Array & String in JavaScript

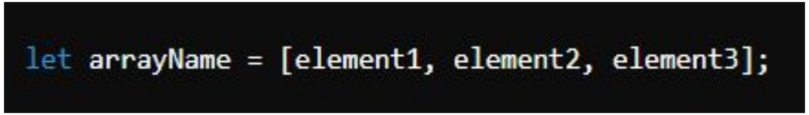
Exploring Array Methods, String Functions, and Looping

Array Functions (Push, Pop, Shift, Unshift, Splice)

Introduction to Arrays

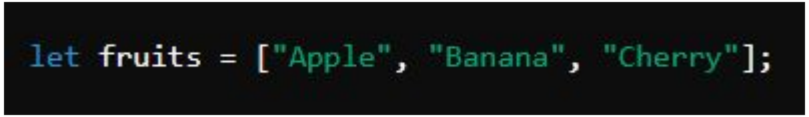
- **Definition**: An array is a collection of elements, each identified by an index.

Syntax :



```
let arrayName = [element1, element2, element3];
```

Example :

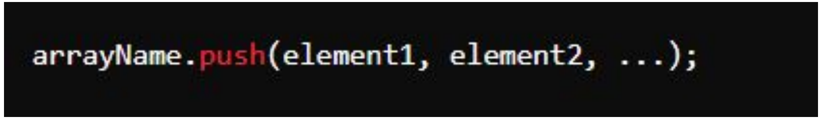


```
let fruits = ["Apple", "Banana", "Cherry"];
```

Array Method: Push

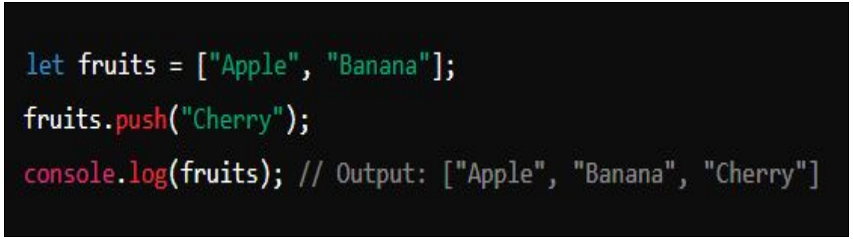
Purpose: Adds one or more elements to the end of an array.

Syntax :



```
arrayName.push(element1, element2, ...);
```

Example :

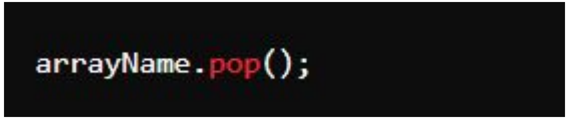


```
let fruits = ["Apple", "Banana"];
fruits.push("Cherry");
console.log(fruits); // Output: ["Apple", "Banana", "Cherry"]
```

Array Method: Pop

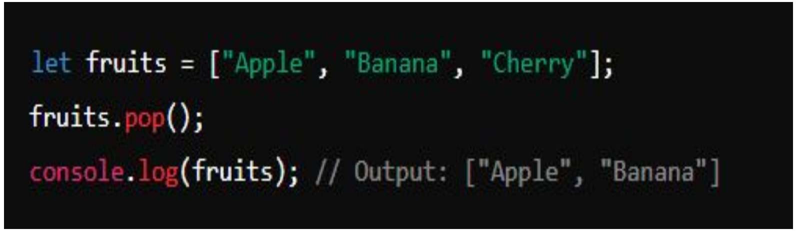
Purpose: Removes the last element from an array.

Syntax :



```
arrayName.pop();
```

Example :

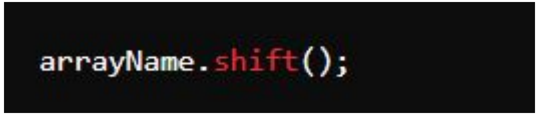


```
let fruits = ["Apple", "Banana", "Cherry"];
fruits.pop();
console.log(fruits); // Output: ["Apple", "Banana"]
```

Array Method: Shift

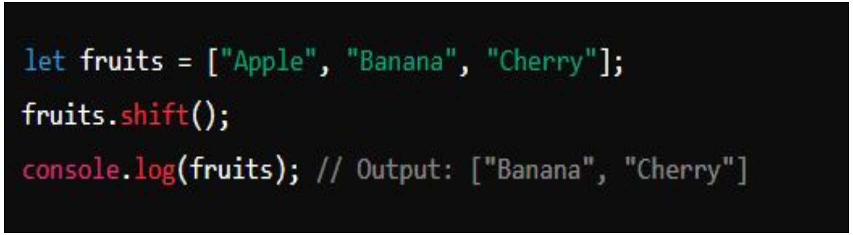
Purpose: Removes the first element from an array.

