Web Technology-II

Javascript(JS):

Javascript is a light weighted, object-oriented programming language that defines the behaviour of web pages. It is usually embedded directly into HTML pages. It can be used for both client-side and server-side scripting.

Features of JavaScript: ***[Important for 5 marks]***

1. **Light-weighted Scripting Language:**
   JavaScript is a lightweight scripting language because it is made for data handling in the browser only.
2. **Dynamic Typing:**
   JavaScript supports dynamic typing which means types of the variable are defined based on the stored value. For example, if you declare a variable **x** then you can store either a string or a Number type value or an array or an object.
3. **Object-oriented Programming support:**
   Javascript supports Encapsulation and inheritance concepts, which are the features of object-oriented programming.
4. **Platform Independent:**
   JavaScript is platform-independent or we can say it is portable, which simply means that you can simply write the script once and run it anywhere and anytime.
5. **Interpreted Language:**
   JavaScript is an interpreted language which means the script written inside javascript is processed line by line. These Scripts are interpreted by a JavaScript interpreter which is a built-in component of the Web browser.
6. **Client-side Validations:**
   This feature allows the programmers to set the validation rule for each website while entering the data in the form available on the website.

Methods/Ways to add Javascript to an HTML page: ***[Imp. 5 marks]***

We can add Javascript code to an HTML page in 3 different ways:

1. Internal Javascript
2. External Javascript
3. Inline Javascript

Internal Javascript:

In internal javascript, the javascript code is written between <script> and </script> tags. The <script> tag can either be placed in <head> or <body> tag within the HTML page.

Example:

```
<html>
<head>
    <title>Internal JavaScript</title>
</head>
<body>
   <script>
       document.write("Internal Javascript");
   </script>
</body>
</html>
```

External Javascript:

In external javascript, the javascript codes are written in a separate file with an extension .js and are linked to an HTML document using <script> tag.

Example: main.html

```
<html>
<head>
    <title>External JavaScript</title>
</head>
<body>
    <button onclick="message()">Click</button>
    <script src="test.js"></script>
</body>
</html>
```

test.js

```
function message()
{
    alert('External Javascript');
}
```

Inline Javascript:
Javascript code can also be inserted directly inside the HTML tag using special tag events such as onclick, onload, on keypress, etc.

```
<html>
<head>
    <title>Inline JavaScript</title>
</head>
<body>
      <button    onclick="alert('Welcome    to    JavaScript Class')">Click Here</button>

</body>
</html>
```

Client-side Scripting:
Client-side scripting is a method of running scripts or codes on the client end (i.e., browser) and does not interact with the server at the time of processing. It focuses on making the interfaces of websites and web applications. The different types of scripting languages such as HTML, CSS, Javascript, etc. are used for client-side scripting.

Server-side Scripting:
Server-side scripting is a method of running scripts or codes on the back end (i.e., server) and involves a server for processing the user requests. It focuses on data accessing methods, error handling and keeps control over the data. The programming languages such as PHP, R, Ruby, Python, ASP.net, etc are used for server-side scripting.

Differences between Client-side and Server-side Scripting: *[Important for S.Q.]*

Client-Side Scripting	Server-Side Scripting
It works on the front end and scripts are run on the browser.	It works on the back end and scripts are run on the server.
It does not require server interaction.	It requires server interaction for the processing of user requests.
It focuses on making the interface of web applications or websites.	It focuses on data accessing methods and error handling.
It is considered insecure or less secure.	It is relatively secure.
Scripting languages such as HTML, CSS, javascript, etc are used for client-side scripting.	Programming languages such as PHP, ASP.net, R, Python, etc are used for server-side scripting.

