TechMaster
Học là có việc

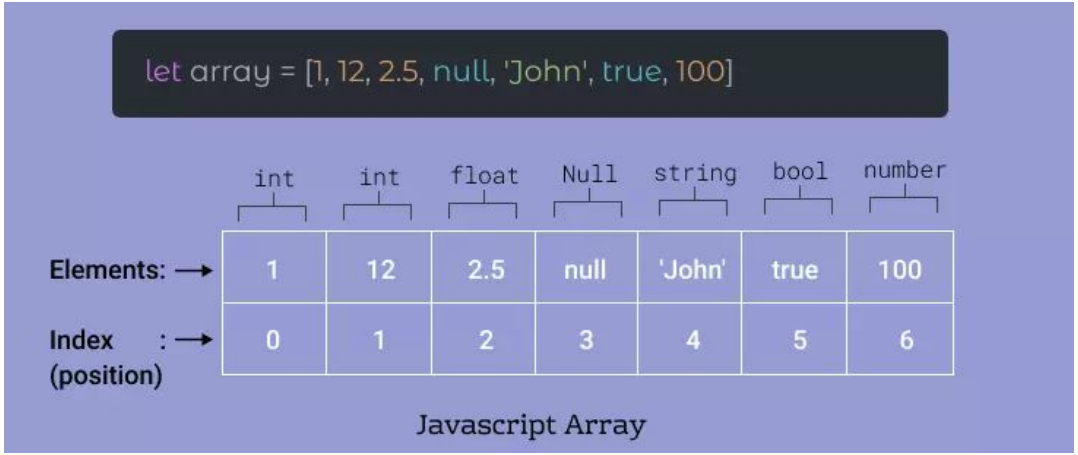
Javascript Basic P2
Array

Trần Huy Hoàng
Email: huyhoang.tran6669@gmail.com
Phone: 0788.719.666

Nội dung

1. Giới thiệu chung về Array
2. Khai báo, tạo mới mảng
3. Truy cập đến các phần tử trong mảng
4. Thêm phần tử vào mảng
5. Thay đổi giá trị của phần tử trong mảng
6. Xóa phần tử khỏi mảng
7. Các hàm hay dùng với array
8. Lab1
9. First class function
10. Arrow function
11. ES6
12. Lab2

