed a test
  - Index starts from 0
  - Optionally the function takes index and array as arguments

```
const icecreamsIndexOf = ["vanilla", "chocolate",
"matcha"];
let chocIndex = icecreamsIndexOf.indexOf("chocolate");
console.log(chocIndex); // 1
```

Array.filter()

- Returns a new array with all elements in the original array that passed a test
- Do not change the original array
- Optionally the function takes index and array as arguments

```
const icecreamCosts = [2.95, 4, 5];
const expensiveIcecreamCosts = icecreamCosts.filter(function (cost, index, array) {
  return cost > 3;
});
console.log(expensiveIcecreamCosts); //results == [4, 5]
```



```
//filter() exercise
const array4 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
/*Use filter() method get this array [2,4,6,8,10]
Hint: %
```

Array.includes()

- Returns a Boolean (true/false) stating whether all elements in the array passed a test

```
const arrayOfThings = ["books", "pens", "paper", "pencils", "words"];

const guess = arrayOfThings.includes("shoe")

console.log(guess);//false
```

- Exercise:
  Create a prompt and let user to guess the element in the array above;
  Show the result as true or false in the console

More Array Methods

- Array.every()
  - Returns a Boolean (true/false) stating whether all elements in the array passed a test
  - Optionally the function takes index and array as arguments
- Array.some()
  - Returns a Boolean (true/false) stating whether at least 1 element in the array passed a test
  - Optionally the function takes index and array as arguments

```
const ages1 = [19, 35, 24, 55];
const everyAgeOver30 = ages1.every(function (age, index,
array) {
  return age > 30;
});
console.log(everyAgeOver30); // false
```

```
const ages2 = [19, 35, 24, 55];
const someAgesOver30 = ages2.some(function (age, index,
array) {
  return age > 30;
});
console.log(someAgeOver30); // true
```

Exercise: Title Capitaliser

Convert the case of the first letter of every word in a string to uppercase. Every other letter of each word should be converted to lowercase.

Suggested methods to include:

String methods:

- String.toLowerCase()
- String.toUpperCase()
- String.split()
- String.slice()

Array methods:

- Array.map()
- Array.join()

Example:
"good MORNING to you" becomes "Good Morning To You"

Find out more about these Array Methods

- concat()
- reverse()
- reduce ()
- …
- https://www.w3schools.com/js/js_array_methods.asp

Array method review

- https://www.w3schools.com/js/js_array_methods.asp

MISSION READY
www.missionreadyhq.com

DARE TO
DEVELOP

Thank you | Ewan Zhang