Data Types in JS: *[Important for MCQs]*
The different data types used in JavaScript are:

- String
- Number
- Boolean
- Undefined
- Null
- Object
- Array

JavaScript Variables:
The variables in JavaScript can be declared using following keywords:

- let
- var
- const

Note: The variables declared using ***const*** cannot be redeclared within the same program.
<u>Examples:</u>
let a = 10;
var b = “Nepal”;

JavaScript Function:

A JavaScript Function is a block of statements that can perform a particular task. We can define JS function by using ***function*** keyword as:

Syntax:

```
function function_name(parameters){
      //statements
}
```

Example: ***JS Program to add any two numbers using function***

```
let a = parseInt(window.prompt("Enter first number"));
let b = parseInt(window.prompt("Enter second number"));
let c = addition(a, b);
function addition(n1,n2){
      sum = n1+n2;
      return sum;
}
document.write("The sum is", c);
```

Note: A JavaScript function needs to be called to execute the statements written within it.

Some Important JavaScript Methods for Programming:

1. document.write( ) - Used to display content on a browser window.
2. console.log( ) - Used to display output on debug terminal.
3. window.prompt( ) - Used to display a dialog box that prompts the user to input data.

Note: window.prompt( ) method takes the value in the form of string, so we need to convert them into numeric using the following functions.

- *parseInt( ) - converts into integer*
- *parseFloat( ) - converts into float*

JavaScript Programs

1. Javascript program to add any two numbers.

```
let a = parseInt(window.prompt("Enter first number"));
let b = parseInt(window.prompt("Enter second number"));
let c = a + b;
document.write("The sum is", c);
```

2. Javascript program to calculate simple interest.

```
let p = parseInt(window.prompt("Enter Principle"));
let t = parseFloat(window.prompt("Enter Time"));
let r = parseFloat(window.prompt("Enter Rate"));
let  i = (p*t*r)/100;
document.write("The interest is", i);
```

3. Javascript program to convert temperature into Celsius.

```
let f = parseFloat(window.prompt("Enter temperature in
Fahrenheit"));
let  c = (5/9)*(f-32)
document.write("The temperature in celsius is", c);
```

4. Javascript program to find the largest or greatest number between two numbers.

```
let a = parseInt(window.prompt("Enter first number"));
let b = parseInt(window.prompt("Enter second number"));
if(a>b)
   document.write("The largest number is", a);
else
   document.write("The largest number is", b);
```

5. Javascript program to find the smallest number among any 3 numbers.

```
let a = parseInt(window.prompt("Enter first number"));
let b = parseInt(window.prompt("Enter second number"));
let c = parseInt(window.prompt("Enter third number"));
if((a<b) && (a<c))
   document.write("Smallest number is", a);
else if((b<a) && (b<c))
   document.write("Smallest number is", b);
else
   document.write("Smallest number is", c);
```

6. Javascript program to check whether a number is odd or even.

```
let a = parseInt(window.prompt("Enter a number"));
if(a%2!=0)
   document.write("Number is odd");
else
   document.write("Number is even");
```

7. Javascript program to reverse a given number.

```
let n = parseInt(window.prompt("Enter a number"));
Let rev = 0;
while(n!=0)
{
   r = n % 10;
   rev = rev * 10 + r;
   n = parseInt(n/10);
}
document.write("Reversed number is", rev);
```

8. Javascript program to check whether a given number is palindrome or not.

```
let n = parseInt(window.prompt("Enter a number"));
let rev = 0;
a = n
while(n!=0)
{
   r = n % 10;
   rev = rev * 10 + r;
   n = parseInt(n/10);
}
if(a==rev)
   document.write("Number is palindrome");
else
   document.write("Number is not palindrome");
```

9. Javascript program to calculate the factorial of a number.

```
let n = parseInt(window.prompt("Enter a number"));
let fact = 1;
for(i = n; i>=1; i--)
{
   fact = fact * i;
}
document.write("The factorial is", fact);
```

OR,

```
let n = parseInt(window.prompt("Enter a number"));
let fact = 1;
for(i = 1; i<=n; i++)
{
   fact = fact * i;
}
document.write("The factorial is", fact);
```

