ES6

# 1. Cú pháp sử dụng các hàm của mảng: forEach, map, filter, reduce

## forEach:

Mục đích: Dùng để lặp qua từng phần tử của mảng và thực thi một hàm cho mỗi phần tử. Không trả về kết quả, chỉ thực hiện các thao tác phụ.

Code ví dụ:

const array = [1, 2, 3];  
array.forEach((element) => {  
 console.log(element);  
});

## map:

Mục đích: Tạo ra một mảng mới bằng cách thực hiện một hàm lên mỗi phần tử của mảng gốc.

Code ví dụ:

const array = [1, 2, 3];  
const newArray = array.map((element) => element \* 2);  
console.log(newArray); // [2, 4, 6]

## filter:

Mục đích: Tạo ra một mảng mới chỉ chứa các phần tử thỏa mãn điều kiện trong hàm callback.

Code ví dụ:

const array = [1, 2, 3, 4, 5];  
const filteredArray = array.filter((element) => element > 2);  
console.log(filteredArray); // [3, 4, 5]

## reduce:

Mục đích: Gộp các phần tử của mảng thành một giá trị duy nhất (số, chuỗi, đối tượng, ...).

Code ví dụ:

const array = [1, 2, 3, 4];  
const sum = array.reduce((accumulator, currentValue) => accumulator + currentValue, 0);  
console.log(sum); // 10

# 2. Tìm hiểu về Spread Operator, Rest Parameters, và Destructuring

## Spread Operator (...):

Mục đích: Dùng để trải (spread) các phần tử của mảng hoặc thuộc tính của đối tượng thành các phần tử hoặc thuộc tính riêng lẻ.

Code ví dụ:

const arr1 = [1, 2, 3];  
const arr2 = [...arr1, 4, 5]; // [1, 2, 3, 4, 5]  
  
const obj1 = {a: 1, b: 2};  
const obj2 = {...obj1, c: 3}; // {a: 1, b: 2, c: 3}

## Rest Parameters (...):

Mục đích: Dùng để gom (collect) các đối số thành một mảng.

Code ví dụ:

function sum(...numbers) {  
 return numbers.reduce((acc, curr) => acc + curr, 0);  
}  
console.log(sum(1, 2, 3)); // 6

## Destructuring:

Mục đích: Dùng để trích xuất giá trị từ mảng hoặc đối tượng và gán cho các biến riêng biệt.

Code ví dụ:

Với mảng:

const [a, b] = [1, 2];  
console.log(a); // 1  
console.log(b); // 2

Với đối tượng:

const {name, age} = {name: 'John', age: 25};  
console.log(name); // 'John'  
console.log(age); // 25

# 3. Cú pháp Arrow function

Mục đích: Arrow function là một cách viết ngắn gọn hơn của function thông thường. Nó không có this riêng mà kế thừa từ ngữ cảnh hiện tại.

Code ví dụ:

const sum = (a, b) => a + b;  
console.log(sum(2, 3)); // 5

Nếu chỉ có một tham số thì không cần ngoặc:

const double = x => x \* 2;  
console.log(double(4)); // 8

# 4. Cú pháp import, export module

## export:

Mục đích: Dùng để xuất các hàm, biến, hoặc class từ một file để có thể sử dụng ở file khác.

Code ví dụ:

Xuất mặc định:

const myFunction = () => { ... };  
export default myFunction;

Xuất nhiều biến hoặc hàm:

export const a = 10;  
export const b = 20;

## import:

Mục đích: Dùng để nhập các hàm, biến, hoặc class từ một file khác.

Code ví dụ:

Nhập mặc định:

import myFunction from './myModule';

Nhập nhiều phần tử:

import { a, b } from './myModule';