

# ES6 Javascript

---

## 1 Lý thuyết:

ES6 (ES2015) là viết tắt của ECMAScript 6, là một phiên bản mới nhất của Javascript theo chuẩn ECMAScript.

ECMAScript là chuẩn quốc tế trong lập trình Javascript, là tập hợp những chuẩn mực, quy tắc trong lập trình Javascript như cách khai báo biến, định nghĩa method, định nghĩa class...

ECMAScript được công bố vào tháng 11 năm 1996, được phát triển bởi **Brendan Eich** và **Netscape**.

### 1.1 Javascript let

**let** statement cho phép khai báo biến trong scope {}, phạm vi hoạt động của biến đó chỉ thuộc scope {}.

Giá trị của biến x ngoài scope {} sẽ thay đổi khi ta khai báo lại trong scope {} khi dùng **var**.

```
var x = 10;  
// Now x is 10  
{  
    var x = 2;  
    // Now x is 2  
}  
// Now x is 2
```

Giá trị của biến x ngoài scope {} sẽ không thay đổi khi ta khai báo lại trong scope {} khi dùng **let**.

```
let x = 10;  
// Now x is 10  
{  
    let x = 2;  
    // Now x is 2  
}  
// Now x is 10
```

Giá trị của biến x sẽ không thay đổi khi ta khai báo lại trong scope {} khi dùng **let**.

### 1.2 Javascript const

**const** statement dùng để khai báo một biến với giá trị không thay đổi.

Biến `x` là một constant trong scope `{}`, nhưng nó là một biến thông thường ở ngoài scope `{}`.

```
let x = 10;
// Now x is 10
{
  const x = 2;
  // Now x is 2
}
// Now x is 10
```

### 1.3 Default Parameter Value

Trong chuẩn ES6 cho phép gán giá trị mặc định trong tham số truyền vào của một method.

```
function myFunction(x, y = 10) {
  // y is 10 if not passed or undefined
  return x + y;
}
myFunction(5); // will return 15
```

### 1.4 Array method

`Array.find`: Trả về giá trị được tìm thấy đầu tiên của mảng trong function kiểm tra.

```
var numbers = [4, 9, 16, 25, 29];
var first = numbers.find(myFunction);
// myFunction is test function
function myFunction(value, index, array) {
  return value > 18;
}
// first is 25
```

`Array.findIndex`: Trả về index của phần tử được tìm thấy đầu tiên của mảng trong function kiểm tra.

```
var numbers = [4, 9, 16, 25, 29];
var first = numbers.find(myFunction);
// myFunction is test function
function myFunction(value, index, array) {
  return value > 18;
}
// first is 3
```

### 1.5 New Number Properties

ES6 thêm một vài property cho đối tượng **Number**.

- EPSILON

- **MIN\_SAFE\_INTEGER**
- **MAX\_SAFE\_INTEGER**

```
// Hằng số điện môi
var x = Number.EPSILON;

// Giá trị nhỏ nhất của kiểu integer
var x = Number.MIN_SAFE_INTEGER;

// Giá trị lớn nhất của kiểu integer
var x = Number.MAX_SAFE_INTEGER;
```

## 1.6 New Number Method

ES6 cung cấp thêm một vài method mới cho đối tượng **Number**.

- **Number.isInteger()**
- **Number.isSafeInteger()**

```
// return true if the argument is an integer
Number.isInteger(10);           // returns true
Number.isInteger(10.5);        // returns false
// return true if the argument is a safe integer
Number.isSafeInteger(10);       // returns true
Number.isSafeInteger(12345678901234567890); //
returns false
```

## 1.7 New Global Methods

ES6 cung cấp thêm một vài method global.

- **isFinite()**
- **isNaN()**

```
// return false if the argument is infinity or NaN
isFinite(10/0);                 // returns false
isFinite(10/1);                 // returns true

// return true if the argument is NaN
isNaN("Hello");                 // returns true
```

## 1.8 Arrow Functions

Arrow Function là phương thức định nghĩa một function trong Javascript một cách ngắn gọn.

```
// ES5
```

```
var x = function(x, y) {  
    return x * y;  
}  
  
// ES6  
const x = (x, y) => x * y;
```

Trong Arrow Function không sở hữu đối tượng **this**.

