JAVASCRIPT DAY – 1 ASSIGNMENT

1. Explore the various methods in console function. Explain them?

Ans:

console object: The console is an object which provides access to the browser debugging console. By clicking f12 in browser, we can open a developer console.

The various methods in console function are,

(i) log ()
(ii) error ()
(iii) warn ()
(iv) time () and timeEnd ()
(v) table ()

(I). console.log (): This method is log(print) the output to the console. We can put any type inside the log (), be it a string, array, object, Boolean, etc.,

Example: console.log('Madhu'); //string

```
console.log (1); //numeric

console.log(true); //boolean

console.log(null);

console.log ([1,2,3,4,5]); //array inside log

console.log ({a:1, b:2, c:3}); //object inside log
```

(ii). console. error (): This method is used to log error message to the console. It is useful in testing of code. The error message will be highlighted with red color.

Example: console.error('This is an error message');

(iii). console.warn(): This method is used to log warning message to the console. By default, the warning message will be highlighted with yellow color.

Example: console.warn('This is warning message');

(iv). console.time() and console.timeEnd(): If 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.

Example: console.time('Time Taken');

```
console.log('fun1');

console.log('fun2');

console.timeEnd('Time Taken');
```

(v). 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.

Example: console.table({‘a’:1, ‘b’:2});

2. Write a difference between var, let and const with code example?

Ans: We can define variables using keywords, like var, let and const.

Example: var a=10; let b=20; const PI=3.14;

var: The scope of a variable defined with the keyword “var” is limited to the “function” within which it is defined. If it is defined outside any function, the scope of the variable is global. **“var is ‘Function Scoped’”.**

let: The scope of a variable defined with the keyword “let” or “const” is limited to the “block” defined by curly braces {}. **“let” and “const” are “block scoped”.**

const: The scope of a variable defined with the keyword “const” is limited to the block defined by curly braces. However, if a variable is defined with keyword const, it can be reassigned.

Example for var:

```
{
var a=10;
console.log(a);
} //blk1
{
a++;
console.log(a);
}//blk2
```

In the above example, since we are using the keyword var to define the variable a, the scope of a is limited to the function within which it is defined. Since a is not defined within any function, the scope of the variable a is global, which means that a is recognized within block 2

In effect if a variable is defined with keyword var, JavaScript does not recognize the {} as the scope delimiter. Instead the variable must be enclosed within a “function” to limit its scope to that function.

Example for let:

```
{
Let a=10;
Console.log(a);
}//blk1
{
a++;
console.log(a);
} blk2
```

when you run the code above you will get an error, variable a not recognized in block2. This is because we have defined the variable a using the keyword let, which limits the scope of variable a to the code block within which it was defined.

Example for const:

```
{
Const PI=3.14;
Console.log (PI);
}blk 1

{
Console.log (PI);
}//blk2
```

that const does NOT mean that the value is fixed and immutable. This is a common misunderstanding amongst many JavaScript developers, and they incorrectly mentioned that a value defined by the const keyword is immutable (i.e. it cannot be changed).

3. Write a brief intro on available datatypes in JavaScript?

Ans:

JavaScript has seven types. Types are valuing that JavaScript can have. Below is a list of data types that JavaScript can have:

1. Number
2. String
3. Boolean

4. Undefined
5. Null
6. Object
7. Symbol

There are two kinds of datatypes they are primitive and non-primitive datatypes.
Letus discuss about it,

Numbers: A number of data type can be an integer, a floating-point value, an exponential value, a 'NaN' or a 'Infinity'.
Example:
var a=250;
var b=25.5;
var c=10e4;

String: The string data type in JavaScript can be any group of characters enclosed by a single or double-quotes or by backticks.
Example:
var s1=" string";
var s2='string';

Boolean: The boolean data type has only two values, true and false. It is mostly used to check a logical condition. Thus, Booleans are logical data types which can be used for comparison of two variables or to check a condition. The true and false implies a 'yes' for 'true' and a 'no' for 'false in some places when we check a condition or the existence of a variable or a value.
Example: typeof(true)
Typeof(false)

Undefined: Undefined data type means a variable that is not defined. The variable is declared but doesn't contain any value.
Example: var a;
Console.log(a);

Null: The null in JavaScript is a data type that is represented by only one value, the 'null' itself. A null value means no value.
Example: var a=null;
Console.log(a);

Symbol: The 'symbol' data type is new in es6. It is one of the new features of es6. The symbol data type defines a property of an object which is private to the object. It refers to the 'key' of the key-value pair of an object.

Object: Let's create an object literal. An object in JavaScript contains key-value pairs in its address. When we refer to obj1, we are actually referring to the address in memory which contains the value {a: 5, b: 6}, instead of the value {a: 5, b: 6} directly.