**==================== lesson 1 ====================**

một số hàm built-in (là hàm được js xây dựng sẵn)

    1. alert

    2. console

    3. confirm

    4. prompt

    5. set timeout

    6. set interval

**==================== lesson 2 ====================**

/\*\* js có 6 giá trị khi dịch sang boolean là False

 \* 0

 \* false

 \* '' hoặc ""

 \* NaN

 \* null

 \* undefined

 \*/

 \* kiểu dữ liệu trong js

 \* 1. kiểu dữ liệu nguyên thủy

 \*      - number

 \*      - string

 \*      - boolean

 \*      - undefined

 \*      - null

 \*      - symbol

 \* 2. kiểu dữ liệu phức tạp

 \*      - function

 \*      - object

 \*/

 giới thiệu về toán tử

    1. toán tử số học - arithmetic

    2. toán tử gán - assignment

    3. toán tử so sánh - comparision

    4. toán tử logic - logical

 \*/

 toán tử         ví dụ       tương đương

    =           x = y        x = y

    +=          x += y       x = x + y

    -=          x -= y       x = x - y

    \*=          x \*= y       x = x \* y

    /=          x /= y       x = x / y

    \*\*=         x \*\*= y      x = x \*\* y

    ++          x ++         x = x + 1 // tăng gtri biến lên 1 và trả về gtri trước khi tăng

    ++          ++ x         x = x + 1 // tăng gtri biến lên 1 và trả về gtri sau khi tăng

    --          x --         x = x - 1

    --          -- x         x = x - 1

 toán tử         ví dụ       tương đương

    ++          x ++         x = x + 1 // tăng gtri biến lên 1 và trả về gtri trước khi tăng

    ++          ++ x         x = x + 1 // tăng gtri biến lên 1 và trả về gtri sau khi tăng

    --          x --         x = x - 1

    --          -- x         x = x - 1

 \* toán tử logic

 \* 1. &&  and

 \* 2. ||  or

 \* 3. !   not

console.log(x + y);

    - nếu x và y cùng là số thì dấu + là dấu + toán học

    - nếu 1 trong 2 không phải là số thì dấu + là toán tử nối chuỗi

 toán tử

 ==             -->             bằng

 !=             -->             khác

 >             -->              lớn hơn

 <             -->              bé hơn

 >=             -->             lớn hơn hoặc bằng

 <=             -->             bé hơn hoặc bằng

/\*\* toán tử so sánh phần 2 (so sánh tuyệt đối)

 \* === (bằng tuyệt đối)

 \* !== (khác tuyệt đối)

 \*/

**==================== lesson 3 ====================**

/\*\*

 \* các loại hàm

 \* 1. declaration function

 \* 2. expression function

 \* 3. arrow function

 \*/

/\*

1. hàm?

    - một khối mã

    - làm 1 việc cụ thể

2. loại hàm

    - built-in

    - tự định nghĩa

3. tính chất

    - không thực thi khi định nghĩa

    - sẽ thực thi khi được gọi

    - có thể nhận tham số

    - có thể trả về 1 giá trị

4. tạo hàm đầu tiên

\*/

/\*

một số điều cần biết về func

1. khi func đặt trùng tên => func trước sẽ bị func sau cùng ghi đè

2. khai báo biến trong hàm

3. định nghĩa hàm trong hàm

\*/

// return

// hàm mà không return thì mặc định trả về undefined

/\*

tham số trong hàm

1. tham số?

    - định nghĩa?

    - kiểu dữ liệu?

    - tính private?

    - 1 tham số

    - nhiều tham số

2. truyền tham số

    - 1 tham số

    - nhiều tham số

3. arguments?

    - đối tượng arguments

    - giới thiệu vòng for of

\*/

**==================== lesson 4 ====================**

/\*

chuỗi (string) trong js

1. tạo chuỗi

    - các cách tạo chuỗi

    - nên dùng cách nào? lí do?