## 2 Bài Tập:

### 2.1 BÀI TẬP 1

#### 2.1.1 Thời lượng: 10 phút

#### 2.1.2 Mô tả bài toán:

Cách sử dụng **let** trong ES6.

#### 2.1.3 Các bước thực hiện:

```
let x = 10;  
document.writeln("<pre>");  
document.writeln("The first value is" + x);  
  
function test() {  
    let x = 5;  
    document.writeln("The second value is" + x);  
}  
test();  
document.writeln("The last value is" + x);  
document.write("</pre>");
```

### 2.2 BÀI TẬP 2

#### 2.2.1 Thời lượng: 10 phút

#### 2.2.2 Mô tả bài toán:

Định nghĩa một function có với tham số có giá trị mặc định.

### 2.2.3 Các bước thực hiện:

```
function myFunction(x, y = 10) {  
    return x + y;  
}  
document.writeln("Result is" + myFunction(5));
```

## 2.3 BÀI TẬP 3

### 2.3.1 Thời lượng: 10 phút

### 2.3.2 Mô tả bài toán:

Dùng method **find** và **findIndex** để tìm giá trị và vị trí của phần tử trong một mảng.

### 2.3.3 Các bước thực hiện:

```
var numbers = [4, 9, 16, 25, 29];  
var first = numbers.find(myFunction);  
// myFunction is test function  
function firstValueFunction(value, index, array) {  
    return value > 18;  
}  
document.writeln("First number over 18 is " +  
firstValueFunction(5));  
  
var firstIndex = numbers.findIndex(myFunction);  
// myFunction is test function  
function firstIndexFunction(value, index, array) {  
    return value > 18;  
}  
document.writeln("First number over 18 has index " +  
firstIndexFunction(5));
```

## 2.4 BÀI TẬP 4

### 2.4.1 Thời lượng: 10 phút

### 2.4.2 Mô tả bài toán:

Định nghĩa một Arrow function.

### 2.4.3 Các bước thực hiện:

```
const myFunction = (x, y) => x * y;  
document.writeln("Result is " + myFunction(5, 5));
```

# React Component Basic

---

## 1 Lý thuyết:

Trong ReactJS, Component được định nghĩa như class hoặc functions.

Component có những đặc tính nổi bật như:

- ❑ Cho phép tách UI thành những phần nhỏ không phụ thuộc.
- ❑ Khả năng tái sử dụng (reusable).

Một React Application được tạo nên bởi nhiều Component khác nhau, mỗi Component thể hiện những chức năng UI khác nhau.

Để tạo một Component, ta cần kế thừa từ React.Component.

```
1 class Welcome extends React.Component {  
2   render() {  
3     return <h1>Hello, {this.props.name}</h1>;  
4   }  
5 }
```

## Rendering React Component

Ví dụ dưới đây thể hiện cách mà React render một Component lên màn hình.

```
1 function Welcome(props) {  
2   return <h1>Hello, {props.name}</h1>;  
3 }  
4  
5 const element = <Welcome name="React" />;  
6 ReactDOM.render(  
7   element,  
8   document.getElementById('root')  
9 );
```

1. Gọi tới `ReactDOM.render()` với element `<Welcome name="React" />`
2. React sẽ khởi tạo `Welcome` element với props `{name: "React"}`
3. `Welcome` component sẽ trả về kết quả là `<h1>Hello, React</h1>` element.
4. **React DOM** update element `<h1>Hello, React</h1>` vào **DOM HTML**.

## 2 Bài Tập:

### 1. BÀI TẬP 1

#### 1.1. Thời lượng: 15 phút

#### 1.2. Mô tả bài toán:

Tạo một Component Welcome sử dụng chuẩn ES6.

#### 1.3. Các bước thực hiện:

Chỉnh file **`src/index.js`** như sau:

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import registerServiceWorker from
'./registerServiceWorker';
const Welcome = (props) =>{ return <h1>Hello,
{props.name}</h1>;}
const element = <Welcome name="Sara" />;
ReactDOM.render( element,
document.getElementById('root'));
registerServiceWorker();
```



## **References**

1. [https://www.w3schools.com/js/js\\_es6.asp](https://www.w3schools.com/js/js_es6.asp)