Js nâng cao: ES6(1 ngày)

# I.ES6 là gì?

ES6 là chữ viết tắt của ECMASCRIPT6. Nó là một tập hợp các kỹ năng nâng cao của Javascript, là một chuẩn mực để chúng ta làm theo.

Các tính năng của ES6:

+Block scoped Contructs Let and Const

+Arrow Function

+Rest Parameter

+Destructuring Assignment

+Default Parameter

+Tempate Literals

+Multi-line String

+Enhanced Object Literals

+Promises

+Classes

…

# II.Cách dùng ES6

1. Block scoped Contructs Let and Const

Block-Scoped là phạm vi trong một khối, nghĩa là nó chỉ hoạt động trong phạm vi được khai báo bởi cặp dấu {}

Bình thường khi khai báo một biến trong một hàm ta thường dùng var. VD trong việc hoán đổi vị trí:

var a=5,b=10;

if(a<b){

var temp=a;

a=b;

b=temp;

}

console.log(a);

console.log(b);

console.log(temp); // Biến vẫn còn có thể dùng được

Kết quả sẽ in ra a=10, b=5, temp=5

Do đó nếu mục đích sử dụng biến temp của chúng chỉ sử dụng bên trong hàm if thì cách khai báo của chúng ta ở đây sẽ lãng phí bộ nhớ của máy tính. Thay vì sử dụng var thì ta sẽ sử dụng let. Let là cách khai báo biến mà nó chỉ được sử dụng bên trong 1 block-scoped. Nếu sử dụng biến mà được khai báo bởi let ở bên ngoài block scoped thì biến đó sẽ gây ra lỗi.

1. Arrow Function

Là cách định nghĩa function bằng dấu mũi tên =>

Bình thường khi khai báo function chúng ta sẽ làm như sau:

function Add(a,b){} // Cách 1

var Add= function (a,b){} // Cách 2

Với ES6 thì ta có cách khai bào hàm mới bằng cách thay function bằng dấu mũi tên:

var Add=(a,b) =>{}

Nếu bên trong hàm chỉ có một câu lệnh thì ta có thể bỏ đi cặp {}

VD:

var Add=(a,b) => console.log(a+b);

Trong trường hợp hàm có một tham số thì ta có thể bỏ đi cặp():

VD:

var Add=a=> console.log(a);

Trong trường hợp không có tham số :

VD:

var Add=()=> console.log(“abc”);

Lưu ý: dấu mũi tên luôn phải cùng dòng với cặp dấu ngoặc () nếu không thì chương trình sẽ khai báo lỗi cú pháp.

1. Rest Parameter

Là cách khai báo một hàm với số lượng tham số không xác định.

Cú pháp: var functionname=(a,b,…,other)=>{}

VD: var functionname=(a,b,…other)=>{

console.log(a);

console.log(b);

console.log(other);

}

functionname(“a”,”b”,”c”,”d”);

Thì kết quả sẽ là

a

b

(2) ["c", "d"]

Các giá trị không xác đinh sẽ được đặt vào mảng như ta đã nhìn thấy ở trên. Để truy cập vào các biến không xác định này thì ta khai báo như một mảng

VD: var functionname=(a,b,…other)=>{

return other; // hàm này sẽ trả về một mảng chứa các phần tử không xác định

}

var arr=functionname(“a”,”b”,”c”,”d”);

// arr khi này là một mảng với 2 phần tử không xác định [“c”,”d”]

console.log(arr[1]);

Thì kết quả sẽ là

d

1. Destructuring Assignment

Là cách tách các phần tử của array hay object thành nhiều biến.

VD:

let date=[10,3,2018];

let [d,m,y]=date;

document.write("Hôm nay là ngày giỗ tổ Hùng Vương "+d+"-"+m+"-"+y);

Kết quả sẽ là: Hôm nay là ngày giỗ tổ Hùng Vương 10-3-2018

Nếu chỉ muốn lấy một phần tử trong mảng đó thì ta chỉ cần bỏ trống các phần tử còn lại.

