JS (JavaScript)

1. JS có thể làm gì ?

-Tạo hiệu ứng động cho trang web, game, vv

-Xử lí sự kiện (click chuột, nhập dữ liệu, vv)

-Giao tiếp với máy chủ API (Application Programming Interface)

-Xây dựng ứng dụng game, web, ứng dụng di động, ứng dụng desktop

-Làm việc và quản lí dữ liệu

1. Lời khuyên

Cố hiểu hơn cố nhớ code

Thực hành sau mỗi bài học

1. Hàm

-Hàm hiện cửa sổ thông báo – Hàm alert

-Hàm xác nhận – hàm confirm

-Hàm nhập liệu – hàm prompt

-Hàm thực thi 1 đoạn code sau 1 khoảng thời gian nhất định – hàm setTimeout

-Hàm thực thi 1 đoạn code nhiều lần sau 1 khoảng thời gian nhất định – setInterval

-Tạo 1 biến – hàm var

1 biến về độ tuổi – var gae

1. Comment

-Dùng để ghi chú

-Vô hiệu hóa code

-Comment có 2 loại : comment 1 dòng, comment nhiều dòng

1. Toán tử

-Số học : + cộng; - trừ; \* nhân; / chia; % chia lấy dư; \*\* lũy thừa; ++ tăng 1 đơn vị; -- giảm 1 đơn vị

VD: var x=1;

var y=2;

console.log(x+y); //3

Prefix & Postfix (tiền tố, hậu tố)

-Tiền tố (++a, --a): thực hiện phép toán trước rồi trả giá trị thay đổi

-Hậu tố (a++, a--): trả về giá trị ban đầu rồi mới thực hiện phép toán

Gán: = gán

+=, -=, \*=, /=, %= gán kết hợp

VD: var a=5;

a +=5 // a = a+5 =10

Chuỗi:

+ dùng để nối chuỗi

VD: var name = “hien”;

Console.log(“hello ” + name); // hello hien

So sánh:

== so sánh bằng (không xét kiểu)

=== so sánh bằng (xét kiểu)

!=, !== khác

>, <, >=, <= lớn, bé, lớn hơn hoặc bằng, bé hơn hoặc bằng

VD: var a = 1;

var b= 1;

Console.log(a == b); // true

1. Kiểu dữ liệu

-Boolean: Là kiểu dữ liệu có 2 giá trị true, false

-Dùng để biểu diễn trạng thái logic, thường xuất hiện trong điều kiện, vòng lặp và kiểm tra giá trị

VD: let isLoggedIn = false;

if (isLoggedIn) {

console.log("Chào mừng!");

} else {

console.log("Vui lòng đăng nhập.");

}

-Number: đại diện cho các giá trị số, bao gồm số nguyên và số thực

VD: var a = 10;

-String: chuỗi ký tự đc đặt dấu nháy hoặc dấu kép

VD: var fullname = “hien”;

-Undefined: biến đc khai báo nhưng chưa gán giá trị

VD: var age; //undenfined

-Null: biến giá trị rỗng, không có gì

VD: var a = null;

Object:

-khi cần định nghĩa key-value

-kiểu dữ liệu phức tạp, chứa nhiều giá trị dưới dạng cặp key-value

Array: khi cần định nghĩa danh sách thì dùng

1. Làm việc với chuỗi

-Có 2 cách tạo chuỗi: nháy đơn hoặc kép

-Kiểm tra data type: console.log(typeof...)

-Nếu muốn thêm dấu nháy trong ngoặc thì dùng \

-Template string ES6

VD: var firstname = “Ha”;

Var lastname = “Hien”;

Console.log(“toi la: ${firstname} + ${lastname}”) ;

