**Lý do cần có biến**: --> *trả lời câu hỏi không có biến sẽ ntn?*

- Tái sử dụng code( tránh trùng lặp code ), maintain dễ hơn, sửa 1 chỗ dùng nhiều chỗ

- Đặt tên tường minh, dễ hiểu, dễ đọc, dễ maintain = định danh

- Biến có thể biến thiên

- Biến không thể bắt đầu bằng số

-------------------------------------------------------------------------------------------------------------

**Cách dùng JS**: trong thẻ body file html thêm dòng:

<script src="index.js"></script>

In ra Tab Console: console.log(...);

- document.write(...): in ra màn hình

Khai báo biến: var <Tên biến>;

Khởi tạo biến: var <Tên biến> = value;

**Các kiểu dữ liệu**: KDL Biến chính là KDL của value:

console.log(typeof true) -> KDL boolean

- log ra KDL của biến

var a=10;

console.log(typeof a) -> KDL number

- Boolean(true,false)

- string (nháy đơn or nháy kép)

- Number: cả số nguyên và số thực

- undefined: duy nhất value undefined

- null: duy nhất value null

- object: lưu trữ nhiều value: {...} hoặc [...]

**Các hàm thông dụng trong JS:**

- Hàm alert(...): sổ ra màn hình, có nút OK

- Hàm confirm(...): trả về true hoặc false, có nút OK và nút hủy

vd: var isConfirm = confirm('Bạn đã đủ 18 tuổi chưa?');

console.log(isConfirm);

- Hàm prompt: hiển thị ô để input dữ liệu

vd: var nameUser = prompt('Xin bạn nhập tên:');

console.log(nameUser);

- Hàm parseInt(..): chuyển chuỗi sang số nguyên

-------------------------------------------------------------------------------------------------------------

**Toán tử số học:** + - \* / % \*\*(mũ)

gán =, x++, +=,\*=

So sánh: >, <, >=, <=

So sánh nghiêm ngặt (value+type): ===

So sánh lỏng lẻo(value): ==

logical: && , ||

// Tất cả những giá trị được xem xét false

// null

// undefined

// ''

// false

// NAN

// 0

// Còn lại là true

&&:// Nhớ: Đi từ trái sang phải sẽ đánh giá từng value một,

// nếu gặp value nào đó tương đương false, nó sẽ trả value đó cho kết quả

// còn không sẽ trả về value cuối cùng

||:// Nhớ: Đi từ trái sang phải sẽ đánh giá từng value một,

// nếu gặp value nào đó tương đương true, nó sẽ trả value đó cho kết quả

// còn không sẽ trả về value cuối cùng

!: phủ định

Lưu ý về thứ tự ưu tiên khi tính toán

\* nhân chia trước, cộng trừ sau

control flow (if, else)

\*

\*/

var isCondition = false;

if (isCondition) {

console.log('đi cắm trại');

} else {

console.log('ở nhà thôi');

}

-------------------------------------------------------------------------------------------------------------

JavaScript chỉ có number, không có int hay float,double

**Chuyển string sang number:**

- dùng hàm Number(..): x= Number(x): giữ nguyên phần thập phân( nên dùng)

- dùng hàm parseInt(..): x= parseInt(x): chuyển sang số nguyên, bỏ phần thập phân

- dùng hàm parseFloat(..)

- dùng x = +x

// Tìm hiểu : nháy đơn và nháy kép khác gì nhau

**Nối chuỗi**: dùng dấu + đối với nháy đơn và nháy kép:

var nameInfo = prompt('Nhập tên bạn')

var nameWelcome = 'Welcome ' + nameInfo;

console.log(nameWelcome);

Với dấu ` `:

- Dùng được biểu thức js : $(...)

- Hiểu được ký tự xuống dòng enter (nháy đơn và nháy kép không hiểu xuống dòng, phải dùng \n)

--> text có xuống dòng nên dùng ` `

var nameWelcome2 = `Welcome ${nameInfo}`;

console.log(nameWelcome2);

Ký tự $(): trả về 1 value: vd $(5 \* 5) > 25

--> đưa vào $(..) biến or biểu thức

\*Lưu ý nhầm lẫn giữa nối chuỗi và phép cộng:

console.log(5 + 2 + ''truong'') -> 7truong

**\*LOOP: VÒNG LẶP:**

- document.write(...): in ra màn hình(string)

- for

- while

- do while

-------------------------------------------------------------------------------------------------------------

**Scope:**

- Biến Global

- Biến Local

- từ khóa var: func scope, cho phép khai báo lại

- từ khóa let: block scope, không cho phép khai báo lại-> nên dùng let

- từ khóa const: tương tự let, không thay đổi được value

-------------------------------------------------------------------------------------------------------------

**MẢNG**

Nếu dùng biến-> 100 biến = 400 bytes

***Dùng mảng sẽ tiết kiệm bộ nhớ hơn, thao tác dễ dàng hơn***

let <tên mảng> =[

, , ,

];

**• Các thông tin liên quan đến mảng:**

khi log mảng sẽ tự động đánh số phần tử từ 0 đến (length -1) và xuất ra length

tên mảng.length : độ dài mảng hay giới hạn mảng

tên mảng[stt] để truy xuất đến value phần tử tại stt đó

**• Tìm ra index (vị trí) phần tử với value bất kỳ (vd Jonas):**

tên mảng.indexOf('Jonas')

**• Thêm phần tử vào mảng:**

-Thêm vào cuối:

tên mảng.push(value cần thêm); // có thể push 1 lần nhiều value

let biến = mảng 1.push(,,,)

log (biến) -> log ra chiều dài mảng 1 sau push

-Thêm vào đầu:

tên mảng.unshift(value cần thêm)

**• Xóa phần tử khỏi màng:**

tên mảng.pop(); // xóa phần tử cuối

tên mảng.shift(); // xóa phần tử đầu

- Xóa bất kỳ vị trí nào

+ Tìm ra index -> xóa:

tên mảng.splice(vị trí cần thay đổi, số phần tử muốn xóa tính từ vị trí cần thay đổi, value thay thế(có hoặc không))

**• Cách truy cập phần tử mảng Dynamic:**

Dùng vòng lặp -> phần tử mảng thứ i:

for(let i=0;i< tên mảng.length; i++){

console.log(tên mảng[i]);

}

**• Convert string to array và ngược lại:**

- string to array: sử dụng split: vd:

let strConvert = `nguyen,phan,nhut,truong`;

let arrConvert = strConvert.split(`,`);

-> tạo thành mảng gồm 4 phần tử

- array to string: sử dụng join:

let joinResult = arrConvert.join(`\_`); // nên dùng join

-> tạo thành mảng nối bởi \_

let toStringResult = arrConvert.toString();