VD: let [,,y]=date;

console.log(y);

Kêt quả sẽ là 2018

Đó là cách xử lý với array. Còn với Object thì ta làm như sau:

let date={ day: 10,month:3,year:2018}

let {day:d,month:m,year:y}=date;

document.write("Hôm nay là ngày giỗ tổ Hùng Vương "+d+"-"+m+"-"+y);

Kết quả tương tự với array ở trên.

1. Default Parameter

Là cách khai báo giá trị mặc định của tham số truyền vào trong một hàm.

VD:

var df=(myname=" Đạt")=>{

console.log(myname);

}

df();

df(“Chí”);

Kết quả khi ta không truyền vào tham số cho hàm sẽ là Đạt .

Còn với hàm mà ta truyền vào thì vẫn in ra giá trị ta mong muốn. Nó sẽ thay myname từ dạng mặc đinh “Đạt” thành “Chí”.

1. Tempate Literals

Là cách hiển thị trong chuỗi một cách ngắn gọn và bỏ đi các kí tự + và ‘’ giúp code rành mạch và rõ ràng hơn.

Bình thường khi ta thêm một chuỗi vào sẽ như thế này:

VD:

var first = "Trần Quốc";

var last = "Đạt";

var name = 'Ten cua minh la:' + first + ' ' + last + '.';

console.log(name);

Nhưng với ES6 thì ta sẽ viết thế này:

VD:

var first = "Tran Van";

var last = "My";

var name = `Ten cua minh la: ${first} ${last}.`;

console.log(name);

Như trên, ta thấy việc khai báo cộng chuỗi bằng dấu cộng sẽ được thay thế bằng ${} và việc bắt đầu và kết thúc chuỗi của chúng ta sẽ thay thế bằng dấu ``(khác với ‘’ nhé).

1. Multi-line String

Là cách viết giúp ta viết được nhiều dòng trong một chuỗi dài. Bằng cách thay thế ‘’ bằng cặp ``. Giúp việc code trở nên gọn gàng hơn bằng cách loại bỏ dấu + khi ta muốn viết nhiều dòng bằng cách thông thường.

Bình thường khi muốn viết nhiều dòng :

VD:

var content = 'Toi ten la Tran Van My,'

+ 'La mot developer cua cong ty Framgia'

+ 'Ad: 13F Keangnam Landmark 72 Tower, Plot E6, Pham Hung Road, Nam Tu Liem District'

+ 'So thich la: code, xem anime, va ngu'

Còn với ES6:

VD:

var content = `Toi ten la Tran Van My,

La mot developer cua cong ty Framgia,

Ad: 13F Keangnam Landmark 72 Tower, Plot E6, Pham Hung Road, Nam Tu Liem District,

So thich la: code, xem anime, va ngu`

1. Enhanced Object Literals

Là cách mô tả Object một cách ngắn gọn hơn.

**Property value shorthand**: ES6 mặc định gán giá trị cho property khi property có tên match với tên của biến

VD:

function createSinger (name, age, address, salary) {

return {

name,

age,

address,

salary,

}

}

**Computed property key**: Với khả năng này ta có thể set tên của property một cách linh hoạt hơn.

VD:

function createSinger (name, age, address, salary) {

return {

['salaryOf' + name] : salary,

}

}

1. Promises

Đã học ở bài trước.

1. Classes

ES6 hỗ trợ cú pháp của lập trình hướng đối tượng vào giống như các ngôn ngữ lập trình khác

class Employee {

constructor (name, age) {

this.name = name;

this.age = age;

}

setName (name) {

this.name = name;

}

getName () {

return this.name;

}

setAge (age) {

this.age = age;

}

getAge () {

return this.age;

}

};

var emp=new Employee();

emp. setName(“Đạt”);

emp. setAge (“Đạt”);

Ta thấy ở trên để tạo thuộc tính cho đối tượng thì ta không thể tạo trực tiếp như Java. Muốn tạo thuộc tính cho đối tượng ta sẽ tạo thông qua từ khoá contructor như ví dụ trên.

1. Const

Là cách khai báo hằng số giống các ngôn ngữ khác. Khi khai báo 1 hằng. Giá trị của nó không thể bị thay đổi.

Cú pháp: const variablename=value;

VD: const PI=3.14;