30 Days Of JavaScript: Arrays

LinkedIn Follow @asabeneh

Author: Asabeneh Yetayeh

January, 2020

[<< Day 4](#) | [Day 6 >>](#)

Day 5

📔 Day 5

Arrays

In contrast to variables, an array can store *multiple values*. Each value in an array has an *index*, and each index has *a reference in a memory address*. Each value can be accessed by using their *indexes*. The index of an array starts from *zero*, and the index of the last element is less by one from the length of the array.

An array is a collection of different data types which are ordered and changeable(modifiable). An array allows storing duplicate elements and different data types. An array can be empty, or it may have different data type values.

How to create an empty array

In JavaScript, we can create an array in different ways. Let us see different ways to create an array. It is very common to use *const* instead of *let* to declare an array variable. If you ar using const it means you do not use that variable name again.

- Using Array constructor

```
// syntax
const arr = Array()
// or
// let arr = new Array()
console.log(arr) // []
```

- Using square brackets([])

```
// syntax
// This the most recommended way to create an empty list
const arr = []
console.log(arr)
```

How to create an array with values

Array with initial values. We use *length* property to find the length of an array.

```
const numbers = [0, 3.14, 9.81, 37, 98.6, 100] // array of numbers
const fruits = ['banana', 'orange', 'mango', 'lemon'] // array of strings, fruits
const vegetables = ['Tomato', 'Potato', 'Cabbage', 'Onion', 'Carrot'] // array of strings, veg
const animalProducts = ['milk', 'meat', 'butter', 'yoghurt'] // array of strings, products
const webTechs = ['HTML', 'CSS', 'JS', 'React', 'Redux', 'Node', 'MongDB'] // array of web tec
const countries = ['Finland', 'Denmark', 'Sweden', 'Norway', 'Iceland'] // array of strings, c

// Print the array and its length

console.log('Numbers:', numbers)
console.log('Number of numbers:', numbers.length)

console.log('Fruits:', fruits)
console.log('Number of fruits:', fruits.length)

console.log('Vegetables:', vegetables)
console.log('Number of vegetables:', vegetables.length)

console.log('Animal products:', animalProducts)
console.log('Number of animal products:', animalProducts.length)

console.log('Web technologies:', webTechs)
console.log('Number of web technologies:', webTechs.length)

console.log('Countries:', countries)
console.log('Number of countries:', countries.length)
```

```
Numbers: [0, 3.14, 9.81, 37, 98.6, 100]
Number of numbers: 6
Fruits: ['banana', 'orange', 'mango', 'lemon']
Number of fruits: 4
Vegetables: ['Tomato', 'Potato', 'Cabbage', 'Onion', 'Carrot']
Number of vegetables: 5
Animal products: ['milk', 'meat', 'butter', 'yoghurt']
Number of animal products: 4
Web technologies: ['HTML', 'CSS', 'JS', 'React', 'Redux', 'Node', 'MongDB']
Number of web technologies: 7
Countries: ['Finland', 'Estonia', 'Denmark', 'Sweden', 'Norway']
Number of countries: 5
```

- Array can have items of different data types

```
const arr = [
    'Asabeneh',
    250,
    true,
    { country: 'Finland', city: 'Helsinki' },
    { skills: ['HTML', 'CSS', 'JS', 'React', 'Python'] }
] // arr containing different data types
console.log(arr)
```

Creating an array using split

As we have seen in the earlier section, we can split a string at different positions, and we can change to an array. Let us see the examples below.

```
let js = 'JavaScript'
const charsInJavaScript = js.split('')

console.log(charsInJavaScript) // ["J", "a", "v", "a", "S", "c", "r", "i", "p", "t"]

let companiesString = 'Facebook, Google, Microsoft, Apple, IBM, Oracle, Amazon'
const companies = companiesString.split(',')

console.log(companies) // ["Facebook", " Google", " Microsoft", " Apple", " IBM", " Oracle", "
let txt =
  'I love teaching and empowering people. I teach HTML, CSS, JS, React, Python.'
const words = txt.split(' ')

console.log(words)
// the text has special characters think how you can just get only the words
// ["I", "love", "teaching", "and", "empowering", "people.", "I", "teach", "HTML,", "CSS,", "J
```

Accessing array items using index

We access each element in an array using their index. An array index starts from 0. The picture below clearly shows the index of each element in the array.

