**JS1**

***Member: Nguyễn Thị Bích Ni***

1. **JS là gì, các thuật ngữ liên quan**

* **JavaScript là một ngôn ngữ kịch bản của WEB (script language), được chạy bởi công cụ JavaScript của trình duyệt web.**
* Mã lệnh của JS được **tích hợp và nhúng trong HTML.**
* Hoạt động trên đa trình duyệt và đa thiết bị;
* Đặc tả ECMAScript cung cấp các tiêu chuẩn mà JavaScript thực hiện.
* JS là ngôn ngữ bất đồng bộ.

1. **Các cách output dữ liệu**

* **Inner:** document.getElementById("demo").innerHTML = 5 + 6;
* **Write:** document.write(5 + 6);
* **Alert:** window.alert(5 + 6);
* **Console.log:** console.log(5 + 6);
* **Print:** <button onclick="window.print()">Print this page</button>

1. **Comment JS**

// comment…

1. **Biến, kiểu dữ liệu**

* ***Biến: Var, let, Const***
  + **Var/Let, Const** khác nhau về Scope, Hoisting. Var được hỗ trợ Hoisting
  + **Let/Const (ES6) :** chỉ có phạm vi trong các khối code như: if else, loop, { }…
  + **Const:** Khi định nghĩa biến và không gán lại biến đó.
* ***Kiểu dữ liệu:***

1. ***Kiểu dữ liệu nguyên thuỷ***

+ Number: var length = 16;

+ String: var lastName = "Ni";

+ Boolean: var isSuccess = true;

+ Underfine: var x;

+ Null: var isNull = null;

+ Symbol: var id = Symbol('id') // unique

1. ***Kiểu dữ liệu phức tạp***

+ Function: var x = function(){...}

+ Object: var x = {firstName:"John", lastName:"Doe"};

1. **Phép gán**

= , += , -= , \*= , /= , %=

*Ví dụ: x\*=y <=> x=x\*y*

1. **Toán tử**

* **Toán tử số học:**

*+ 🡪 Cộng;*

*- 🡪 Trừ;*

*\* 🡪 Nhân;*

*\*\* 🡪 Luỹ thừa;*

*/ 🡪 Chia;*

*% 🡪 Chia lấy số dư;*

*++ 🡪 Tăng 1 giá trị số;* var a = 5; a++ //6

*- - 🡪 Giảm 1 giá trị số;*

***\* Toán tử ++ -- với tiền tố và hậu tố***

* **Tiền tố:**

Var a = 6;

Var output = ++a //7

* + - **Giải thích:** Việc 1: +1 cho a, a = a + 1 => a = 7;

Việc 2: Trả về a sau khi a + 1;

* **Hậu tố:**

Var a = 6;

Var output = a++ //6

Console.log(a) //7

* + - **Giải thích:** Việc 1: `a copy`, `a copy` = 6

Việc 2: Cộng 1 của a, a = a + 1, => a = 7;

Việc 3: Trả về `a copy`

1. **Function**

* **Hàm?**

+ Là khối mã

+ Làm một công việc cụ thể

* **Loại hàm?**

+ Built-in

+ Tự định nghĩa

* Declare Function:

*function showMessage(){*

*Console.log(“Declare”);*

*}*

* Express Function:

*var showMessage2 = function(){*

*Console.log(“Express”);*

*}*

🡪 *Declare có thể gọi trước khi định nghĩa.*

* **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

* **Arguments**
* function write(){
* var myString = "";
* for(var params of *arguments*){
* myString += `${params} - `;
* }
* console.log(myString);
* }
* write("Log1","Log2","Log3") // Log1 - Log2 - Log3 -

