**Tìm hiểu về javascript**

**1. Khai báo biến**

- **Biến** trong js có chức năng chứa dữ liệu

- **Biến** có thể khai báo bằng từ khóa **var** hoặc **let**

Ví dụ :

var a = 5; // ta vừa thực hiện khai báo biến x có giá trị là 5

let name = “Duc” // ta vừa thực hiện khai báo biến name có giá trị là “Duc”

- Ta có thể thực hiện các phương thức tính toán trên biến

Ví dụ :

var x = 2+3; // x = 5

var y = ‘Minh’ + ‘ ‘ + ‘Duc’; // x sẽ hiện chuỗi ‘Minh Duc’

var z = 2+3+’5’ // x sẽ hiện chuỗi ‘55’

var t = ‘5’+2+3 // x sẽ hiện chuỗi ‘523’

**2. Function**

- **Function** (hàm) là một khối lệnh dùng để thực hiện một nhiệm vụ cụ thể

- **Function** sẽ được thực hiện khi được gọi

- Cú pháp khai báo function :

*function*<tên function> (tham số 1, tham số 2, tham số 3,…*)* {  
 // code

}

- Cú pháp gọi function :

*<tên function> (giá trị 1, giá trị 2, giá trị 3,…);*

Ví dụ :

function add(a,b){

return a + b ;

}

var x = add(3,4); // x = 7

- Một số lưu ý :

+ Một hàm có thể có tham số hoặc không có

+ Một hàm có thể trả về giá trị hoặc không

+ Một hàm có thể gọi lại chính nó (hàm đệ quy)

- Phạm vi của hàm : Các biến được định nghĩa bên trong một hàm không thể được truy cập từ nơi nào khác bên ngoài hàm, bởi vì biến đó được định nghĩa chỉ trong phạm vi của hàm. Tuy nhiên, một hàm có thể truy cập đến mọi biến và mọi hàm khác trong cùng phạm vi mà nó được định nghĩa .

**3. Object**

- Trong JavaScript, một **object** là một thực thể độc lập, với thuộc tính (**properties**) và phương thức (**methods**)

Ví dụ :

|  |  |  |
| --- | --- | --- |
| **Object** | **Properties** | **Methods** |
| phone  iPhone 6s Plus 32GB | Giá rẻ, chính hãng, nhiều khuyến mãi | phone.name = “iphone6s”  phone.color = “white”  phone.brand = “iphone”  phone.price = “150$” | phone.on()  phone.priceInfo()  phone.takepic()  phone.call() |

- **Object** có thể hiểu như là biến , nhưng mà chứa nhiều giá trị khác nhau

- Ta có thể khai báo 1 object như sau :

var phone = {

name : ‘phone6s’ ,

color : ‘white’ ,

brand : ‘iphone’ ,

price : 150 ,

priceInfo : function() {  
    return "Price : " + this.price + "$";  
  }

} ;

- Muốn truy cập vào 1 thuộc tính của object , ta dùng cú pháp : *objectName.propertyName*

Ví dụ : nếu ta muốn truy cập vào thuộc tính name trong object phone , ta dùng cú pháp : phone.name

- Muốn truy cập vào 1 phương thức của object , ta dùng cú pháp : *objectName.methodName()*

Ví dụ : nếu ta muốn truy cập vào phương thức priceInfo trong object phone , ta dùng cú pháp : phone.priceInfo()

**4. Array**

- **Array** (mảng) là một biến có thế lưu trữ nhiều giá trị

- Cú pháp khai báo mảng :

var <tên array> = [giá trị 1, giá trị 2, ...];

Ví dụ:

var phone = [‘iphone’,’samsung’,’nokia’];

- Truy cập đến các phần tử trong array :

*<tên array> [chỉ số];*

Ví dụ :

var x = phone[0]; // x = ‘iphone’

- Lưu ý : chỉ số các phần tử trong array bằng đầu từ số 0

phone = [‘iphone’,’samsung’,’nokia’];

// phone[0] phone[1] phone[2]

- Một số phương thức của array ( array method )

+ map() : tạo một array mới là kết quả của một function được gọi cho mỗi phần tử trong array ban đầu

+ filter(): tạo một array mới chứa những phần tử của array ban đầu thỏa mãn điều kiện được cho bởi 1 function

+ find() : trả về một phần tử đầu tiên trong array ban đầu thỏa mãn điều kiện được cho bởi 1 function

+ forEach() : gọi function cho mỗi phần từ trong array ban đầu theo thứ tự

- Ngoài ra còn rất nhiều các phương thức khác của array như :