arr index

```
const fruits = ['banana', 'orange', 'mango', 'lemon']
let firstFruit = fruits[0] // we are accessing the first item using its index

console.log(firstFruit) // banana

secondFruit = fruits[1]
console.log(secondFruit) // orange

let lastFruit = fruits[3]
console.log(lastFruit) // lemon
// Last index can be calculated as follows

let lastIndex = fruits.length - 1
lastFruit = fruits[lastIndex]

console.log(lastFruit)  // lemon
```

```
const numbers = [0, 3.14, 9.81, 37, 98.6, 100]  // set of numbers

console.log(numbers.length)  // => to know the size of the array, which is 6
console.log(numbers)         // -> [0, 3.14, 9.81, 37, 98.6, 100]
console.log(numbers[0])      //  -> 0
console.log(numbers[5])      //  -> 100

let lastIndex = numbers.length - 1;
console.log(numbers[lastIndex]) // -> 100


const webTechs = [
  'HTML',
  'CSS',
  'JavaScript',
  'React',
  'Redux',
  'Node',
  'MongoDB'
] // List of web technologies

console.log(webTechs)        // all the array items
console.log(webTechs.length) // => to know the size of the array, which is 7
console.log(webTechs[0])     //  -> HTML
console.log(webTechs[6])     //  -> MongoDB

let lastIndex = webTechs.length - 1
console.log(webTechs[lastIndex]) // -> MongoDB


const countries = [
  'Albania',
  'Bolivia',
  'Canada',
  'Denmark',
  'Ethiopia',
  'Finland',
  'Germany',
  'Hungary',
  'Ireland',
  'Japan',
  'Kenya'
] // List of countries

console.log(countries)      // -> all countries in array
console.log(countries[0])   //  -> Albania
console.log(countries[10])  //  -> Kenya

let lastIndex = countries.length - 1;
console.log(countries[lastIndex]) //  -> Kenya
```

```
const shoppingCart = [
  'Milk',
  'Mango',
  'Tomato',
  'Potato',
  'Avocado',
  'Meat',
  'Eggs',
  'Sugar'
] // List of food products

console.log(shoppingCart) // -> all shoppingCart in array
console.log(shoppingCart[0]) //  -> Milk
console.log(shoppingCart[7]) //  -> Sugar

let lastIndex = shoppingCart.length - 1;
console.log(shoppingCart[lastIndex]) //  -> Sugar
```

Modifying array element

An array is mutable(modifiable). Once an array is created, we can modify the contents of the array elements.

```
const numbers = [1, 2, 3, 4, 5]
numbers[0] = 10      // changing 1 at index 0 to 10
numbers[1] = 20      // changing  2 at index 1 to 20

console.log(numbers) // [10, 20, 3, 4, 5]

const countries = [
  'Albania',
  'Bolivia',
  'Canada',
  'Denmark',
  'Ethiopia',
  'Finland',
  'Germany',
  'Hungary',
  'Ireland',
  'Japan',
  'Kenya'
]

countries[0] = 'Afghanistan'  // Replacing Albania by Afghanistan
let lastIndex = countries.length - 1
countries[lastIndex] = 'Korea' // Replacing Kenya by Korea

console.log(countries)
```

```
["Afghanistan", "Bolivia", "Canada", "Denmark", "Ethiopia", "Finland", "Germany", "Hungary", "
```

Methods to manipulate array

There are different methods to manipulate an array. These are some of the available methods to deal with arrays:*Array, length, concat, indexOf, slice, splice, join, toString, includes, lastIndexOf, isArray, fill, push, pop, shift, unshift*

Array Constructor

Array:To create an array.

```
const arr = Array() // creates an an empty array
console.log(arr)

const eightEmptyValues = Array(8) // it creates eight empty values
console.log(eightEmptyValues) // [empty x 8]
```

Creating static values with fill

fill: Fill all the array elements with a static value

```
const arr = Array() // creates an an empty array
console.log(arr)

const eightXvalues = Array(8).fill('X') // it creates eight element values filled with 'X'
console.log(eightXvalues) // ['X', 'X','X','X','X','X','X','X']

const eight0values = Array(8).fill(0) // it creates eight element values filled with '0'
console.log(eight0values) // [0, 0, 0, 0, 0, 0, 0, 0]

const four4values = Array(4).fill(4) // it creates 4 element values filled with '4'
console.log(four4values) // [4, 4, 4, 4]
```