1. **Lệnh điều kiện**

* ***6 giá trị trả về là false***

+ 0

+ false

+ ‘ ‘ – “ “

+ undefined

+ NaN

+ null

🡪 còn lại sẽ là True

* ***Lệnh rẽ nhánh Switch***

*Switch(a){*

*Case 1: console.log(“So 1”);*

*Break;*

*Case 2: console.log(“So 2”);*

*Break;*

*Default: console.log(“Khong ton tai”);*

*}*

1. **Vòng lặp**

* For: Lặp với điều kiện đúng
* For/in: Lặp qua key của đối tượng
* For/of: Lặp qua value của đối tượng
* While: Lặp khi điều kiện đúng
* Do/while: Lặp ít nhất 1 lần, sau đó lặp khi điều kiện đúng

1. **String và các method liên quan**

|  |  |
| --- | --- |
| Code | Result |
| \' | **‘** |
| \" | **“** |
| \\ | **\** |
| \b | Backspace |
| \f | Form Feed |
| \n | New Line |
| \t | Horizontal Tabulator |
| \v | Vertical Tabulator |
| \r | Carriage Return |

* **Method**

Var myString = “Ni Nguyen”

**+ Length**: console.log(myString.length) //9

+ **Find index**: console.log(myString.indexOf(a)) //-1

console.log(myString.indexOf(N,1)) //3

console.log(myString.lastIndexOf(n)) //8

console.log(myString.search(n)) //chỉ truyền được 1 đối số

**+ Cut String:**

* slice(*start*, *end*)
* slice(*start)*
* substring(*start*, *end*)
* substr(*start*, *length*)

**+ Replace:**

var txt = str.replace("Microsoft","W3Schools");

**+ Split**

var str = "a,b,c,d,e,f";

var arr = str.split(",");

document.getElementById("demo").innerHTML = arr[0]; //a

+ **Concat**

var text1 = "Hello";  
var text2 = "World";  
var text3 = text1.concat(" ", text2); //Hello World

**+ Trim**

var str = "       Hello World!        ";  
console.log(str.trim()); // Hello World!

**+ CharAt:**

var str = "HELLO WORLD";  
str.charAt(0);            // returns H

**+ Upper and Lower Case**

var text1 = "Hello World!";       // String  
var text2 = text1.toUpperCase(); // HELLO WORLD!

1. **Number và các method liên quan**

* ***isNaN()***

var x = 100 / "Apple";  // x will be NaN (Not a Number)

* toFixed()

var x = 9.656;  
x.toFixed(0); //9

* toString()

var x = 9.656;  
x.toString(); // “9.656”

* The Number() method

Number(true);          // returns 1  
Number(false);         // returns 0  
Number("10");          // returns 10  
Number(" 10  ");       // returns 10  
Number("10.33");       // returns 10.33  
Number("10,33");       // returns NaN

* The parseInt() method

parseInt("10");         // returns 10  
parseInt("10.33");      // returns 10  
parseInt("10 20 30");   // returns 10  
parseInt("10 years");   // returns 10  
parseInt("years 10");   // returns NaN

* The parseFloat() method

parseFloat("10");        // returns 10  
parseFloat("10.33");     // returns 10.33  
parseFloat("10 years");  // returns 10  
parseFloat("years 10");  // returns NaN

1. **Array và các method liên quan**

* **Tạo mảng:**

var points = new Array(40, 100, 1, 5, 25, 10); // Bad  
var points = [40, 100, 1, 5, 25, 10];          // Good

* **Kiểm tra Array:**

Console.log(Array.isArray(a))

* **Method:**

**+ ToString**

Console.log(array.toString());

Console.log(array.join(‘’));

+ **Pop:** xoá phần tử cuối mảng, return phần tử đã xoá

+ **Push:** thêm 1 or nhiều phần tử vào cuối mảng, return mảng mới.

Console.log(array.push(“ni”,”na”)

+ **Shift:** xoá phần tử đầu mảng, return phần tử đã xoá

+ **Unshift:** thêm 1 or nhiều phần tử vào đầu mảng, return mảng mới.

