DOM

Document object model

BOM – window

Document

Navigators

Frames

Location

History

function myFunc() {

// I am inside the scope of function myFunc()

}

// Now I am outside the scope of myFunc()

function myFunction() {

var data = "hi";

function function2() {

var msg = "hello";

console.log(data); // Call to outside variable

}

console.log(msg); // Call to msg variable inside the function2()

}

myFunction();

**Primitive types:**

All the primitive data types are like this, immutable

null

undefined

boolean

number

string

NaN

Symbol - new in ES6

And

Object – is not primitive , and also its mutable

**JAVASCRIPT IS DYNAMICALLY TYPED LANGUAGE –**

Means data types are “dynamic”. Changes according to value we re assign

1. String

let firstName = "alex";

console.log(typeof(firstName));

1. Number

let age = 20;

console.log(typeof(age)); // number

const PI = 3.14;

console.log(typeof(PI));

task:

check for infinity & nan using type of – task

1. Boolean

let isValid = false;

console.log(typeof(isValid));

1. Null

let nothing = null;

console.log(typeOf); // object

1. Undefined

let name;

console.log(name); // undefined

1. Symbols: new in js and its distinct

let mySymbol = Symbol("welcome");

console.log(typeOf(mySymbol)); // symbol

console.log(Symbol("welcome”) === Symbol("welcome")); o/p? is it false ? why?

//=== checks for value and content

**When Working with Strings Please note this!!!**

console.log("is it possible! “Like this” to be given");

console.log("is it possible! \“Like this\” to be given");

or replace using single quotes.

## TEMPLATE LITERALS(TEMPLATE STRINGS) – new in ES6

let name = "my name";

console.log(`My name is ${name}.`);

## we have to make use of back ticks.

console.log(`hi welcome , ${name}`);

hi welcome , rinu

## any string concatenated with int or float will also beocme String

console.log(name+ ”age ” + 5);

## Arithmetic expression:

console.log(3 \*\* 2); // output: 9

CHECK what happens:

let hexStringNumber = '0x1E';

console.log(+hexStringNumber); /

## [String comparison](https://javascript.info/comparison" \l "string-comparison):

alert( 'Z' > 'A' ); // true

alert( 'Glow' > 'Gle' ); // true

## checks char by char

If both strings end at the same length, then they are equal. Otherwise, the longer string is greater.

## comparing String with char or int:

alert( '2' > 1 );

alert(‘01’==1);

alert( true == 1 ); // true

alert( false == 0 ); // tru0065

## task:

let a = 0;

alert( Boolean(a) ); // false

let b = "0";

alert( Boolean(b) ); // true

alert(a == b); // true!

## How a and b are same ?

## Compare zero and null

## o/p of “2” >”12”

## For a strict equality check ===

## Alert(undefined===null)

## For a non strict equality check ==

## Alert(undefined==null)

JS can be used in different ways apart from console of browser:

1. Using embedded

<script> js codes </script> inside head tag

1. <script src="/examples/javascript/ myscript.js"></script>

Create a myscript.js file in external location

<button type="button" onclick="document.write('Hello World!')">Click Me</button>

Inside <script> tag

document.getElementById("result").innerHTML = “hi hello”;

**EVENT HANDLING:**

**We have different types of events:**

**1. Mouse**

**2. Keyboard**

**3. Form**

**4. Document / Window**

**Mouse Event:**

**=================**

**Onclick:**

<a href="#" onclick="alert('You have clicked a link!');">Click Me</a>

<button type="button" onclick="alert('You have clicked a button!');">Click Me</button>

<button type="button" id="myBtn">Click Me</button>

<script>

function sayHello()

{

alert('Hello World!');

}

document.getElementById("myBtn").onclick = sayHello;

</script>

**Oncontextmenu: works when we right click**

<a href="#" oncontextmenu="alert('You have clicked a link!');">Click Me</a>

**Onmouseover : works when we place mouse over the content**

<a href="#" onmouseover="alert('You have placed mouse pointer over a link!');">Place Mouse Over</a>

**KeyBoard Events:**

<input type="text" onkeydown="alert('You have pressed a key inside text input!')">

<input type="text" onkeyup="alert('You have released a key inside text input!')">

<input type="text" onkeypress="alert('You have pressed a key inside text input!')">

**Form Events:**

is fired when a form control receive or loses focus or when the user modify a form control value such as by typing text in a text input, select any option in a select box etc.

**onfocus :**

 focus event occurs when the user gives focus to an element on a web page.

<input type="text" onfocus="highlightInput(this)">

<script>

function highlightInput(this)

{

this.style.background = "yellow";

}

</script>

Task:

Need a text box , when text box gets focus shud change the colour of font to blue , style should be italics and its content what ever gets types should be visible in alert box after u type

Onblur event: when element losses focus

Onchange: when choosing options in radio button or drop down

<select onchange="alert('You have changed the selection!');"> <option>Select</option> <option>Male</option> <option>Female</option> </select>

Onsubmit:

<form action="some action" method="post" onsubmit="alert('call function to validate!');">

<label>First Name:</label> <input type="text" name="first-name" required> <input type="submit" value="Submit"> </form>

**Window Event:**

<body onload="window.alert('Page is loaded successfully!');">

Onunload is not supported by most browsers, not recommened to be used

**Onresize event:**

**String methods:**

**Splice - adds in specified pos of an array**

**Search**

**Replace with /i and /ig ----- call kjsbcjasbcbc Call jhsguyas call**

**g- for replace all , /i for case insensitive**

**charAt**

**split**

**substr**

**Numbers:**

**Parsing:**

console.log(parseInt("3.14")); // 3

console.log(parseInt("50px")); // 50

console.log(parseInt("12pt")); // 12

console.log(parseInt("0xFF", 16)); // 255

console.log(parseInt("20 years")); // 20

console.log(parseInt("Year 2048")); // NaN

console.log(parseInt("10 12 2018")); // 10

**find?**

**converting int to string, float to string , string to other types**

var counts = ["one", "two", "three", "four", "five"];

var reversed = counts.reverse(); alert(counts); // Outputs: five,four,three,two,one alert(reversed);

counts.sort();

**find use of**

Math.max.apply(null, array);

Math.min.apply(null, array);

**Looping:**

for(variable in object) {  
    // Code to be executed  
}

var time={“12”,”1”,”2”};

**for(t in time){**

**console.log(“time is ”+t);**

**}**

**// An object with some properties**

**var person = {"name": "Clark", "surname": "Kent", "age": "36"};**

**// Loop through all the properties in the object**

**for(var prop in person) {**

**document.write("<p>" + prop + " = " + person[prop] + "</p>");**

**}**

Note :

for-in loop should not be used to iterate over an array where the index order is important. You should better use a for loop with a numeric index.

ES6 introduces a new for-of loop which allows us to iterate over [arrays](https://www.tutorialrepublic.com/javascript-tutorial/javascript-arrays.php) or other iterable objects (e.g. [strings](https://www.tutorialrepublic.com/javascript-tutorial/javascript-strings.php)) very easily. Also, the code inside the loop is executed for each element of the iterable object.

Note :

**For…of will not work for iterating object**

**JS Objects:**

var person = { name: "Peter", age: 28, gender: "Male", displayName: function() { alert(this.name); } };

document.write(person.name)

document.write(person[“name”])

**deleting a property in an object**

delete person.age;

var dept = { name: "It", loc: “chn”, did: "10" };

var d = dept; // Assign person variable to a new variable

d.name = "Harry";

document.write(dept.name); // Prints: Harry document.write(d.name); // Prints: Harry