Concatenating array using concat

concat:To concatenate two arrays.

```
const firstList = [1, 2, 3]
const secondList = [4, 5, 6]
const thirdList = firstList.concat(secondList)

console.log(thirdList) // [1, 2, 3, 4, 5, 6]
```

```
const fruits = ['banana', 'orange', 'mango', 'lemon']                 // array of fruits
const vegetables = ['Tomato', 'Potato', 'Cabbage', 'Onion', 'Carrot'] // array of vegetables
const fruitsAndVegetables = fruits.concat(vegetables)                 // concatenate the two a

console.log(fruitsAndVegetables)
```

```
["banana", "orange", "mango", "lemon", "Tomato", "Potato", "Cabbage", "Onion", "Carrot"]
```

Getting array length

Length:To know the size of the array

```
const numbers = [1, 2, 3, 4, 5]
console.log(numbers.length) // -> 5 is the size of the array
```

Getting index an element in arr array

indexOf:To check if an item exist in an array. If it exists it returns the index else it returns -1.

```
const numbers = [1, 2, 3, 4, 5]

console.log(numbers.indexOf(5)) // -> 4
console.log(numbers.indexOf(0)) // -> -1
console.log(numbers.indexOf(1)) // -> 0
console.log(numbers.indexOf(6)) // -> -1
```

Check an element if it exist in an array.

- Check items in a list

```
// let us check if a banana exist in the array

const fruits = ['banana', 'orange', 'mango', 'lemon']
let index = fruits.indexOf('banana')  // 0

if(index === -1){
   console.log('This fruit does not exist in the array')
} else {
    console.log('This fruit does exist in the array')
}
// This fruit does exist in the array

// we can use also ternary here
index === -1 ? console.log('This fruit does not exist in the array'): console.log('This fruit
// let us check if an avocado exist in the array
let indexOfAvocado = fruits.indexOf('avocado')  // -1, if the element not found index is -1
if(indexOfAvocado === -1){
   console.log('This fruit does not exist in the array')
} else {
    console.log('This fruit does exist in the array')
}
// This fruit does not exist in the array
```

Getting last index of an element in array

lastIndexOf: It gives the position of the last item in the array. If it exist, it returns the index else it returns -1.

```
const numbers = [1, 2, 3, 4, 5, 3, 1, 2]

console.log(numbers.lastIndexOf(2)) // 7
console.log(numbers.lastIndexOf(0)) // -1
console.log(numbers.lastIndexOf(1)) //  6
console.log(numbers.lastIndexOf(4)) //  3
console.log(numbers.lastIndexOf(6)) // -1
```

includes:To check if an item exist in an array. If it exist it returns the true else it returns false.

```
const numbers = [1, 2, 3, 4, 5]

console.log(numbers.includes(5)) // true
console.log(numbers.includes(0)) // false
console.log(numbers.includes(1)) // true
console.log(numbers.includes(6)) // false

const webTechs = [
  'HTML',
  'CSS',
  'JavaScript',
  'React',
  'Redux',
  'Node',
  'MongoDB'
] // List of web technologies

console.log(webTechs.includes('Node'))  // true
console.log(webTechs.includes('C'))     // false
```

Checking array

Array.isArray:To check if the data type is an array

```
const numbers = [1, 2, 3, 4, 5]
console.log(Array.isArray(numbers)) // true

const number = 100
console.log(Array.isArray(number)) // false
```

Converting array to string

toString:Converts array to string

```
const numbers = [1, 2, 3, 4, 5]
console.log(numbers.toString()) // 1,2,3,4,5

const names = ['Asabeneh', 'Mathias', 'Elias', 'Brook']
console.log(names.toString()) // Asabeneh,Mathias,Elias,Brook
```

Joining array elements

join: It is used to join the elements of the array, the argument we passed in the join method will be joined in the array and return as a string. By default, it joins with a comma, but we can pass different string parameter which can be joined between the items.

```
const numbers = [1, 2, 3, 4, 5]
console.log(numbers.join()) // 1,2,3,4,5

const names = ['Asabeneh', 'Mathias', 'Elias', 'Brook']

console.log(names.join()) // Asabeneh,Mathias,Elias,Brook
console.log(names.join('')) //AsabenehMathiasEliasBrook
console.log(names.join(' ')) //Asabeneh Mathias Elias Brook
console.log(names.join(', ')) //Asabeneh, Mathias, Elias, Brook
console.log(names.join(' # ')) //Asabeneh # Mathias # Elias # Brook

const webTechs = [
  'HTML',
  'CSS',
  'JavaScript',
  'React',
  'Redux',
  'Node',
  'MongoDB'
] // List of web technologies

console.log(webTechs.join())       // "HTML,CSS,JavaScript,React,Redux,Node,MongoDB"
console.log(webTechs.join(' # '))  // "HTML # CSS # JavaScript # React # Redux # Node # MongoD
```