10. Javascript program to swap(interchange) the values of two variables.

```
let a = parseInt(window.prompt("Enter first number"));
let b = parseInt(window.prompt("Enter second number"));
let c;
//Swapping values
let c = a;
a = b;
b = c;
document.write("Value of a after swapping", a);
document.write("Value of b after swapping", b);
```

11. Javascript program to display the sum of the first ten terms of natural numbers.

```
let sum = 0;
for(i = 1; i<=10; i++)
{
   sum = sum + i;
}
document.write("The sum is", sum);
```

12. Javascript program to display the sum of n terms of natural numbers.

```
let n = parseInt(window.prompt("Enter number of terms"));
let sum = 0;
for(i = 1; i<=n; i++)
{
   sum = sum + i;
}
document.write("The sum is", sum);
```

13. Javascript program to display the first 20 terms of a fibonacci series: 0 1 1 2 3…… up to 20th terms.

```
let t1 = 0;
let t2 = 1;
for(i = 1; i<=20; i++)
{
   document.write(t1);
   t = t1 + t2;
   t1 = t2;
   t2 = t;
}
```

Form Validation:

Form validation is the process of checking if the form data is valid or not before it is processed.

Example: Basic Form Validation

```
<html>
<head>
    <title>Javascript Form</title>
</head>
<body>
 <form name="myform" onsubmit="return validateform()">
      Name: <input type="text" name="name">
      Password: <input type="password" name="password">
      <input type="submit" value="register">
   </form>
   <script>
      function validateform() {
            var name = document.myform.name.value;
            var password = document.myform.password.value;
            if (name == null || name == "") {
                alert("Name can't be blank");
                return false;
            }
            else if (password.length < 6) {
               alert("Password must be at least 6 characters
long.");
                return false;
            }
        }
    </script>
</body>
</html>
```

Javascript Email Validation:

```
<html>
      <head>
          <title>Javascript Form</title>
      </head>
<body>
      <form name="myform" onsubmit="return validate();">
            Email: <input type="text" name="email">
            <input type="submit" value="register">
       </form>

      <script>
         function validate() {
            var emailid = document.myform.email.value;
            var atposition = emailid.indexOf("@");
            var dotposition = emailid.lastIndexOf(".");
            if (atposition<1 || dotposition<atposition + 2)
            {
               alert("Please enter a valid email address");
               return false;
             }
         }
      </script>
    </body>
</html>
```

PHP(Hypertext Preprocessor):

PHP is a server-side scripting language that can be used to make dynamic and interactive websites. The PHP codes or scripts run on the server. It was known as a *"Personal Home Page"* on early days.

The default file extension of PHP file is .php.

Basic PHP Syntax: ***[Important for 1 Marks ~ MCQs]***

A PHP script or code can be inserted as follows:

```
<? php
   //PHP script or code
?>
```

It starts with <?php and ends with ?>

PHP Comments:

In PHP, comments can be single line or multiline and can be written as:

```
// Single Line Comment
# Single Line Comment

/*Multiline comment
in PHP */
```

PHP variables: *[Important for 1 marks ~ MCQs]*
A variable in PHP always starts with $ sign, followed by a variable name.
Examples: $a, $b, etc.

PHP Output:
We can either use ***echo*** or ***print*** statements to get the output in php.
Example: PHP program to add any two numbers

```
<? php
   $a = 10;
   $b = 20;
   $c = $a + $b;
   echo "Sum is ", $c;
   print "The sum is ",$c
?>
```

PHP Data Types: *[Important for 1 marks ~ MCQs]*
The data types in PHP can be group into following 3 types as:

- ➔ Scalar Types
  - integer
  - float
  - boolean
  - string
- ➔ Compound Types
  - array
  - object
- ➔ Special Types
  - resource
  - NULL

Note:
The period (.) operator is used to concatenate or join string data.

Example:

```
<? php
   $a = "Computer"
   $b = "Science";
   $c = $a.$b;
   echo $c;
?>
```

Here the output will be ComputerScience

⊙⊙⊙