CLIENT SIDE SCRIPTING

THEORIES TO WRITE IN PRACTICAL BOOK

Practical No. 01

JavaScript:

- JavaScript is a scripting language (A lightweight programming language).
- A core technology of the Web alongside HTML and CSS
- 99% of websites use JavaScript on the Client side for webpage behavior.
- It is designed to add interactivity to HTML pages.
- It is an interpreted language (it executes without preliminary compilation).
- It is a weakly typed language. [M1]

How JS runs on Browser?

- Every language needs a tool to be converted into machine code.
- Web browsers have a dedicated JavaScript engine that executes the client code.
- These engines are responsible for converting JS code into machine code so that the computer executes it successfully.

How to write and run JS?

- For small purposes, like short temporary codes, we can use Console window in Inspect Tab on browser. But it is never recommended.
- We can use <script> tag to embed JS in HTML. Write this tag just before the <body> ends. It is ok to use it for small tasks.
- For writing more lengthy codes, we must create another file with extension “.js”.
- Link this file using “src” attribute of <script> tag.
  Example: <script src=”script.js”></script>

Variables in JavaScript:

- The rule for identifier (variable’s name) is as same as of previous languages -> C, C++ and Java.
- Since it is a weakly typed language (see M1), there’s no need to specify datatype while declaration.
- Three keywords to declare variables :
  - var
  - let
  - const

Keyword	Redeclaration	Redefinition	Scope
var	Yes	Yes	function
let	No	Yes	block
const	No	No	block

CLIENT SIDE SCRIPTING

THEORIES TO WRITE IN PRACTICAL BOOK

Operators in JavaScript:

- Arithmetic operators
  1) Addition +
  2) Subtraction -
  3) Multiplication *
  4) Division /
  5) Modulus %
  6) Exponentiation **
  7) Increment ++
  8) Decrement --
- Assignment operators
  - = += -= *= **= %= /=
- Logical operators
  - Logical AND &&
  - Logical OR ||
  - Logical NOT !
- Bitwise operators
  - Bitwise AND & Bitwise OR | Bitwise NOT !
- String operator
  - Concatenation +
- Comparison / Relational operator
  - < > != == >= <=
- Ternary operator/Conditional operator:
  - ?:

Defing an OBJECT using literal:

```
const person = {

     name: 'John' ,

     age:25,

};
```

Defing an OBJECT using method:

```
var person = Object.create({name: 'John', age:25});
```

GETTER: A function defined under an object used to access a particular property of an object.

SETTER: A function defined under an object used to set/redefine/change a property of an object.

NOTE: Though we can directly access and change the properties of an object, getter and setter are used to have more control over accessing and changing properties.

- An object property is a name, a value, and a set of attributes, the value may be replaced by one or two methods, known as a getter and a setter.

- Properties defined by getters and setters are sometimes known as accessor properties to distinguish them from data properties that have a simple value.