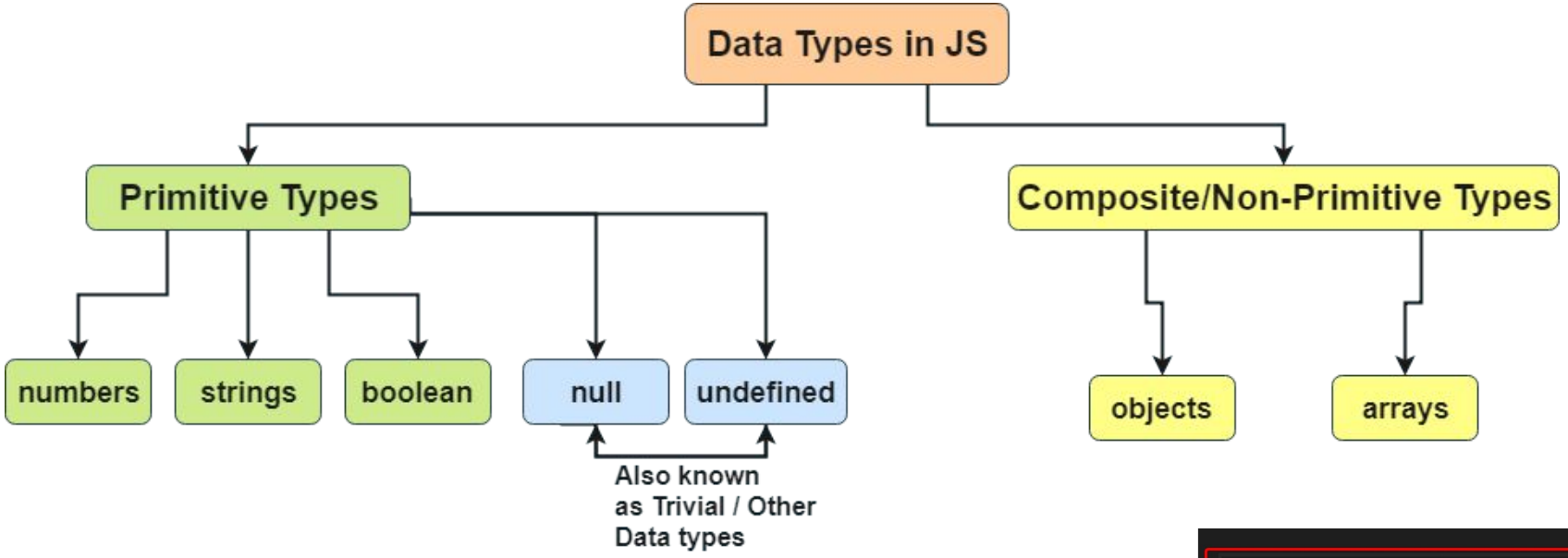
1. Giới thiệu chung về Array



Array(mảng) được dùng để lưu trữ nhiều phần tử

Mỗi phần tử của mảng (element) có thể có các kiểu dữ liệu khác nhau

Kiểu dữ liệu trong Javascript



Để kiểm tra kiểu dữ liệu của 1 biến, chúng ta sử dụng toán tử **typeof**

```
let name = "Thuan nguyen";
console.log(typeof name);
```

String

2. Khai báo, tạo mới mảng

C1 Tạo mảng với cặp ngoặc []

```
const array1 = ["eat", "sleep"];
```

```
// empty array
const myList = [ ];

// array of numbers
const numberArray = [ 2, 4, 6, 8];

// array of strings
const stringArray = [ 'eat', 'work', 'sleep'];

// array with mixed data types
const newData = ['work', 'exercise', 1, true];
```

C2 Tạo mảng với từ khóa Array trong JS

```
const array2 = new Array("eat", "sleep");
```

```
const newData = [
    {'task1': 'exercise'},
    [1, 2 ,3],
    function hello() { console.log('hello')}
];
```

3. Truy cập đến các phần tử trong mảng

Array indexing in JavaScript

- Các phần tử trong mảng được truy cập thông qua chỉ số (**index**)
- Chỉ số của mảng bắt đầu từ 0, kết thúc ở arr.length - 1

3. Truy cập đến các phần tử trong mảng

Lab1: Viết hàm tính tổng các số lẻ và chia hết cho 3 trong mảng

arr[1, 2, 3, 4, 5, 6]

Lab2: Viết hàm tính tổng các số nguyên tố trong mảng

arr[2, 3, 4, 5, 6, 10, 12, 17]

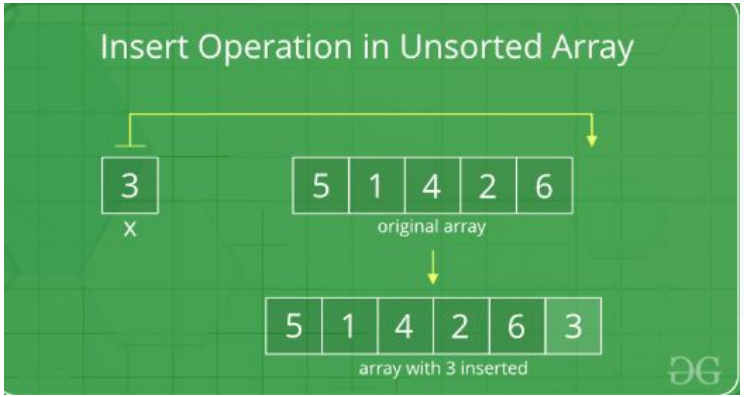
Lab3: Viết function cho phép truyền vào 1 mảng các số, kết quả trả về là 1 mảng mới với các số là số dư tương ứng khi chia mảng cũ cho 2

Ví dụ : [4,2,5,6,2,7] => [0,0,1,0,0,1]

Lab4: Viết function để kiểm tra 1 giá trị có nằm trong mảng hay không?

