Javascript

# DateTime

* 4 cách tạo 1 object Dateime:

new Date()

new Date(year, month, day, hours, minutes, seconds, milliseconds)

new Date(milliseconds)

new Date(date string)

* UTC

new Date("2014-01-01") => 31/12/2013 theo giờ UTC

Hàm

.getDay(): Thứ

# Function

* Lấy element thông qua selector
* Id: document.getElementById(id)

* Tâp hợp tất cả các element con trong 1 element cha
* element.childNodes[0]
* Lấy data, text trong 1 element
* .data

Muốn set lại data của textNode thì dùng

const textNode = element.childNodes[0]

textNode.data = text;

* String
* text[5]: lấy ký tự thứ 5 của text
* text.length: độ dài của text
* substring(): Lấy ra một chuỗi con từ 1 chuỗi cha
  + str.substring(1, 3): Lấy ra một chuỗi từ index 1 đến 2
  + str.substring(2): Lấy ra 1 chuỗi từ index 2 trở đi
* .split('.'): cắt chuỗi tại dấu “.”
* .pop(): lấy phần tử cuối cùng trong array
* String(): convert -> string
* .toLowerCase(): in thường
* .toUpperCase(): in hoa
* .repeat(n): lặp lại chuỗi str n lần - str.repeat(n)

Unicode:

* .charCodeAt(): lấy mã unicode của một chữ
* String.fromCharCode(char): chuyển số thành chữ
* Array
* Truy cập array.key hoặc array[‘key’]. Ưu tiên xét trong ngoặc [] trước
* [array[1], array[2], array[0]]: khởi tạo
* arr.pop(): Lấy phần tử cuối cùng trong mảng
* reverse(): đảo ngược 1 chuỗi
* sort(): sắp xếp

list\_code.sort(function(a, b){ return a - b}) : Sắp xếp mảng tăng dần

* .indexOf(1) : nums.indexOf(1) – tìm index của số “1” trong mảng nums
* .map(): return mảng mới với điều kiện đã cho - const map1 = array1.map(x => x \* 2);

str\_ara.map(v => max = Math.max(max, v.length));

* [...to\_string]: clone 1 mảng
* .splice(2, 1): xóa 1 phần tử ở vị trí index = 2
* URL
* document.URL: lấy URL.
* document.URL.href: lấy href current

Tip:

* Xóa element empty trong array

var split\_href = href.split('/').filter(item => item);

# Syntax

* For

for ( var i=1; i<=10; i++ ){

list\_btn.push(`<button>${i}</button>`)

}

* For in

var list\_btn = [‘1’,’2’, ‘3’]

for ( element in list\_btn ){

console.log(list\_btn[element])

}

“element” là key, “list\_btn[element]” là value

# Method

* setInterval(): thực hiện 1 hành động gì đấy được lặp lại trong 1 khoảng thời gian nhất định.
* setTimeout: chờ một khoảng thời gian thì thực hiện
* clearTimeout: hủy bỏ lệnh setTimeout

# Math

* Math.random() \* 20: random number từ 0->19. Math.floor(Math.random() \* 20) + 1
* Math.floor: làm tròn số (5.95->5; 5.05->5; 5->5; -5.05->-6
* Math.abs(): trả về giá trị tuyệt đối của 1 số
* Math.max(): tìm giá trị lớn nhất trong mảng - Math.max(max, temp) .

# ES6

* (/a...b/).test(str) : Kiểm tra chuỗi “str” có ký tự “a” và “b” phân cách nhau bởi 3 ký tự ở giữa hay không
* Array
* .includes(1): nums.includes(1) – kiểm tra trong mảng có số “1” không. !nums.includes(1): phủ định

# Regular Expressions

<https://www.w3schools.com/jsref/jsref_obj_regexp.asp>

<https://regexr.com/>

* Khởi tạo:
* let re = /ab+c/;
* Cú pháp:
* /[^p]/g

Example:

- Kiểm tra trong câu ký tự đầu tiên có viết hoa không: /^([A-Z])/

- Kiểm tra trong câu ký tự cuối cùng có phải là dấu chấm không: /[.]$/

# Other

* NaN: Khi check == NaN thì dùng !NaN
* Lấy style của một element:

var element = document.querySelector(".line-shadow-white").style.background = 'url("file:///E:/Growme-core/SourceCode/static/js\_create\_thumb\_templates/img/slide.png") repeat scroll 0% 0% / cover padding-box border-box, url("file:///E:/Growme-core/SourceCode/static/js\_create\_thumb\_templates/img/slide-line.png") repeat scroll 0% 0% / cover padding-box border-box, rgba(192, 192, 192, 0.08) none repeat scroll 0% 0% / cover padding-box border-box';

* Tạo một dãy button bằng js

var element\_list\_btn = document.getElementById("list-btn")

var element\_node = element\_list\_btn.childNodes[0]

var list\_btn = []

for ( var i=1; i<=10; i++ ){

list\_btn.push(`<button>${i}</button>`)

}

for ( element in list\_btn ){

element\_list\_btn.innerHTML += list\_btn[element]

}

* A variable using a user-defined name

var var\_name = 'abcd'

var n = 120;

this[var\_name] = n;

console.log(this[var\_name]);

* Toán tử bậc 3:

nums[0] > nums[2] ? nums[0] : nums[2]

* Swap – giữ được vị trí các phần tử khác trong arr

function swap(arr){

[arr[0], arr[arr.length - 1]] = [arr[arr.length - 1], arr[0]];

return arr

}