Syntax :



```
arrayName.shift();
```

Example :

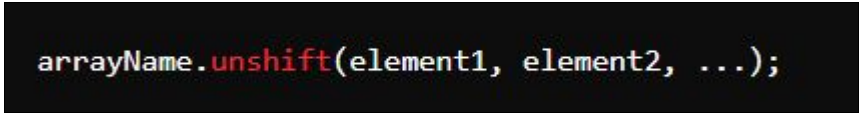


```
let fruits = ["Apple", "Banana", "Cherry"];
fruits.shift();
console.log(fruits); // Output: ["Banana", "Cherry"]
```

Array Method: Unshift

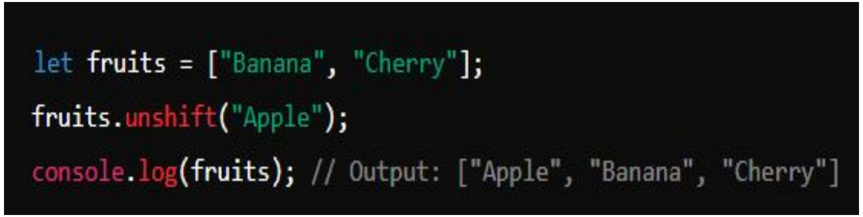
Purpose: Adds one or more elements to the beginning of an array.

Syntax :



```
arrayName.unshift(element1, element2, ...);
```

Example :

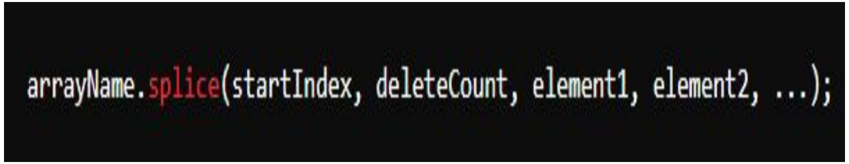


```
let fruits = ["Banana", "Cherry"];
fruits.unshift("Apple");
console.log(fruits); // Output: ["Apple", "Banana", "Cherry"]
```

Array Method: Splice

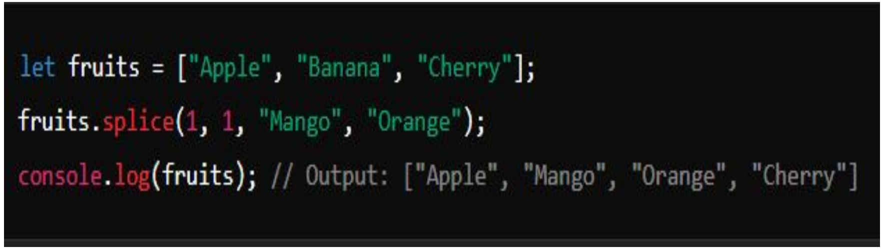
Purpose: Adds/Removes elements from an array at a specific index.

Syntax :



```
arrayName.splice(startIndex, deleteCount, element1, element2, ...);
```

Example :

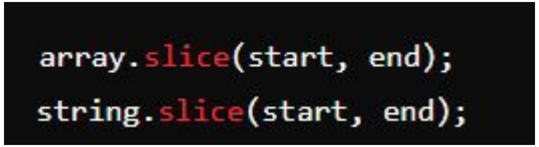


```
let fruits = ["Apple", "Banana", "Cherry"];
fruits.splice(1, 1, "Mango", "Orange");
console.log(fruits); // Output: ["Apple", "Mango", "Orange", "Cherry"]
```

Array Method: Slice

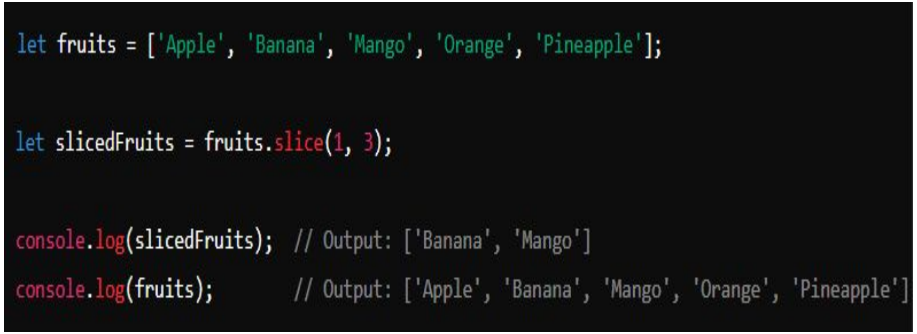
Purpose : The `slice()` method in JavaScript is used to extract a section of an array or string and return it as a new array or string without modifying the original array or string.