4. Thêm phần tử vào mảng

Push() method




```
let dailyActivities = ['eat', 'sleep'];

// add an element at the end
dailyActivities.push('exercise');

console.log(dailyActivities); //  ['eat', 'sleep', 'exercise']
```

4. Thêm phần tử vào mảng

unshift() method

```
let dailyActivities = ['eat', 'sleep'];

//add an element at the start
dailyActivities.unshift('work');
```

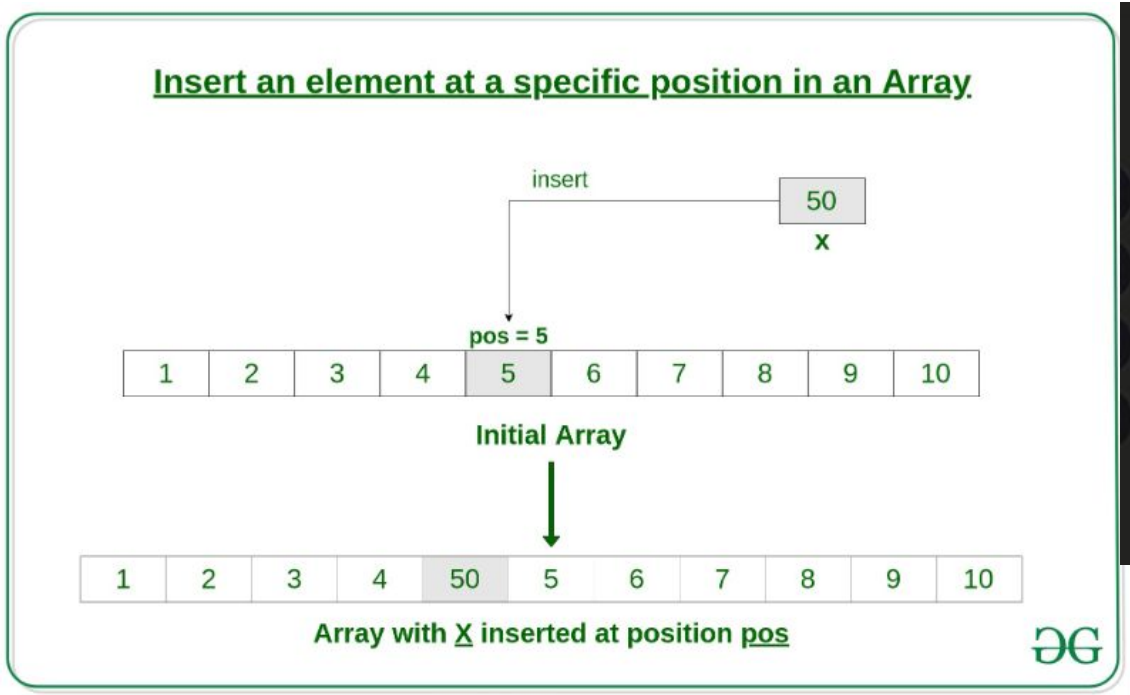
Lab 5: Viết hàm trả về một mảng chỉ gồm các số chia hết cho 3 và 5 nhỏ hơn 100

Lab 6: Viết hàm trả về một mảng chỉ gồm các số nguyên tố từ một mảng đã cho ban đầu

[1, 2, 4, 5, 8, 9, 11] => [2, 5, 11]

4. Thêm phần tử vào mảng

Thêm phần tử vào một vị trí bất kì




```javascript
// javascript Program to Insert an element
// at a specific position in an Array
    function insertElement(arr, n, x, pos)
    {

        // shift elements to the right
        // which are on the right side of pos
        var i =0;
        for (i; i >= pos; i--)
        {
            arr[i + 1] = arr[i];
        }
        arr[pos] = x;
    }
```

4. Thêm phần tử vào mảng

```
var arr = Array(15).fill(0);
arr[0] = 2;
arr[1] = 4;
arr[2] = 1;
arr[3] = 8;
arr[4] = 5;
var n = 5;
var x = 10;
var pos = 2;
console.log("Before Insertion: ");
var i = 0;
for (i; i < n; i++)
{
    console.log(arr[i] + " ");
}

// Inserting key at specific position
insertElement(arr, n, x, pos);
n += 1;
console.log("\n\nAfter Insertion: ");
i = 0;
for (i; i < n; i++)
{
    console.log(arr[i] + " ");
}
```

