# Javascript Output

## Javascript display possibilities.

* InnerHTML : Writing into an HTML element using InnerHTML.
  + - Ex: document.getElementById(‘demo’).InnerHTML = ‘Text’;
    - This will display the output in a HTML element ( Div or P tag)
* document.write : Writing into an HTML element using document.write.
  + - Ex: document.write(‘Text’).
    - This will display the out in the web browser.
* Alert : Writing into an alert box by using alert method.
  + - Ex: alert(‘text’);
    - This will display on page load before DOM content will loaded.
* Console.log : Writing in to the content in the browser by using console.log
  + - Ex: console.log(‘text’);
    - This will display the content in the console.
    - Right click -> Inspect-> console.

# Javascript statements

* Ex:
  + Let x, y, z;
  + Let x = 10;
  + Let y = 10;
  + document.getElementById(‘demo’).InnerHTML = ‘Text’;

# Javascript commenting

Ex:

Single comment use //

Multi comment use /\* xxxxxxxxxxxxxxxxxxxxxxxxxx \*/

# Javascript variables

We can define variables in 3 types

* let
* var
* const

Ex 1:

var x = 10;

var y = 20;

var z = x + y;

document.write(‘The sum is’ + z);

Ex 2:

let x = 10;

let y = 20;

let z = x + y;

document.write(‘The sum is’ + z);

Ex 3:

const x = 10;

const y = 20;

const z = x + y;

document.write(‘The sum is’ + z);

## Difference between var vs let vs const.

By using var, we can re-declare the variable.

By using var, we can re-assign a value to the variable.

var fName = ‘Welcome’;

xxxxxx

xxxxxx

xxxxxx

var fName = ‘Welcome India’;

By using let, we can’t re-declare the variable.

By using let, we can re-assign the variable.

let fName = ‘Welcome’;

xxxxxx

xxxxxx

xxxxxx

let fName = ‘Welcome India’; 

fName = ‘Welcome India’; 

By using const, we can’t re-declare the variable.

By using const, we can’t re-assign the variable.

const fName = ‘Welcome’;

xxxxxx

xxxxxx

xxxxxx

const fName = ‘Welcome India’; 

fName = ‘Welcome India’; 

# Javascript Arithmetic Operators.

## Addition +

Ex: let a , b;

a=10; b=20;

z = a+b;

Subtraction -

Ex: let a , b;

a=10; b=20;

z = a - b;

Multiplication (\*) -

Ex: let a , b;

a=10; b=20;

z = a \* b;

Division (/) - (will give quotient)

Ex: let a , b;

a=20; b=10;

z = a/ b; - Output is 2

Module (%) - (will give reminder)

Ex: let a , b;

a=20; b=10;

z = a % b; - Output is 0.

Increment ( ++) -

Ex: let a = 2

a++; - Output is 3.

Decrement ( --) -

Ex: let a = 2

a--; - Output is 1.

# Javascript Logical Operators.

## Greaterthan ( > )

Ex: let a = 2;

let b = 6;

if(b > a)

{

return true;

}

## Lessthan ( < )

Ex: let a = 2;

let b = 6;

if(a < b)

{

return true;

}

## Greaterthanorequalto ( >= )

Ex: let a = 2;

let b = 2;

if(b >= a)

{

return true;

}

## Lessthanorequalto ( <= )

Ex: let a = 2;

let b = 2;

if(b <= a)

{

return true;

}

# Javascript assignment Operators.

|  |  |  |
| --- | --- | --- |
| Operator | Usage | Actual functionality. |
| = | x = y | Assign y value to x. |
| += | x += y | x = x +y;  let x = 5;  let y = 10;  x = x + y;  x += y  output : 15; |
| -= | x -= y | x = x - y; |
| \*= | x \*= y | x = x \* y; |
| /= | x /= y | x = x / y; |
| %= | x %= y | x = x % y; |

# Javascript datatypes

* String
* Number
* Boolean
* BigInt
* Undefined
* Null
* Object
  + Object
  + Array.

String:

Ex: let fName = ‘Emplicare’;

let fName = “New emplicar”;

Number:

Ex:

var a = 10;

var a = 10.50;

Note: Integers can only accurate upto 15 digits.

Boolean:

Ex:

let is\_active = true;

let is\_active = false;

BigInt.

Let bigint\_var = BigInt(1234567890123456789012345);

Note: A bigint variable can hold more than 15 digits.

Undefined :

Without defining the variable try to execute the variable.

Ex:

document.write(fName);

typeOf(fName) -> Undefined.

fName variable not defined.

Null:

Null value means nothing, mean it should not contain either ‘0’ or ‘blank’.

let department = NULL;

Object :

A object contains one or more values with key & value pair.

Let person= {fName:’Emplicar’, ‘lName’ : ‘Software solutions’};

How to get value from the object.

person. fName : Output -> Emplicar.

Array.

A array contains one or more values with index based or key based.

Index based is the default array

Key based is the associate array.

Default array:

Ex:

let student = [‘Ben’, ‘Can’, ‘Ram’];

How to get the value from the default array.

Student[2] : Output -> Ram.

Associative array:

Ex:

let users = [‘internal’=>’Staff’, ‘external’=>’Students’];

How to get the value from the associative array.

users [‘internal’] = Staff.

# Javascript functions.

What is the function?

To perform / execute a single task is called function.

Functions are two types

1. Void
2. Non-void

Void functions should not return any value.

Ex:

function displayMessage(msg)

{

document.write(msg);

}

displayMessage(‘My new message’);

Ex: Non-void.

function addingtwoNumbers(a , b)

{

return a + b;

}

let result = addingtwoNumbers(25,10);

displayMessage(result);

function addingtwoNumberswithtrycatch(a , b)

{

try{

return a + b;

}catch(err){

document.write(err.message());

}

}

addingtwoNumberswithtrycatch (25,10);

# Objects.

How to define a object.

let employees = {};

How to add values to an object.

employees.fName = ‘Emplicar’;

employees.lName = ‘Software Solutions’;

employees.cources = [‘PHP’, ‘Java’];

employees.fee = 10;

How to get the value from an object.

document.write(employes.lName);