Syntax :



```
array.slice(start, end);
string.slice(start, end);
```

Example :

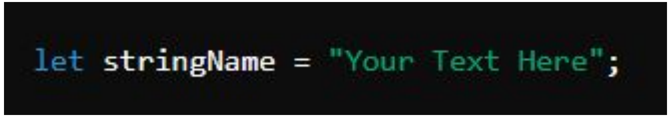


```
let fruits = ['Apple', 'Banana', 'Mango', 'Orange', 'Pineapple'];

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

console.log(slicedFruits);  // Output: ['Banana', 'Mango']
console.log(fruits);        // Output: ['Apple', 'Banana', 'Mango', 'Orange', 'Pineapple']
```

Introduction to Strings

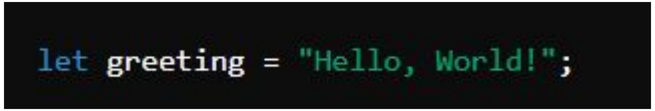
Definition: A string is a sequence of characters.

Syntax :



```
let stringName = "Your Text Here";
```

Example :

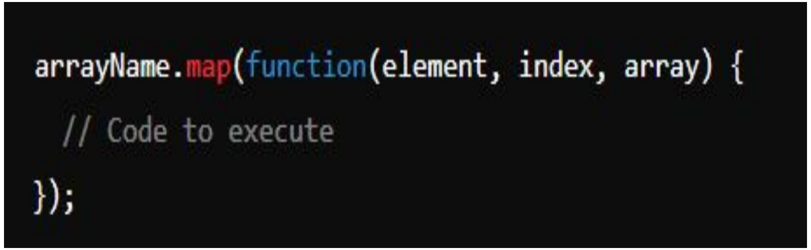


```
let greeting = "Hello, World!";
```

Array Method: map

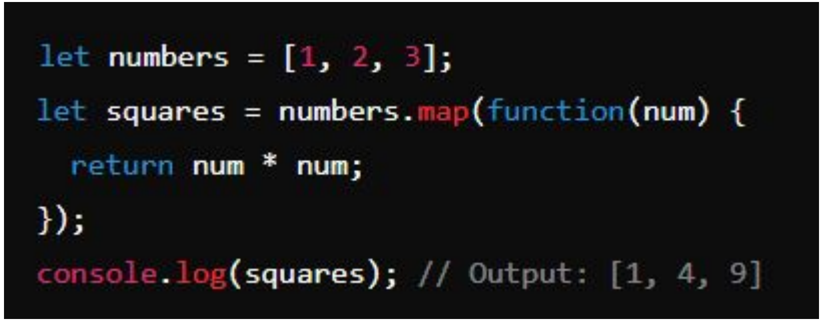
Purpose: Creates a new array with the results of calling a provided function on every element in the array.

Syntax :



```
arrayName.map(function(element, index, array) {
  // Code to execute
});
```

Example :

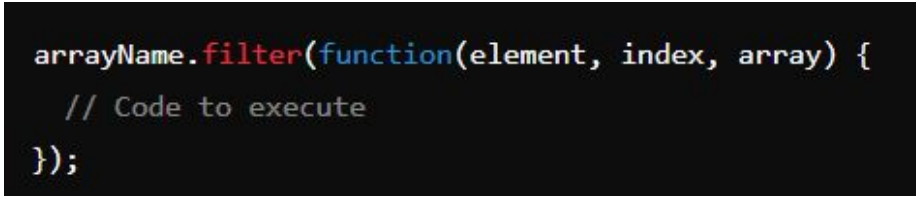


```
let numbers = [1, 2, 3];
let squares = numbers.map(function(num) {
  return num * num;
});
console.log(squares); // Output: [1, 4, 9]
```

