ECMAScript 6+ (ES6 trở lên) mang đến nhiều cải tiến giúp JavaScript mạnh mẽ, dễ đọc và tối ưu hơn. Dưới đây là tổng hợp các tính năng quan trọng kèm ví dụ:

**1. let & const (Khai báo biến)**

* let: có phạm vi block scope, không bị hoisting như var.
* const: giống let, nhưng không thể gán lại giá trị.

javascript

Sao chépChỉnh sửa

let x = 10;

x = 20; // Hợp lệ

const y = 30;

y = 40; // Lỗi: Assignment to constant variable.

**2. Template Literals (Chuỗi Template)**

Giúp nối chuỗi dễ dàng hơn bằng dấu `.

javascript

Sao chépChỉnh sửa

const name = "Alice";

const age = 25;

console.log(`Tên tôi là ${name}, tôi ${age} tuổi.`);

**3. Arrow Functions (Hàm mũi tên)**

Cú pháp ngắn gọn hơn và không có this riêng.

javascript

Sao chépChỉnh sửa

const add = (a, b) => a + b;

console.log(add(5, 3)); // 8

**4. Default Parameters (Tham số mặc định)**

Giúp tránh lỗi khi không truyền đối số.

javascript

Sao chépChỉnh sửa

function greet(name = "Guest") {

console.log(`Hello, ${name}!`);

}

greet(); // "Hello, Guest!"

**5. Destructuring (Phân rã đối tượng & mảng)**

Trích xuất dữ liệu từ object hoặc array.

javascript

Sao chépChỉnh sửa

const person = { name: "Bob", age: 30 };

const { name, age } = person;

console.log(name, age); // Bob 30

const numbers = [1, 2, 3];

const [a, b] = numbers;

console.log(a, b); // 1 2

**6. Spread & Rest Operators (...)**

**Spread (...): Sao chép hoặc kết hợp mảng/đối tượng.**

javascript

Sao chépChỉnh sửa

const arr1 = [1, 2, 3];

const arr2 = [...arr1, 4, 5];

console.log(arr2); // [1, 2, 3, 4, 5]

**Rest (...): Gom nhiều giá trị vào một mảng.**

javascript

Sao chépChỉnh sửa

function sum(...numbers) {

return numbers.reduce((total, num) => total + num, 0);

}

console.log(sum(1, 2, 3, 4)); // 10

**7. Enhanced Object Literals (Cải tiến Object)**

Cho phép viết gọn hơn.

javascript

Sao chépChỉnh sửa

const name = "Charlie";

const age = 28;

const person = { name, age, greet() { console.log("Hello!"); } };

console.log(person); // { name: "Charlie", age: 28, greet: ƒ }

**8. Classes (Lớp)**

Giúp viết mã hướng đối tượng rõ ràng hơn.

javascript

Sao chépChỉnh sửa

class Animal {

constructor(name) {

this.name = name;

}

speak() {

console.log(`${this.name} makes a noise.`);

}

}

class Dog extends Animal {

speak() {

console.log(`${this.name} barks.`);

}

}

const dog = new Dog("Rex");

dog.speak(); // Rex barks.

**9. Modules (import/export)**

Giúp chia code thành nhiều file.

**File math.js**

javascript

Sao chépChỉnh sửa

export const add = (a, b) => a + b;

export const subtract = (a, b) => a - b;

**File main.js**

javascript

Sao chépChỉnh sửa

import { add, subtract } from "./math.js";

console.log(add(5, 3)); // 8

**10. Promises & Async/Await (Xử lý bất đồng bộ)**

**Promises**

javascript

Sao chépChỉnh sửa

const fetchData = () => new Promise(resolve => setTimeout(() => resolve("Data loaded"), 2000));

fetchData().then(data => console.log(data)); // "Data loaded" (sau 2 giây)

**Async/Await**

Viết code bất đồng bộ dễ hiểu hơn.

javascript

Sao chépChỉnh sửa

const fetchData = async () => {

return new Promise(resolve => setTimeout(() => resolve("Data loaded"), 2000));

};

(async () => {

const data = await fetchData();

console.log(data); // "Data loaded"

})();

Đây là các tính năng quan trọng của ES6+. Bạn muốn tìm hiểu sâu hơn về phần nào không? 🚀