**(Master these-> Object and Browser Event to become good js developer)**

**DATATYPES IN JAVASCRIPT**

PRIMITIIVE DATATYPES

1. number - 2, 99, 9.5 (in js – int, float, double all comes under number only + NaN (not a number) also comes under number datatype)
2. String – “harshan”, ‘Vivek’
3. Boolean- true, false
4. null – means nothing, when we do not want to assign any value use null, it is explicitly given by programmer.
5. undefined- undefined means till now the value is not defined there is a chance that in future it will be defined. For ex- undefined is assigned to our variable implicitly, if we do not initialize it.

Note – NaN(Not a Number is not a datatype in javascript rather it is a special value of number datatype and it comes when programmer wants to perform arithmetic operation but arithmetic operation is not possible with those values).

For ex- console.log( “major” - ”captain” ) -> NaN

NON PRIMITIVE DATATYPES

1. array

* represented by [ ]
* collection of homogenous as well as heterogeneous elements
* for ex let arr = [10,2.5,false,”MajorMohit”,null,undefined,NaN]

1. object

* represented by {}
* for ex - let heroes = {name: ”Captain Manoj Pandey”

medal: “Ashok Chakra”

}

**Variables -** variable are named memory location which we use to store our data.

* In javascript variable are declared using -> var, let, const (these are keywords which are used in variable declaration)
* In javascript variables are not strictly-typed, they are dynamically typed means we do not have to mention datatype while declaring the variable, on the basis of data we store in the variable, the variable datatype is decided.

For ex let a

console.log(a) //undefined

const b= 52

console.log(b) //52

Note- b variable is declared using const keyword, so now it cannot be re-initialized.

Note- we should avoid declaration using var.

**Operators**

1. Assignment operator (=) -> this operator is used to assign value to the variable.

For ex - let a = 30

modulus operator – gives remainder. For ex 9/2 = 1

2) Arithmetic operator ( + , - , \* , / ,% , \*\* , ++ , --)

Exponential operator - number raised to the power. for ex 2^3 = 8

Division operator – use to divide the number. For ex- 9/2 = 4.5 (in js it will give floating point answer, even though both the values are of int type because here int and float both comes under number datatype)

Comparison operator – used to compare two values. That first value is <, >, <= , >= , != , == , === than the second value. It will always return the answer in Boolean datatype .

For ex-> console.log(20>=30) // false

== equal to

In equal to operator only values will be checked not datatype.

For ex console.log(300==”300”) //true because values are equal

=== strict equality operator

In strict equality operator both datatype and values will be checked. Both should be equal only then the answer will be true.

For ex console.log(300===”300”) // false because datatype is different

1. Logical operator(&& , || , !)

&&- Logical AND

Both should be true, then only the answer will be true otherwise false

let value1 = true

let value2 = false

For ex- console.log( value1 && value2) //false

!! Logical OR

If any one would be true, answer will be true

For ex- console.log(value1 || value2) //true

! Logical NOT

Inverts the boolean value.

For ex - console.log(!value1) //false

**DOM (Document Object Model)**

What is DOM?

* Document Object Model is a hierarchical tree like structure of our webpage, where every node(element) is an object. Like document itself is an object, elements are objects, text inside elements are object.

BOM (Browser Object Model)

(complete the diagram)

Why it is needed?

* DOM connects our webpage and programming language, so if we want to modify our webpage dynamically or in real-time, it can only happen with the help of programming language and our programming language manipulate or target the DOM objects only. Because modifying DOM means modifying webpage.(as DOM is nothing but structure of our webpage).
* We can (create, access, update, delete)DOM objects through java script.



In this example on clicking button the text inside h1 will change. So here we are

manipulating DOM.

**DOM Methods**

To select a particular element

* document.getElementsById(“id”) - it will return the single element(datatype-Object)
* document.getElementsByClassName(“className”) – it will return the collection of elements(datatype -HTML collection of Objects)
* document.getElementsByTagName(“tagname”)- it will return collection of elements(datatype- HTML collection of Objects)
* document.querySelector(“id,class,tag”)- it will return single element.(datatype-Object)

Note- if there will multiple elements first element will be returned.

* document.querySelectorAll(“id,class,tag”)- it will return collection of Objects(datatype- Nodelist)

Difference between HTML Collection and NodeList

* HTML collection do not support forEach,map
* NodeList support forEach but map is not supported

Array Methods

To check a particular variable is Array or not

* Array.isArray(variableName) 🡪 it will return Boolean value true(means array) or false(not array)

To convert a variable into Array

* Array.from(VariableName) 🡪 return a new Array

Difference between innerHTML and innerText and textContent

innerText will only print text inside that element, and if any tag is present inside that tag it will ignore the tag.

For ex-   <h3>Brigadier Saurabh singh <span style="display: none;">Shekhawat</span></h3>

*let* a =document.querySelector("h3")

console.log(a.innerText)

Output- Brigadier Saurabh singh

innerHTML will print the text + tag (if any tag is present inside the element).

console.log(a.innerHTML)

Output- Brigadier Saurabh singh <span style="display:none;">Shekhawat</span>

textContent will print the text(even if that text is hidden by using css properties).

console.log(a.textContent)

Output- Brigadier Saurabh singh Shekhawat

**Event Propagation**

When one element event gets triggered along with that it also triggered all its super element’s event is called event propagation.

Two types

Event bubbling - in which event trigger start from inside. (like bubble)

Event capturing – in which event trigger start from outside. (like capturing of fort happens in war, first outside and then inside)

addEventListener(“event”, “function”{

}, true/false)

Third parameter will decide Event bubbling will happen or Event capturing will happen. By default if you not mention third argument it will be false by default.

false – means event bubbling.

true – means event capturing.