Array Method: filter

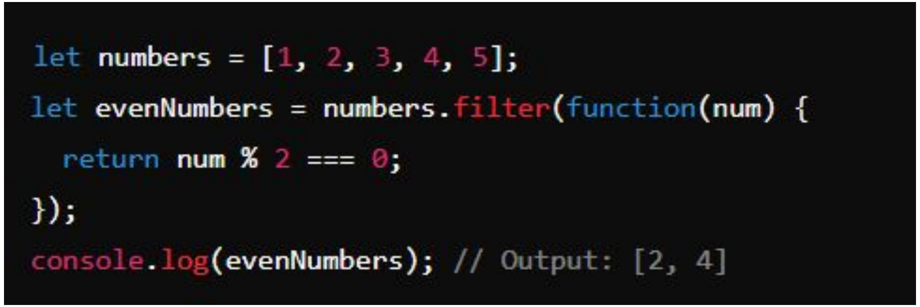
Purpose: Creates a new array with all elements that pass the test implemented by the provided function.

Syntax :



```
arrayName.filter(function(element, index, array) {
  // Code to execute
});
```

Example :

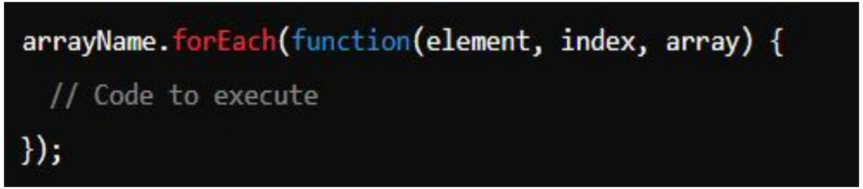


```
let numbers = [1, 2, 3, 4, 5];
let evenNumbers = numbers.filter(function(num) {
  return num % 2 === 0;
});
console.log(evenNumbers); // Output: [2, 4]
```

Array Method: forEach

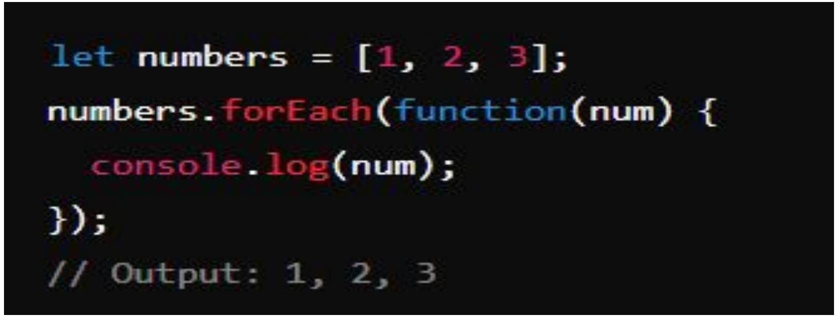
Purpose: Executes a provided function once for each array element.

Syntax :



```
arrayName.forEach(function(element, index, array) {
  // Code to execute
});
```

Example :

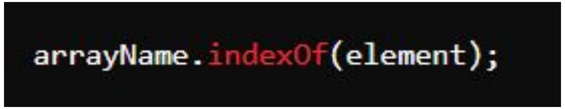


```
let numbers = [1, 2, 3];
numbers.forEach(function(num) {
  console.log(num);
});
// Output: 1, 2, 3
```

Array Method: indexOf

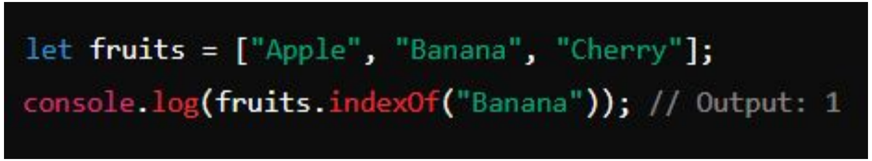
Purpose: Returns the first index at which a given element can be found in the array, or -1 if it is not present.