Phương thức lấy thông tin chuỗi

-Length: trả về độ dài chuỗi

VD: “hello”.length // 5

-charAt(index): lấy ký tự tại vị trí

VD: “hello”.charAt(3) // l

-At(index): giống charAt nhưng hỗ trợ chỉ số âm

VD: “hello”.at(-1) // o

Phương thức cắt chuỗi

slice(start, end): trích xuất phần chuỗi

VD: “JavaScript”.slice(4, 10) // Script

-Substring(start, end): tương tự slice nhưng ko nhận chỉ số âm

-Substr(start, length): trích chuỗi theo độ dài

Phương thức biến đổi chuỗi

-toUpperCase(): chuyển sang chữ hoa

VD: “hello”.toUpperCaser() // HELLO

-toLowerCase(): chuyển sang chữ thường

-trim(): loại bỏ khoảng trắng đầu/cuối

-padStart(length, char): thêm ký tự ở đầu

VD: “5”.padStart(3, “0”) // “005”

-repeat(n): lặp lại chuỗi

VD: “hien”.repeat(3) // hienhienhien

Phương thức tìm kiếm chuỗi

-Includes(substring): kiểm tra chuỗi con

VD: “JavaScript”.includes(“Script”) // true

-indexOf(substring): vị trí đầu tiên của chuỗi con

VD: var mystring = “hoc JS tại JS F8!”

Console.log(mystring.indexOf(“JS”, 6)) // 11 (nếu có nhiều hơn 1 từ giống nhau)

-startsWith(prefix), endsWith(suffix): kiểm trả đầu/cuối chuỗi

Phương thức thay thế và tách chuỗi

-replace(old, new): thay thế chuỗi con đầu tiên

VD: “I love JS”.replace(“JS”, “JavaScript”) // “I love JavaScript”

“học JS tại JS”.replace(/JS/g, “JavaScript”) // học JavaScript tại JavaScript (nếu có nhiều hơn 1 từ giống nhau muốn thay thế)

-replaceAll(old, new): thay thế tất cả chuỗi con

-split(delimiter): tách chuỗi thành mảng

VD: “a, b, c”.split(“,”) // [“a”, “b”, “c”]

Lưu ý: chuỗi là kiểu dữ liệu nguyên thủy

Chuỗi ko thể thay đổi từng ký tự trực tiếp

1. Làm việc với kiểu số

Cách tạo: VD: var PI =3.14;

Làm tròn: console.log(PI.toFixed()) // 3

1. Làm việc với mảng

Cách tạo: VD: var languages = [

‘JavaScript’,

‘PHP’,

‘Ruby’,

];

* 1. Cấu trúc cơ bản

-length: trả về số lượng phần tử trong mảng

VD: let arr = [1, 2, 3];

arr.length // 3

* 1. Thêm và xóa phần tử

-push(): thêm phần tử vào cuối mảng

-pop(): xóa phần tử cuối cùng

-unshift(): thêm phần tử và đầu mảng

-shift(): xóa phần tử đầu tiên

* 1. Tìm kiếm và kiểm tra

-indexOf(): tìm vị trí của 1 mảng

-includes(): kiểm tra phần tử có tồn tại trong mảng

-lasIndexof(): trả về vị trí cuối cùng của phần tử

-find(): tìm phần tử đầu tiên thỏa điều kiện

-findIndex(): tìm vị trí phần tử đầu tiên thỏa mãn điều kiện

-some(): kiểm trả nếu ít nhất một phần tử thỏa mãn

-every(): kiểm tra nếu tất cả phần tử thỏa mãn

* 1. Lặp và biến đổi

-forEach(): lặp qua từng phần tử và thực hiện hàm

-map(): tọa mảng mới bằng cách biến đổi từng phần tử

-filter(): tạo mảng mới với các phần tử thảo mãn điều kiện

-reduce(): gộp các phần tử thành 1 giá trị duy nhất

-flat(): làm phẳng mảng lồng nhau

-flatMap(): ấp dụng map rồi làm phẳng 1 cấp độ

* 1. Sắp xếp và chuỗi hóa

-sort(): sắp xếp mảng (mặc định là theo chuỗi)

-reverse(): đảo ngược thứ tự phần tử

-join(): nối các phần tử thành chuỗi với dấu phân cách

-toString(): chuyển mảng thành chỗi phân cách bằng dấu phẩy

* 1. Các phương thức tiện ích khác

-concat(): nối 2 hoặc nhiều mảng

-fill(): gán giá trị tất cả phần tử

-copyWithin(): sap chép 1 phần tử đến vị trí khác trong mảng