5. Thay đổi giá trị của phần tử trong mảng

Question: Mảng sau sẽ thay đổi như thế nào trong 2 ví dụ sau?

```
let dailyActivities = [ 'eat', 'sleep'];

// this will add the new element 'exercise' at the 2 index
dailyActivities[2] = 'exercise';
```

```
let dailyActivities = [ 'eat', 'sleep'];

// this will add the new element 'exercise' at the 3 index
dailyActivities[3] = 'exercise';
```

6. Xóa phần tử khỏi mảng

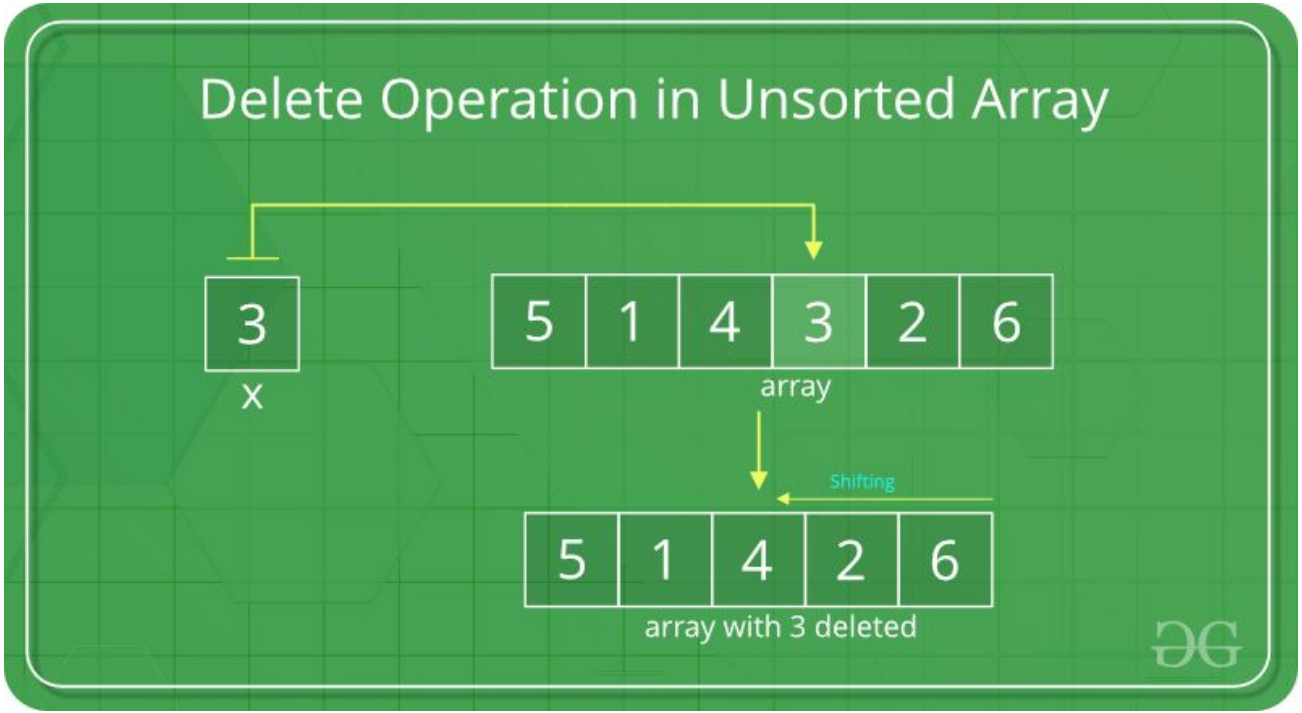
```
let dailyActivities = ['work', 'eat', 'sleep', 'exercise'];

// remove the last element
dailyActivities.pop();
console.log(dailyActivities); // ['work', 'eat', 'sleep']

// remove the last element from ['work', 'eat', 'sleep']
const removedElement = dailyActivities.pop();

//get removed element
console.log(removedElement); // 'sleep'
console.log(dailyActivities);  // ['work', 'eat']
```

Pop() method

```
let dailyActivities = ['work', 'eat', 'sleep'];

// remove the first element
dailyActivities.shift();

console.log(dailyActivities); // ['eat', 'sleep']
```