Syntax :



```
arrayName.indexOf(element);
```

Example :

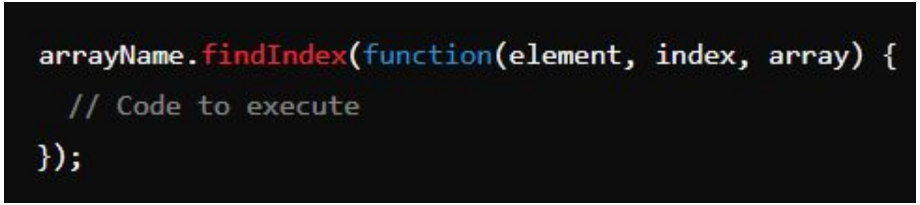


```
let fruits = ["Apple", "Banana", "Cherry"];
console.log(fruits.indexOf("Banana")); // Output: 1
```

Array Method: findIndex

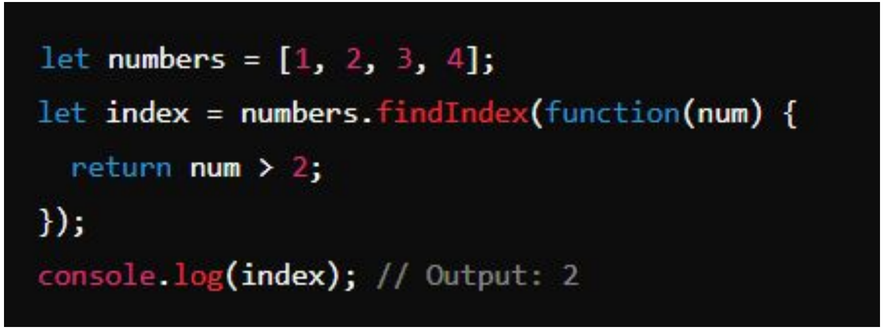
Purpose: Returns the index of the first element in the array that satisfies the provided testing function. Otherwise, it returns -1.

Syntax :



```
arrayName.findIndex(function(element, index, array) {
  // Code to execute
});
```

Example :

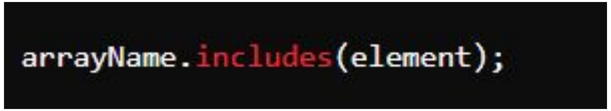


```
let numbers = [1, 2, 3, 4];
let index = numbers.findIndex(function(num) {
  return num > 2;
});
console.log(index); // Output: 2
```

Array Method: includes

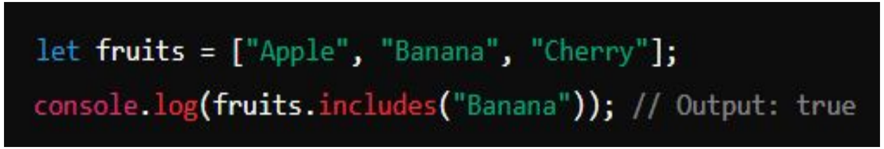
Purpose: Determines whether an array includes a certain element, returning true or false as appropriate.

Syntax :



```
arrayName.includes(element);
```

Example :

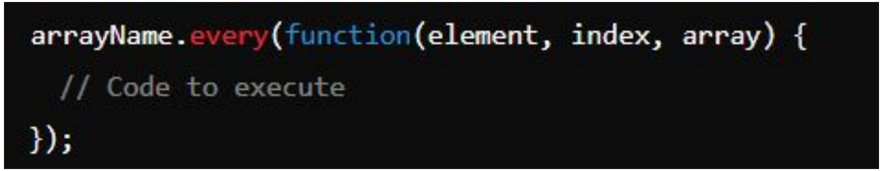


```
let fruits = ["Apple", "Banana", "Cherry"];
console.log(fruits.includes("Banana")); // Output: true
```

Array Method: every

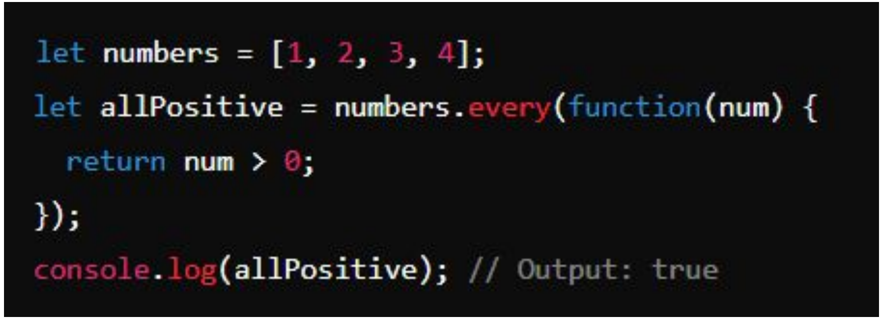
Purpose: Tests whether all elements in the array pass the test implemented by the provided function.