Slice array elements

Slice: To cut out a multiple items in range. It takes two parameters:starting and ending position. It doesn't include the ending position.

```
const numbers = [1,2,3,4,5]

console.log(numbers.slice()) // -> it copies all  item
console.log(numbers.slice(0)) // -> it copies all  item
console.log(numbers.slice(0, numbers.length)) // it copies all  item
console.log(numbers.slice(1,4)) // -> [2,3,4] // it doesn't include the ending position
```

Splice method in array

Splice: It takes three parameters:Starting position, number of times to be removed and number of items to be added.

```
const numbers = [1, 2, 3, 4, 5]
numbers.splice()
console.log(numbers)                // -> remove all items
```

```
const numbers = [1, 2, 3, 4, 5]
  numbers.splice(0,1)
console.log(numbers)            // remove the first item
```

```
const numbers = [1, 2, 3, 4, 5, 6]
  numbers.splice(3, 3, 7, 8, 9)
console.log(numbers.splice(3, 3, 7, 8, 9))  // -> [1, 2, 3, 7, 8, 9] //it removes three item
```

Adding item to an array using push

Push: adding item in the end. To add item to the end of an existing array we use the push method.

```
// syntax
const arr  = ['item1', 'item2','item3']
arr.push('new item')
console.log(arr)
// ['item1', 'item2','item3','new item']
```

```
const numbers = [1, 2, 3, 4, 5]
numbers.push(6)
console.log(numbers) // -> [1,2,3,4,5,6]
```

```
numbers.pop() // -> remove one item from the end
console.log(numbers) // -> [1,2,3,4,5]
```

```
let fruits = ['banana', 'orange', 'mango', 'lemon']
fruits.push('apple')
console.log(fruits)    // ['banana', 'orange', 'mango', 'lemon', 'apple']

fruits.push('lime')
console.log(fruits)   // ['banana', 'orange', 'mango', 'lemon', 'apple', 'lime']
```

Removing the end element using pop

pop: Removing item in the end.

```
const numbers = [1, 2, 3, 4, 5]
numbers.pop() // -> remove one item from the end
console.log(numbers) // -> [1,2,3,4]
```

Removing an element from the beginning

shift: Removing one array element in the beginning of the array.

```
const numbers = [1, 2, 3, 4, 5]
numbers.shift() // -> remove one item from the beginning
console.log(numbers) // -> [2,3,4,5]
```

Add an element from the beginning

unshift: Adding array element in the beginning of the array.

```
const numbers = [1, 2, 3, 4, 5]
numbers.unshift(0) // -> add one item from the beginning
console.log(numbers) // -> [0,1,2,3,4,5]
```

Reversing array order

reverse: reverse the order of an array.

```
const numbers = [1, 2, 3, 4, 5]
numbers.reverse() // -> reverse array order
console.log(numbers) // [5, 4, 3, 2, 1]
```

```
numbers.reverse()
console.log(numbers) // [1, 2, 3, 4, 5]
```

Sorting elements in array

sort: arrange array elements in ascending order. Sort takes a call back function, we will see how we use sort with a call back function in the coming sections.

```
const webTechs = [
  'HTML',
  'CSS',
  'JavaScript',
  'React',
  'Redux',
  'Node',
  'MongoDB'
]

webTechs.sort()
console.log(webTechs) // ["CSS", "HTML", "JavaScript", "MongoDB", "Node", "React", "Redux"]

webTechs.reverse() // after sorting we can reverse it
console.log(webTechs) // ["Redux", "React", "Node", "MongoDB", "JavaScript", "HTML", "CSS"]
```

Array of arrays

Array can store different data types including an array itself. Let us create an array of arrays

```
const firstNums = [1, 2, 3]
const secondNums = [1, 4, 9]

const arrayOfArray =  [[1, 2, 3], [1, 2, 3]]
console.log(arrayOfArray[0]) // [1, 2, 3]

 const frontEnd = ['HTML', 'CSS', 'JS', 'React', 'Redux']
 const backEnd = ['Node','Express', 'MongoDB']
 const fullStack = [frontEnd, backEnd]
 console.log(fullStack)   // [["HTML", "CSS", "JS", "React", "Redux"], ["Node", "Express", "Mo
 console.log(fullStack.length)  // 2
 console.log(fullStack[0])  // ["HTML", "CSS", "JS", "React", "Redux"]
 console.log(fullStack[1]) // ["Node", "Express", "MongoDB"]
```

🌕 You are diligent and you have already achieved quite a lot. You have just completed day 5 challenges and you are 5 steps a head in to your way to greatness. Now do some exercises for your brain and for your muscle.