Shift() method

6. Xóa phần tử khỏi mảng



6. Xóa phần tử khỏi mảng

Lab 7: Viết hàm trả xóa các phần tử có giá trị bằng k trong một mảng

```
// function to search a key to
// be deleted
function findElement(arr,n,key)
{
    let i;
    for (i = 0; i < n; i++)
        if (arr[i] == key)
            return i;

    return -1;
}
```

```
// Function to delete an element
function deleteElement(arr,n,key)
{
    // Find position of element to be
    // deleted
    let pos = findElement(arr, n, key);

    if (pos == -1)
    {
        document.write("Element not found");
        return n;
    }

    // Deleting element
    let i;
    for (i = pos; i< n - 1; i++)
        arr[i] = arr[i + 1];

    return n - 1;
}
```

7. Các hàm hay dùng với array trong javascript

STT	Hàm	Chức năng
1	indexOf()	Trả về vị trí của 1 phần tử trong array
2	find()	Trả về vị trí xuất hiện đầu tiên của một phần tử trong array
3	concat()	Nối hai hay nhiều mảng với nhau
4	forEach()	Duyệt các phần tử trong mảng
5	includes()	Kiểm tra một mảng có chứa một phần tử nào đó không
6	sort()	Sắp xếp một mảng
7	slice()	Cắt một mảng con từ mảng to
8	push()	Thêm một phần tử vào cuối
9	Pop()	Xóa một phần tử ở cuối
10	shift()	Xóa một phần tử ở đầu
11	unshift()	Thêm một phần tử vào đầu

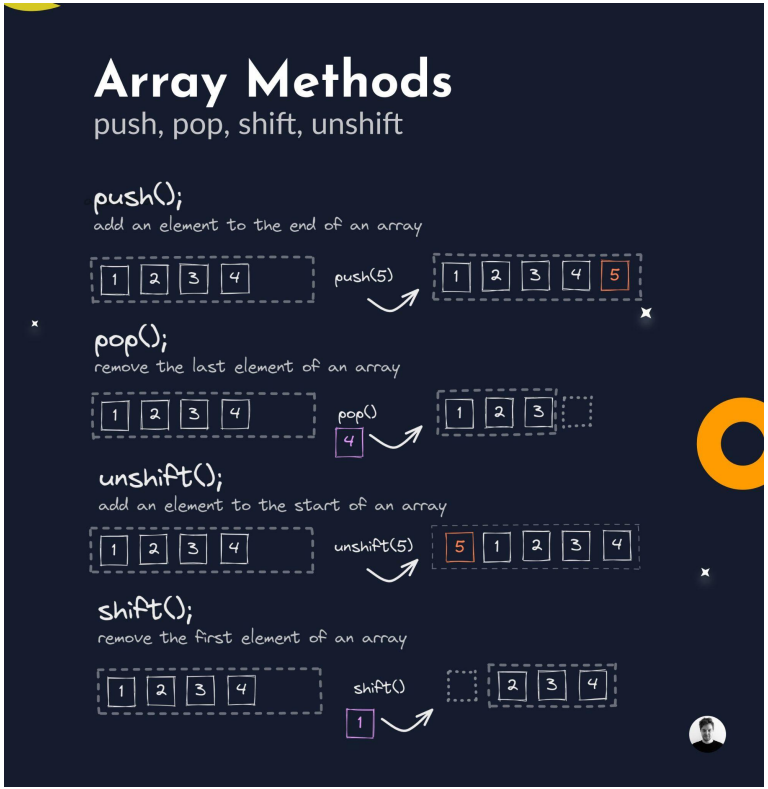
7. Các hàm hay dùng với array trong javascript

```
// 2 Ways to Merge Arrays

const cars = ['🚗', '🚙'];
const trucks = ['🚚', '🚛'];

// Method 1: Concat
const combined1 = [].concat(cars, trucks);

// Method 2: Spread
const combined2 = [...cars, ...trucks];

// Result
// [ '🚗', '🚙', '🚚', '🚛' ]
```