Syntax :



```
arrayName.every(function(element, index, array) {
  // Code to execute
});
```

Example :

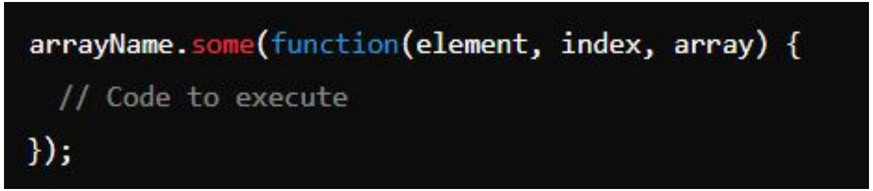


```
let numbers = [1, 2, 3, 4];
let allPositive = numbers.every(function(num) {
  return num > 0;
});
console.log(allPositive); // Output: true
```

Array Method: some

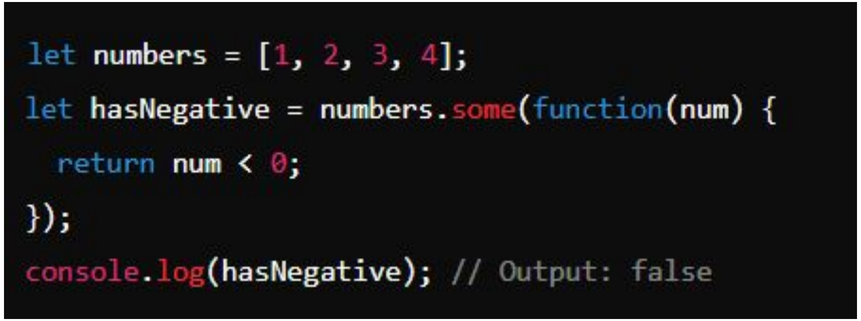
Purpose: Tests whether at least one element in the array passes the test implemented by the provided function.

Syntax :



```
arrayName.some(function(element, index, array) {
  // Code to execute
});
```

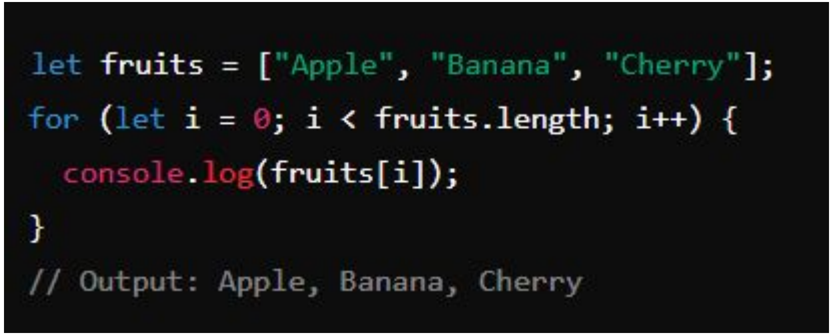
Example :



```
let numbers = [1, 2, 3, 4];
let hasNegative = numbers.some(function(num) {
  return num < 0;
});
console.log(hasNegative); // Output: false
```

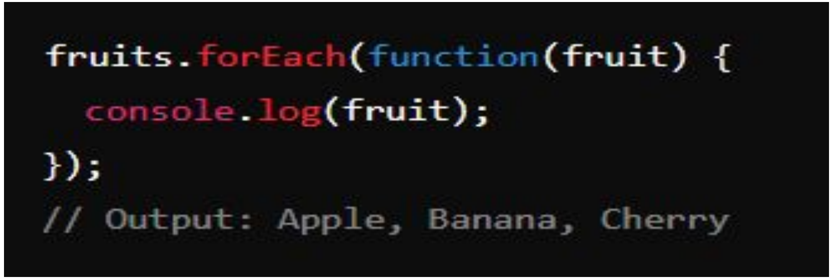
Looping in Arrays

Using for loop :



```
let fruits = ["Apple", "Banana", "Cherry"];
for (let i = 0; i < fruits.length; i++) {
  console.log(fruits[i]);
}
// Output: Apple, Banana, Cherry
```

Using `forEach` loop :



```
fruits.forEach(function(fruit) {
  console.log(fruit);
});
// Output: Apple, Banana, Cherry
```

Summary

- **Array Methods**: Push, Pop, Shift, Unshift, Splice, map, filter, forEach, indexOf, findIndex, includes, every, some.

- **Looping Techniques**: Using `for` loop and `forEach` method to iterate over arrays.