**DAY 1 ASSIGNMENT**

**1) console.log()**

Mainly used to log(print) the output to the console. We can put any type inside the log(), be it a string, array, object, boolean etc.

**console.error()**

Used to log error message to the console. Useful in testing of code. By default the error message will be highlighted with red color.

**console.warn()**

Used to log warning message to the console. By default the warning message will be highlighted with yellow color.

**console.clear()**

Used to clear the console. The console will be cleared, in case of Chrome a simple overlayed text will be printed like : ‘Console was cleared’ while in firefox no message is returned.

**console.time() and console.timeEnd()**

Whenever we want to know the amount of time spend by a block or a function, we can make use of the time() and timeEnd() methods provided by the javascript console object. They take a label which must be same, and the code inside can be anything( function, object, simple console).

**console.table()**

This method allows us to generate a table inside a console. The input must be an array or an object which will be shown as a table.

**console.count()**

This method is used to count the number that the function hit by this counting method.

**2)**

**var** variables can be updated and re-declared within its scope; **let** variables can be updated but not re-declared; **const** variables can neither be updated nor re-declared. They are all hoisted to the top of their scope. But while **var** variables are initialized with undefined , **let and const** variables are not initialized.

Example-🡪

1.var a=2;

2.{

Let a=2;

}

3. const a=4;

**3)**

JavaScript variables can hold many **data types**: numbers, strings, objects and more:

var length = 16;                               // Number  
var lastName = "Johnson";                      // String  
var x = {firstName:"John", lastName:"Doe"};    // Object

//boolean

var x = 5;  
var y = 5;  
var z = 6;  
(x == y)       // Returns true  
(x == z)       // Returns false