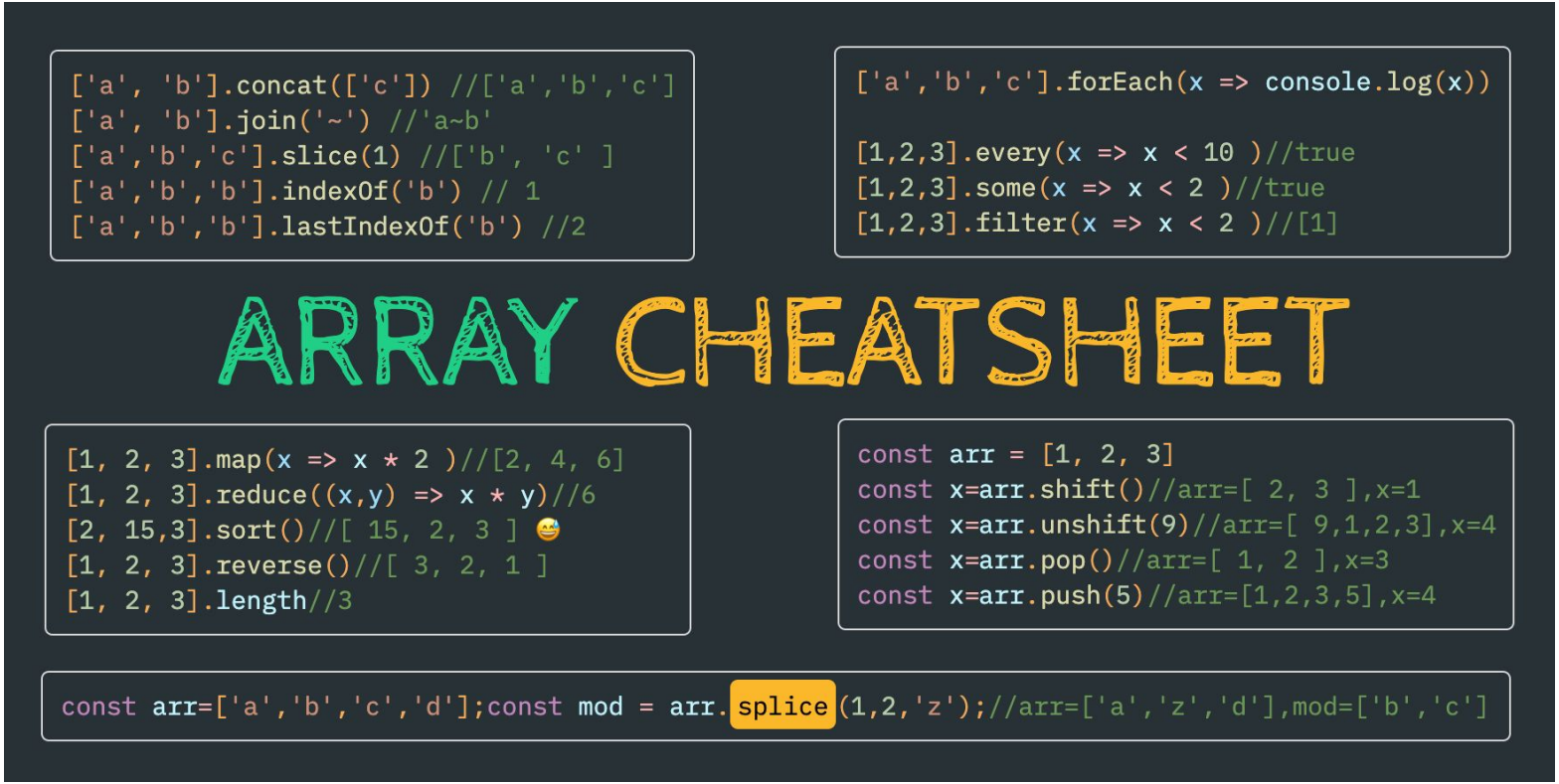
```
const cars = ['🚗', '🚙'];
const trucks = ['🚚', '🚛'];

const combined = cars.concat(trucks);
// [ '🚗', '🚙', '🚚', '🚛' ]

console.log(cars); // ['🚗', '🚙'];
console.log(trucks); // ['🚚', '🚛'];
```

7. Các hàm hay dùng với array trong javascript

```
let numbers = ["one","two","three"];

numbers.forEach(function(value, index) {
  console.log(index, value);
});

// => 0, one
// => 1, two
// => 2, three

numbers.forEach((value, index) => {
  console.log(index, value);
});

// => 0, one
// => 1, two
// => 2, three
```