    - kiểm tra data type

2. một số case sử dụng backslash (\)

3. xem độ dài chuỗi

4. chú ý độ dài khi viết code

5. template string es6

\*/

// cách 1: dùng 1 cặp nháy đơn or nháy kép

// cách 2: dùng new String()

// => nên dùng cách 1

// làm việc với chuỗi

var myString = 'Phi Hoang Khong';

// 1. length

// console.log(myString.length);

// 2. find index

// console.log(myString.indexOf('Hoang')); // không có thì trả về -1, nếu có thì trả về vị trí đầu tiên tìm được

// console.log(myString.indexOf('Hoang', 6)); // tìm bắt đầu từ vị trí 6

// console.log(myString.lastIndexOf('Hoang')); // tìm lần cuối xuất hiện

// console.log(myString.search('Hoang')); // chỉ tìm được lần đầu xuất hiện

// 3. cut string

// console.log(myString.slice(4, 9)); // start - end

// console.log(myString.slice(4)); // start

// console.log(myString.slice(0)); // start

// console.log(myString.slice(-5, -1)); // start - end

// 4. replace

// console.log(myString.replace('Hoang', 'hihi')); // thay thế 1 lần chuỗi đầu tiên

// console.log(myString.replace(/Hoang/g, 'hihi')); // thay thế tất cả chuỗi

// 5. convert to upper case

// console.log(myString.toUpperCase(myString));

// 6. convert to lower case

// console.log(myString.toLowerCase(myString));

// 7. trim

// console.log(myString.trim()); // bỏ khoảng trắng ở đầu và cuối

// console.log(myString.trim().length); // bỏ khoảng trắng ở đầu và cuối

// 8. split

// var language = 'js, php, java';

// console.log(language.split(', ')); // chuyển string thành array

// var str = 'php';

// console.log(str.split('')); // chuyển string thành mảng chứa từng chữ cái

// 9. get a character by index

// const str = 'phi khong';

// console.log(str.charAt(2)); // lấy kí tự ở vị trí được truyền, trường hợp index không có thì trả về chuỗi rỗng

// console.log(str[2]); // lấy kí tự ở vị trí được truyền, trường hợp index không có thì trả về undefined

**==================== lesson 5 ====================**

/\*

kiểu số (number)

1. tạo giá trị number

    - các cách tạo

    - dùng cách nào? tại sao?

    - kiểm tra data type

2. làm việc với number

    - to string

    - to fixed

\*/

**==================== lesson 6 ====================**

/\*

kiểu mảng (array)

1. tạo mảng

    - cách tạo

    - dùng cách nào? tại sao?

    - kiểm tra data type

2. truy xuất mảng

    - độ dài mảng

    - lấy phần tử theo index

\*/

/\*

làm việc với array

1. to string

2. join

3. pop

4. push

5. shift

6. unshift

7. splicing

8. concat

9. slicing

\*/

**==================== lesson 7 ====================**

/\*

Math obj

    - Math.PI

    - Math.round()

    - Math.abs()

    - Math.ceil()

    - Math.floor()

    - Math.random()

    - Math.min()

    - Math.max()

\*/

/\*

object prototype (nguyên liệu tạo nên ngôi nhà)

1. prototype là gì?

2. sử dụng khi nào?

\*/

**==================== lesson 9 ====================**

/\*

đệ quy:

    1. xác định điểm dừng

    2. logic handle => tạo ra điểm dừng

\*/

/\* bài toán

input:

    ['a', 'b', 'c', 'b', 'a']

output: (xóa các phần tử trùng lặp)

    ['a', 'b', 'c']

\*/

/\*

vòng lặp

1. for - vòng lặp với điều kiện đúng

2. for/in - lặp qua key của đối tượng

3. for/of - lặp qua value của đối tượng

4. while - lặp khi điều kiện đúng

5. do/while - lặp ít nhất 1 lần, sau đó lặp khi điều kiện đúng

\*/

// for/in => dùng lấy key của đối tượng

// for/of => lấy ra phần tử 1 mảng or từng chữ cái 1 chuỗi (không dùng được với obj)

**==================== lesson 10 ====================**

/\*

Array methods: 7 methods dưới đây đều có tham số truyền vào là func

    1. forEach()

    2. every()

    3. some()

    4. find()

    5. filter()

    6. map()

    7. reduce()

\*/

/\*

đệ quy:

    1. xác định điểm dừng

    2. logic handle => tạo ra điểm dừng

\*/