- Các giá trị khi convert sang boolean mang giá trị false: null ; NaN ; undefined ; ‘ ’ ; false. Các giá trị khác khi convert sang boolean mang giá trị true

- Toán tử = là gán giá trị, toán tử == là so sánh giá trị (không so sánh kiểu dữ liệu), toán tư === là so sánh tuyệt đối cả giá trị lẫn kiểu dữ liệu

- Toán tử logical && :

var result = ‘A’ && ‘B’ && NaN && ‘D’;

result mang value cuối cùng của các biểu thức nếu tất cả mang giá trị true, nếu có bất kỳ biểu thức nào có giá trị khác 6 value false bên trên thì result mang giá trị false

- Toán tử logical || :

var result = ‘A’ && ‘B’ && NaN && ‘D’;

result sẽ mang giá trị của “biểu thức khác 6 value false” đầu tiên tìm thấy được, nếu tất cả biểu thức mang giá trị false thì result trả về false

- Chuỗi: javascript string methods

+ cách tạo:

var fullName = ’HoangAnh’; // trả về kiểu dữ liệu string

var fullName = new String(‘HoangAnh’); // trả về kiểu dữ liệu object

+ dùng dấu ‘\’ trước các ký tự đặc biệt như ‘ hay “ ở trong chuỗi để đưa ký tự vào chuỗi (backlash in js)

+ độ dài : fullName.length

+ template string ES6: đưa biến vào trong chuỗi

var fName = ‘Hoang Anh’;

var lName = ‘Huy’;

console.log(`Toi la: ${fName} ${lName}`);// dấu “ ` ” ở dưới phím esc

+ vị trí chuỗi:

// nếu không tìm được vị trí trả giá trị -1

console.log(fullName.indexOf(‘oa’));// trả về vị trí chuỗi đầu tiên tìm đc từ 0

// trả về 2

console.log(fullName.indexOf(‘n’, 3));// bắt đầu tìm ký tự sau ký tự thứ 3

// trả về 6

console.log(fullName.lastIndexOf(‘n’));// trả về vị trí cuối cùng

console.log(fullName.search(‘oa’));// giống indexOf luôn tìm vị trí đầu tiên

// dùng để tìm theo biểu thức chính quy - Regular expression

fullName.include(‘oa’);// giống indexOf nhưng trả về giá trị kiểu boolean

+ cắt chuỗi

console.log(fullName.slice(4,6));// trả về An

+ thay thế:

console.log(fullName.relapce(‘N’, ‘n’));// trả về HoaNgAnh

// chỉ sửa chuỗi đầu tiên tìm thấy, sửa đổi khác thì dùng biểu thức chính quy

+ chuyển tất cả chữ hoa: fullName.toUpperCase()

+ chuyển tất cả chữ thường: fullName.toLowerCase()

+ xóa khoảng trắng đầu cuối, 2 khoảng trắng kề nhau: fullName.trim()

+ cắt 1 chuổi thành array:

var languages = ‘JS, C#, C++, PHP’;

console.log(languages.split(‘, ’));// cắt các ngôn ngữ tách biệt bởi dấu ‘, ’ lưu vào 1 mảng

+ lấy ký tự theo vị trí

console.log(fullName.charAt(2))// trả về ký tự thứ 2: a; nếu truyền về số lớn hơn length trả về 1 chuỗi rỗng kiểu string ‘’

console.log(fullName[3])// trả về ký tự thứ 3: n, nếu số lớn hơn length trả về undefined

- Number:

+ khai báo:

var age = 20;

var otherNumber = new Number(21);// kiểu object

var result = 20 / ‘ABC’;// lấy number chia chuỗi, trả về NaN

console.log(result);// trả về NaN

// kiểm tra có phải kiểu NaN

console.log(result == NaN);// trả về false

console.log(isNaN(result));// trả về true

+ chuyển sang string: age.toString()

console.log(typeof age.toString());// trả về String

+ làm tròn phần số thập phân

var PI = 3.14;

console.log(PI.toFixed());// trả về 3

cosole.log(PI.toFixed(1));// trả về 3.1

- Mảng:

+ khai báo:

var languages = [‘JS’, ‘PHP’, ‘Ruby’];

// có thể chứa tất cả các kiểu dữ liệu cung 1 lúc

var arr = [‘chuoi’, null, NaN, funcion(){ }, {} ];// hợp lệ

var array = new Array(‘JS’, ‘PHP’, ‘Ruby’);// không khuyến khích

console.log(typeof languages);// kiểu dữ liệu mảng luôn trả về 1 object

console.log(Array.isArray(languages));// trả về true: để kiểm tra array với object

+ truy xuất mảng

languages.length : lấy số phần tử

languages[2] = ‘pHp’; đổi phần tử 2 bằng pHp

+ chuyển tất sang string: languages.toString(): chuỗi thành string và thêm dấu “,” vào giữa các phần tử

languages.join(‘-’);// chuyển string nhưng để ngăn cách giữa các phân tử bằng “-”

+ xóa phần tử cuối mảng và trả về phần tử đó:

languages.pop();// trả về phần tử vừa xóa nếu mảng rỗng trả về undefined

+ thêm 1 hoặc nhiều phần tử cuối mảng và trả về độ dài mới của mảng

languages.push(‘Dart’, ‘Java’);// trả về 5

+ xóa 1 phần tử đầu mảng mảng và trả về phần tử đó

languages.shift();// trả về phần tử vừa xóa nếu mảng rỗng trả về undefined

+ thêm 1 hoặc nhiều phần tử đầu mảng, trả về số lượng phần tử

languages.unshift(‘Dart’, ‘Java’);// trả về 5

+ xóa 1 hoặc nhiều phần tử liền nhau ở vị trí bất kỳ

languages.splice(1, 2);// từ vị trí 1 xóa đi 2 phần tử

+ chèn 1 hoặc nhiều phần tử liền nhau ở vị trí bất kỳ (có thể xóa cùng lúc)

languages.splice(1, 0, ‘Dart’);// thêm Dart sau phần tử vị trí thứ 1 và không xóa phần tử nào sau vị trí thứ 1

+ nối 2 mảng

languages.concat(array);// nối array sau languages

+ cắt lấy 1 vài phần tử của mảng

languages.slice(1,2);// trả về PHP: cắt tử vị trí số 1 đến trước vị trị số

languges.slice(1);// cắt từ phần từ 1 đến hết mảng, truyền vào 0 để copy mảng, truyền số âm lấy phần từ cuối mảng

- Funciton: có 3 loại function

+ Declaration function// có thể hosting: gọi trước khi định nghĩa

function showMessage() {

};// phải đặt tên

+ Expression function// có thể đặt tên function mục đích dễ hiểu

var showMessage2 = function() {

};

setTimeout(function() {

});

var myObject = {

myFunction: function() {}

}

+ Arrow function