Javascript cung cấp công cụ để truy cập HTML Dom. Nằm trong html <script src=”...”></script>

Một số hàm cơ bản built-in:

1. alert: bật lên cửa sổ hộp thoại
2. console: in ra những dòng thông báo trong console:
   1. console.log(‘day la mot thong bao’);
3. confirm: bật lên cửa sổ hộp thoại
4. prompt: bật lên cửa sổ hộp thoại
5. setTimeout: thực thi một đoạn code sau một tg nhất định
6. setInterval: thực thi liên tục đoạn code sau một tg nhất định

một số toán tử trong Javascript:

1. toán tử số học-arithmetic
2. toán tử gán-assignment
3. toán tử so sánh- comparison
4. toán tử logic-logical

HTML Dom là tiêu chuẩn của W3c đưa ra cần tuân theo để xây dựng website

Dom: document object model không thuộc ngôn ngữ Javascript. Dom có 3 thành phần: element, attribute( là các thuộc tính) và text. HTML Dom có thể thay đổi các element của một website

1. Element: get id, class, tag, css selector, html collection
   1. getElementById: trả về trực tiếp element
   2. getElementsByClassName: trả về html collection
   3. getElementsByTagName: trả về html collection
   4. querySelector: trả về trực tiếp element
   5. querySelectorAll:trả về node list

Vd: Var name\_variable= Document.getElementById(‘name\_id’);

Console.log({

Element: name\_variable

});

1. Attribute
2. Text

Object: nằm trong dấu ngoặc nhọn.   
 Vd: let person={

firstName: ‘john’,

lastName: ‘Doe’

};

* Truy cập thông qua dấu chấm(.) <ObjectName.propertyName>

Vd: person.firstName

Console.log(person.firstName);

* Truy cập giống như mảng [ ] < ObjectName[ ‘propertyName ‘]>

Vd: console.log(person[‘ firstName’]);

Note: nếu tên thuộc tính( propertyName) có khoảng trắng thì cần đặt trong dấu nháy đơn. Và chỉ được gọi thông qua kiểu giống mảng.

Vd: let address={

‘building no’: 3960,

street: ‘ north 1st street’

};

* Thay đổi giá trị của thuộc tính ( modifying the value of a property)

Vd: person.firstName=’Jane’;

* Adding a new property to an object

Vd: person.age=25;

* Deleting a property of an object

Vd: delete person.age;

* Checking if a property exists ( return true or false)

Vd: console.log( ‘ssn’ in person);

Ouput sẽ hiển thị false do ‘ssn’ không có trong person

Array

* 2 cách tạo array:

Vd cách1: var languages= new Array(

‘PHP’,

‘Java’

);

Vd cách 2: var languages=[

‘PHP’,

‘Java’

];

* forEach: duyệt mảng
* every: kiểm tra mảng thỏa điều kiện hay không( return true or false)
* some: kiểm tra chỉ cần 1 cái thỏa đk
* find: tìm được 1 phần tử trong mảng nếu có thì trả về phần tử đó không thì trả về undefined
* filter:tìm phần tử thỏa đk trả về tất cả các phần tử thỏa mãn đk
* map:bổ sung phần tử hoặc giá trị cho mảng
* reduce:
* Getting the array size

Console.log(languages.lenght);

* Adding an element
* Languages.push(‘C#’); //to the end of an array
* Languages.unshift(‘C++’) // to the begiinning of an array
* Removing an element
  + Const lastelement= languages.pop();//to the end of an array
  + Const firstElement= languages.shift(); // to the beginning of an array
* Check if a value in an array
* Console.log(Array.isArray(languages)); //output la true
* Finding an index of an element in the array
  + Var index= languages.indexOf(‘PHP’);

Console.log(index); // output la 0

Promise

Sync: đồng bộ

Async: bất đồng bộ: setTimeout, setInterval, fetch,XMLHttpRequset, file reading, request animation frame. Sử dụng Callback để xử lý những vấn đề bất đồng bộ này nhưng gặp vấn đề là Callback hell nên mới xuất hiện promise

Có 3 trạng thái:

1. Pendding
2. Fulfilled
3. Rejected

var promise= new Promise (

//excutor

Function(resolve, reject){

//logic

//thành công: resolve()

//thất bại: reject()

}

);

Promise

//khi resolve được gọi

.then(function(){

})

//khi reject được gọi

.catch(function(){

})

.finally(function(){

});