7. Các hàm hay dùng với array trong javascript



8. Lab1

Lab 8: Viết function cho phép truyền vào 1 mảng các số, kết quả trả về là 1 mảng mới với các số là số dư tương ứng khi chia mảng cũ cho 2

Ví dụ : [4,2,5,6,2,7] => [0,0,1,0,0,1]

Lab 9: Viết function truyền vào 1 chuỗi, hãy sao chép đó chuỗi lên 10 lần (Sử dụng array để làm)

Ví dụ: repeatString(‘a’) => Kết quả trả về là ‘aaaaaaaaaa’

Lab 10: Viết function truyền vào 1 chuỗi, hãy sao chép đó chuỗi lên 10 lần, ngăn cách nhau bởi dấu gạch ngang (Sử dụng array để làm)

Ví dụ: repeatString(‘a’) => Kết quả trả về là ‘a-a-a-a-a-a-a-a-a-a’

9. First Class function

Một ngôn ngữ hỗ trợ first class functions cho phép

- Function có thể gán cho biến
- Truyền vào làm tham số cho function khác
- Có thể được trả về từ functions khác.

```
// 1. Function có thể gán cho biến
const sum = function (a, b) {
    return a + b
}

console.log(sum(3, 4));
```

```
// 2. Truyền vào làm tham số cho function khác

// Kiểm tra 1 số có là số lẻ hay không
const isOdd = function (num) {
    return num % 2 == 1;
}

// In ra danh sách số lẻ trong mảng
const listOdd = function (arr, isOdd) {
    for (let i = 0; i < arr.length; i++) {
        if (isOdd(arr[i])) {
            console.log(arr[i]);
        }
    }
}
```

```
// 3. Có thể được trả về từ functions khác
const funcA = function (a, b) {
    return function (c) {
        return a + b + c
    }
}

const funcReturnFromFuncA = funcA(4, 5); //
console.log(funcReturnFromFuncA(6));
```

10. Arrow function

Trong ES6, Arrow Functions là một cú pháp mới dùng để viết các function trong JavaScript. Nó giúp tiết kiệm thời gian phát triển và đơn giản hóa phạm vi function (function scope)

- Cú pháp này là cách ngắn gọn hơn dùng để viết function. Ở đây sử dụng kí tự “=>”
- Bằng cách sử dụng arrow function, chúng ta tránh được việc phải gõ từ khóa function, return và dấu ngoặc nhọn.

```
// ES6
let sum = (a, b) => {
    return a + b;
};

// Viết ngắn gọn (chỉ áp dụng với function chỉ có 1 câu lệnh)
let sum = (a, b) => a + b;
```

11. ES6


ES1
ES2
ES3
ES4
ES5
ES6
ES. NEXT
1997
1998
1999
2009
2015
JS


export
import
MODULES
ARROWS
Arrows function
ADVANTAGES OF JAVASCRIPT
ES6
Template Literals
string
ENHANCED OBJECT LITERALS
CLASSES
Exist from ES6
BLOCK SCOPING
PROMISES
Rest parameters
For of
Parameter default
Let, const

11. ES6

Rest Parameter

Ví dụ, giả sử dụng ta muốn cộng nhiều số (mà chưa rõ là có bao nhiêu số). Ta sử dụng Rest Parameter như sau:

```
function sum(...args) {
    let total = 0;
    for (const x of args) {
        total += x;
    }
    return total;
}

console.log(sum(1, 2, 3));
console.log(sum(1, 2, 3, 4, 5, 6));
```

11. ES6

Rest Operator

Bài toán: Tìm max của 1 mảng sử dụng hàm Math.max()

```
let arr = [3, 5, 1];

console.log( Math.max(arr) ); // NaN
```

```
Math.max(arr[0], arr[1], arr[2])
```