Console.log(array.push(“ni”,”na”)

**+ Splice:** arr.splice(vị trí, số ptử xoá, giá trị thêm vào)

**+ Concat:** arr.concat(arr2)

**+ Slice:** cắt lấy 1 vài element

arr.slice(start, end)

1. **Date và các method liên quan**

Var date = new Date();

* Get Method

Var year = date.getFullYear();

Var month = date.getMonth(); //0-11

Var hour = date.getHours();

* Set Method

Var setDate = date.setDate(15);

Var setHour = date.setHours(0);

1. **Object Math**

* Math.PI
* Math.round(4.9); làm tròn số gần nhất    // returns 5
* Math.ceil(4.4); làm tròn lên    // returns 5
* Math.floor(4.9); làm tròn xuống   // returns 4
* Math.abs(-4.7);     // returns 4.7
* Math.random();  return số thập phân nhỏ hơn 1
* Math.min(0, 150, 30, 20, -8, -200);  // returns -200
* Math.max(0, 150, 30, 20, -8, -200);  // returns 150

1. **Boolean và các toán tử Logic, So sánh**

**Toán tử so sánh:** == ; != ; > ; < ; >= ; <= ; === ; !==

**Kiểu dữ liệu boolean:** var isSuccess = true;

**Toán tử Logical:** && ; || ; !

**InnerText và textContent**

* <h1 *id*="heading">
* <span>New</span>
* <span>Heading</span>
* </h1>
* <script>
* var headingNode = document.getElementById("heading");
* headingNode.innerText = "New Heading"
* console.log(headingNode.innerText); // New Heading
* console.log(headingNode.textContent) //
* New
* Heading
* //
* </script>

**+ InnerText:** Chỉ lấy nội dung text

**+ textContent:** Lấy toàn bộ khoảng trắng

**JS6**

1. **localStorage**

LocalStorage giúp website lưu trữ dữ liệu vĩnh viễn trên trình duyệt. Trừ khi người dùng xóa cache hoặc làm các hành động như cài lại trình duyệt.

**Syntax:** window.localStorage

**SAVING data to localStorage:** localStorage.setItem("key", "value");

**READING data:** var lastname = localStorage.getItem("key");

**REMOVING data:** localStorage.removeItem("key");

**REMOVING ALL the localStorage:** localStorage.clear();

1. **sessionStorage**

**sessionStorage** khá giống với localStorage. Vì chúng đều thuộc về web storage API.

**Ví dụ:** sessionStorage.setItem('key', 'value');

* + **Sự khác nhau của localStorage và sessionStorage**
* Khi close tab hay close browser thì data ở localStorage vẫn tồn tại, và chỉ bị mất khi user xoá cache hoặc clear web data. Còn đối với sessionStorage thì data sẽ bị mất ngay khi close tab hoặc close browser.
* **Về phạm vi**: sessionStorage: giới hạn trong một cửa sổ hoăc thẻ của trình duyệt. Một trang web được mở trong hai thẻ của cùng một trình duyệt cũng không thể truy xuất dữ liệu lẫn nhau. Như vậy, khi bạn đóng trang web thì dữ liệu lưu trong sessionStorage hiện tại cũng bị xóa. Còn localStorage: có thể truy xuất lẫn nhau giữa các cửa sổ trình duyệt. Dữ liệu sẽ được lưu trữ không giới hạn thời gian.

1. **Cookie**

* **localStorage** chỉ access được trên browser client; còn **cookies** thì có thể access được ở **browser client và cả phía server** (khi tạo một http request thì cookies của browser sẽ được attach vào header Cookie, từ đó phía server có thể parse header này và get được data cookie).
* **cookies** có thời gian hết hạn expires, sau thời gian này thì cookies sẽ biến mất khỏi browser.
* **Cookie** thường được dùng để đăng nhập, lưu vài giá trị cơ bản và đơn giản. Cookie không bị giới hạn bởi độ dài dung lượng hay ký tự.

**Ví dụ hàm set-cookie**

document.cookie = "username=John Doe; expires=Thu, 18 Dec 2013 12:00:00 UTC; path=/";