💻 Exercise

Exercise: Level 1

```js
const countries = [
  'Albania',
  'Bolivia',
  'Canada',
  'Denmark',
  'Ethiopia',
  'Finland',
  'Germany',
  'Hungary',
  'Ireland',
  'Japan',
  'Kenya'
]

const webTechs = [
  'HTML',
  'CSS',
  'JavaScript',
  'React',
  'Redux',
  'Node',
  'MongoDB'
]
```

1. Declare an *empty* array;
2. Declare an array with more than 5 number of elements
3. Find the length of your array
4. Get the first item, the middle item and the last item of the array
5. Declare an array called *mixedDataTypes*, put different data types in the array and find the length of the array. The array size should be greater than 5
6. Declare an array variable name itCompanies and assign initial values Facebook, Google, Microsoft, Apple, IBM, Oracle and Amazon
7. Print the array using *console.log()*
8. Print the number of companies in the array
9. Print the first company, middle and last company
10. Print out each company
11. Change each company name to uppercase one by one and print them out
12. Print the array like as a sentence: Facebook, Google, Microsoft, Apple, IBM,Oracle and Amazon are big IT companies.

13. Check if a certain company exists in the itCompanies array. If it exist return the company else return a company is *not found*
14. Filter out companies which have more than one 'o' without the filter method
15. Sort the array using *sort()* method
16. Reverse the array using *reverse()* method
17. Slice out the first 3 companies from the array
18. Slice out the last 3 companies from the array
19. Slice out the middle IT company or companies from the array
20. Remove the first IT company from the array
21. Remove the middle IT company or companies from the array
22. Remove the last IT company from the array
23. Remove all IT companies

Exercise: Level 2

1. Create a separate countries.js file and store the countries array in to this file, create a separate file web_techs.js and store the webTechs array in to this file. Access both file in main.js file

2. First remove all the punctuations and change the string to array and count the number of words in the array

```
let text =
'I love teaching and empowering people. I teach HTML, CSS, JS, React, Python.'
console.log(words)
console.log(words.length)
```

```
["I", "love", "teaching", "and", "empowering", "people", "I", "teach", "HTML", "CSS", "JS"

13
```

3. In the following shopping cart add, remove, edit items

```
const shoppingCart = ['Milk', 'Coffee', 'Tea', 'Honey']
```

- add 'Meat' in the beginning of your shopping cart if it has not been already added
- add Sugar at the end of you shopping cart if it has not been already added
- remove 'Honey' if you are allergic to honey
- modify Tea to 'Green Tea'

4. In countries array check if 'Ethiopia' exists in the array if it exists print 'ETHIOPIA'. If it does not exist add to the countries list.

5. In the webTechs array check if Sass exists in the array and if it exists print 'Sass is a CSS preprocess'. If it does not exist add Sass to the array and print the array.

6. Concatenate the following two variables and store it in a fullStack variable.

```
const frontEnd = ['HTML', 'CSS', 'JS', 'React', 'Redux']
const backEnd = ['Node','Express', 'MongoDB']

console.log(fullStack)
```

```
["HTML", "CSS", "JS", "React", "Redux", "Node", "Express", "MongoDB"]
```

Exercise: Level 3

1. The following is an array of 10 students ages:

```
const ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]
```

   - Sort the array and find the min and max age
   - Find the median age(one middle item or two middle items divided by two)
   - Find the average age(all items divided by number of items)
   - Find the range of the ages(max minus min)
   - Compare the value of (min - average) and (max - average), use *abs()* method 1.Slice the first ten countries from the countries array

2. Find the middle country(ies) in the countries array

3. Divide the countries array into two equal arrays if it is even. If countries array is not even , one more country for the first half.

🎉 CONGRATULATIONS ! 🎉

<< Day 4 | Day 6 >>