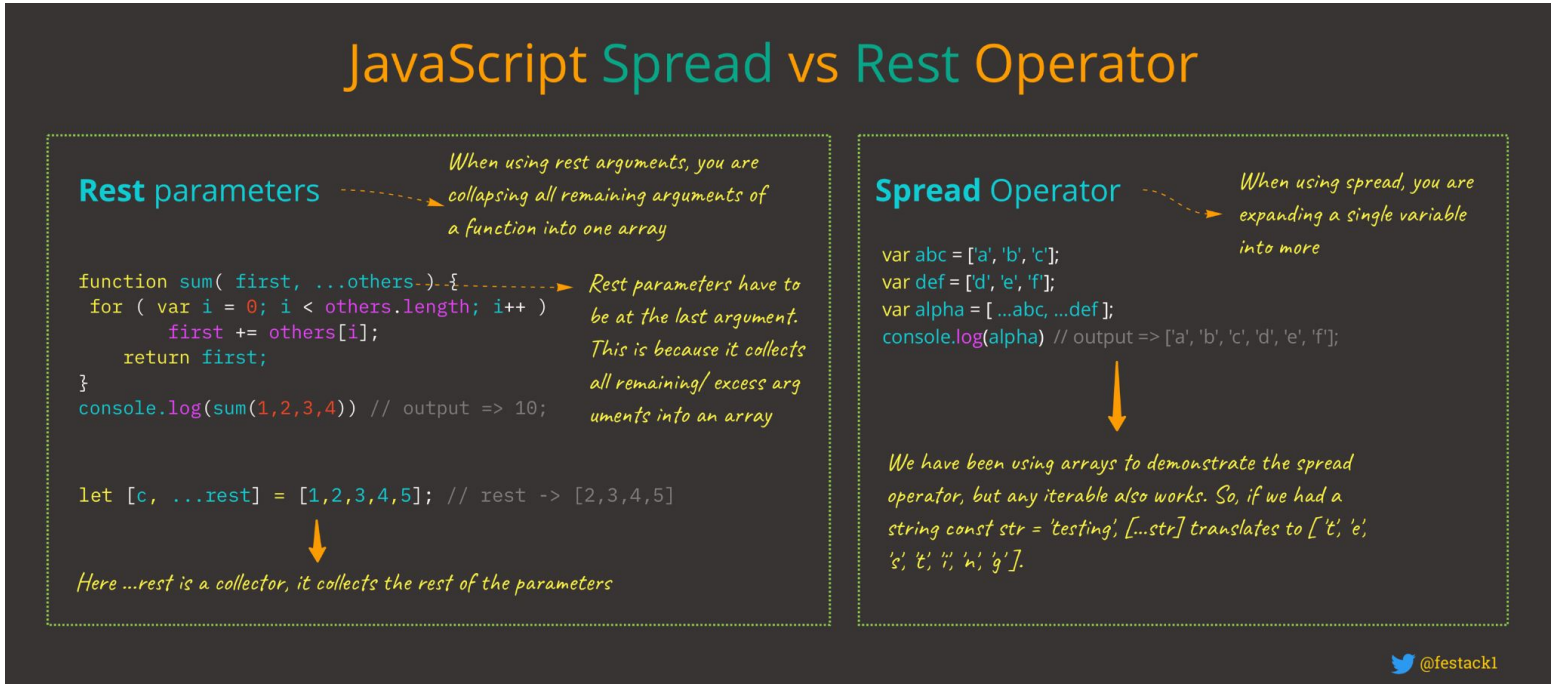
```
let arr = [3, 5, 1];

console.log( Math.max(...arr) );
```

```
let arr1 = [1, -2, 3, 4];
let arr2 = [8, 3, -8, 1];

console.log( Math.max(...arr1, ...arr2) ); // 8
```

```
let arr1 = [1, -2, 3, 4];
let arr2 = [8, 3, -8, 1];

console.log( Math.max(1, ...arr1, 2, ...arr2, 25) ); // 25
```

11. ES6

Rest Parameter cho phép bạn biểu diễn một số lượng vô hạn đối số dưới dạng một mảng.



12. Lab2

Bài 1: Viết function để kiểm tra 1 giá trị có nằm trong mảng hay không?

checkElementExist([1,2,3,4,5], 5) => true

checkElementExist([1,2,3,4,5], 6) => false

Bài 2: Viết function truyền vào 1 mảng các số, và 1 giá trị bất kỳ. Trả về mảng mới với các giá trị lớn hơn giá trị truyền vào

getElementGreater([1,2,3,4,5], 3) => [4,5]

getElementGreater([1,2,3,4,5], 5) => []

Bài 3: Viết function để tạo mã màu HEX ngẫu nhiên.

randomHexCode() => #728a98

randomHexCode() => #a72938

The End