# What is an Array?

An array is a special variable, which can hold more than one value at a time. For example, if you want to store a list of fruits, you can use an array:

let fruits = ["Apple", "Banana", "Mango", "Orange"];

# Creating a Arrays

This is 2 ways

1. Arrays literals

let fruits = ["Apple", "Banana", "Mango", "Orange"];

1. `Array` constructor:

let fruits = new Array("Apple", "Banana", "Mango", "Orange");

# Accessing Array Elements

let firstFruit = fruits[0]; // Apple

let secondFruit = fruits[1]; // Banana

# Array Methods

JavaScript arrays come with a variety of methods that allow you to perform different operations on the arrays.

1. push()

Adds one or more elements to the end of an array:

fruits.push("Pineapple");

console.log(fruits); // ["Apple", "Banana", "Mango", "Orange", "Pineapple"]

2. pop()

Removes the last element from an array:

let lastFruit = fruits.pop();

console.log(lastFruit); // Pineapple

console.log(fruits); // ["Apple", "Banana", "Mango", "Orange"]

3. shift()

Removes the first element from an array:

let firstFruit = fruits.shift();

console.log(firstFruit); // Apple

console.log(fruits); // ["Banana", "Mango", "Orange"]

4. unshift()

Adds one or more elements to the beginning of an array:

fruits.unshift("Strawberry");

console.log(fruits); // ["Strawberry", "Banana", "Mango", "Orange"]

5. length

Returns the number of elements in an array:

console.log(fruits.length); // 4

6. concat()

Joins two or more arrays:

let moreFruits = ["Pineapple", "Grapes"];

let allFruits = fruits.concat(moreFruits);

console.log(allFruits); // ["Strawberry", "Banana", "Mango", "Orange", "Pineapple", "Grapes"]

7. slice()

Returns a new array containing a portion of an existing array:

let someFruits = fruits.slice(1, 3);

console.log(someFruits); // ["Banana", "Mango"]

1. splice()

Adds or removes elements from an array:

// Remove 1 element at index 2

fruits.splice(2, 1);

console.log(fruits); // ["Strawberry", "Banana", "Orange"]

// Add 2 elements at index 2

fruits.splice(2, 0, "Mango", "Kiwi");

console.log(fruits); // ["Strawberry", "Banana", "Mango", "Kiwi", "Orange"]

9. forEach()

Executes a provided function once for each array element:

fruits.forEach(function(fruit) {

console.log(fruit);

});

// Output:

// Strawberry

// Banana

// Mango

// Kiwi

// Orange

10. map()

Creates a new array with the results of calling a provided function on every element in the calling array:

let upperCaseFruits = fruits.map(function(fruit) {

return fruit.toUpperCase();

});

console.log(upperCaseFruits); // ["STRAWBERRY", "BANANA", "MANGO", "KIWI", "ORANGE"]

11. filter()

Creates a new array with all elements that pass the test implemented by the provided function:

let longFruits = fruits.filter(function(fruit) {

return fruit.length > 5;

});

console.log(longFruits); // ["Strawberry", "Banana", "Orange"]

12. reduce()

Executes a reducer function (that you provide) on each element of the array, resulting in a single output value:

let totalLength = fruits.reduce(function(total, fruit) {

return total + fruit.length;

}, 0);

console.log(totalLength); // 29

13. includes()

Determines whether an array includes a certain value among its entries:

let hasMango = fruits.includes("Mango");

console.log(hasMango); // true

14. indexOf()

Returns the first index at which a given element can be found in the array, or -1 if it is not present:

let indexOfKiwi = fruits.indexOf("Kiwi");

console.log(indexOfKiwi); // 3

15. join()

Joins all elements of an array into a string:

let fruitString = fruits.join(", ");

console.log(fruitString); // "Strawberry, Banana, Mango, Kiwi, Orange"