

# **Bài 2**

## **Tổng Quan ES6**

Module: Web app building with JS



# Mục tiêu

- Trình bày khái niệm tổng quan về ES6
- Trình bày cách hoạt động của let & const
- Trình bày được Arrow Functions
- Triển khai được Default Parameters
- Triển khai được vòng lặp for
- Trình bày được Spread Attributes
- Trình bày được cơ chế module
- Sử dụng lambda expression
- Sử dụng được Maps, Sets
- Sử dụng được Static methods
- Sử dụng được Getter và Setter

# Tổng quan ES6



- ECMAScript 2015 hay ES2015 là một bản cập nhật quan trọng cho ngôn ngữ lập trình JavaScript
- ES2015 thường được gọi là ES6
- Một số tính năng mới như: các biến phạm vi, vòng lặp mới...
- Các thư viện framework hiện đại của JavaScript như React.js, Angular, Vue rất hay sử dụng các tính năng mới này.

# Let & const



- **let** là phạm vi khối `{ }` (block scope)
- **var** có phạm vi trong hàm (function scope) và hành vi hoisting trong JavaScript xảy ra

```
// Cú pháp ES5
for (var i = 0; i < 5; i++) {
    console.log(i); // 0,1,2,3,4
}
console.log(i); // 5
```

```
// Cú pháp ES6
for (let i = 0; i < 5; i++) {
    console.log(i); // 0,1,2,3,4
}
console.log(i); // undefined
```

- Từ khóa **const** mới được giới thiệu trong ES6 **định nghĩa các hằng số**
- **Hằng số** ở chế độ **chỉ đọc**

# ARROW FUNCTION (1)



- Cung cấp cú pháp ngắn gọn hơn để viết **biểu thức hàm**
- Các Arrow Function được định nghĩa bằng cú pháp mới, ký hiệu suy ra =>

// Cách 1: sử dụng Anonymous Function

```
var sum = function(a) {  
    return a + 100;  
}  
console.log(sum(2, 3)); // 5
```

// Cách 2: Sử dụng Arrow function

```
var sum = (a) => {  
    return a + 100;  
}
```

# ARROW FUNCTION (2)



- 1 tham số, nếu biểu thức tính toán là đơn giản thì không cần câu lệnh return.

```
param => expression
```

- Nhiều tham số, nếu biểu thức tính toán là đơn giản thì không cần câu lệnh return.

```
(param1, paramN) => expression
```

- Nhiều câu lệnh trong thân hàm, trong trường hợp này cần sử dụng dấu {} và câu lệnh return:

```
param => {  
    let a = 1;  
    return a + param;  
}
```

- Nhiều tham số cần dùng dấu (). Nhiều câu lệnh cần dùng ngoặc {} và câu lệnh return:

```
(param1, paramN) => {  
    let a = 1;  
    return a + param1 + paramN;  
}
```



# Giá trị mặc định

- Trong ES6, bạn có thể chỉ định các **giá trị mặc định** cho các tham số
- Nếu không có đối số nào được cung cấp cho hàm khi nó được gọi thì các giá trị tham số mặc định này sẽ được sử dụng

```
function greet(name, greeting, message = greeting + ' ' + name) {  
    return [name, greeting, message]  
}
```

```
greet('David', 'Hi') // ["David", "Hi", "Hi David"]
```

```
greet('David', 'Hi', 'Happy Birthday!') // ["David", "Hi", "Happy Birthday!"]
```



# Vòng lặp for

- ES6 giới thiệu vòng lặp **for of** sử dụng để duyệt qua từng phần tử trong mảng hoặc đối tượng.
- Các vòng lặp có thể sử dụng gồm: for of, for in, foreach, for

```
const iterable = [10, 20, 30];
```

```
for (const value of iterable) {  
  console.log(value);  
}
```

```
// 10  
// 20  
// 30
```





# Spread Attributes

- Giúp bạn làm việc với các cấu trúc dữ liệu nhanh và gọn hơn.
- Cú pháp spread đơn giản được biểu diễn bởi dấu 3 chấm: ...
- Spread Syntax cho phép duyệt qua các phần tử và truyền vào phương thức như các đối số.

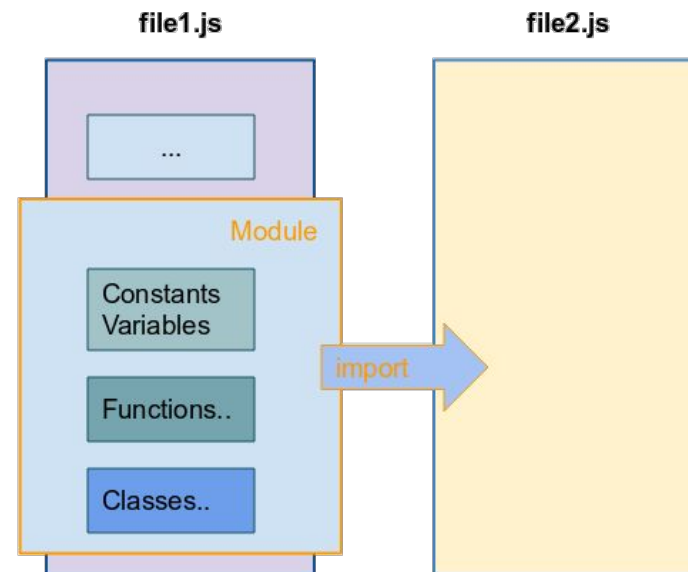
Ví dụ:

```
// Create an Array  
const tools = ['hammer', 'screwdriver']  
const otherTools = ['wrench', 'saw']  
  
// Unpack the tools Array into the allTools Array  
const allTools = [...tools, ...otherTools]  
console.log(allTools)
```

# Module



- ECMAScript 6 giới thiệu ES6 **Module Syntax** giúp các lập trình viên module hóa (modularize) code.
- Có thể viết code trên các tập tin riêng rẽ.
- Trên tập tin này có thể xuất (export) dữ liệu cần thiết dưới dạng các module, các tập tin khác có thể nhập (import) các module của tập tin đó để sử dụng.





# Tóm tắt bài học

- ECMAScript 6 (ES6) cung cấp **1 số tính năng mới** cho JavaScript
- Giúp code trở nên **tường minh và dễ viết** hơn.
- Các thư viện / framework hiện đại của JavaScript như React.js, Angular, Vue rất hay sử dụng các tính năng mới này.