

Chapters

3

WEB - TECHNOLOGY II

1. Write a function to add any two numbers in Javascript. (Model question)

→

<html>

 <head>

 <title> sum </title>

 </head>

 <body>

 <script>

 var x = 5;

 var y = 10;

 function sum()

 {

 var sum = x + y;

we can write also alert instead of document.write("The sum of "+x+" and "+y+" is "+sum);

 }

 sum(); // calling Javascript function

 </script>

 </body>

</html>

Output:

The sum of 5 and 10 is 15

2. Demonstrate the external CSS implemented in web page
→ CSS stands for cascading style sheets. CSS can control the layout of the multiple webpage once. There are three types of CSS, inline CSS, internal CSS and external CSS.

External CSS contains separate CSS file which contains only style property with the help of attributes. The separate CSS file should be linked with the HTML document using `<link>` tag.

HTML file (index.html)

```
<html>
  <head>
    <title>external CSS</title>
    <link rel = "stylesheet" href = "external.css">
  </head>
  <body>
    <p>surkuna</p>
  </body>
</html>
```

CSS file (external.css)

P.S

```
color: blue;
font-size: 30px;
```

3. write down the data types in java script with examples.

→ java script provides different data types to hold different types of values. There are two types of data types in java script.

- a) primitive data type
- b) Non-primitive data type

a) primitive data type

primitive data types can hold one value at a time. There are five types of primitive in javascript. They are as follows:

Data type	Description
String	Represents a sequence of characters e.g. "Hello"
Number	Represents numeric values e.g. 100
Boolean	Represents Boolean value either false or true
undefined	Represents an undefined value
Null	Represents null i.e., no value at all

b) Non-primitive Data Type

Non-primitive data types can hold collections of values and more complex entities. The non-primitive data types are as follows:

Data type	Description
Object	Represents an instance through which we can call member
Array	Represents a group of similar values
Regexp	Represents a regular expression

4. Explain the major feature and importance of JavaScript.



The major features of javascript are as follows:-

- a. Javascript is an object-based scripting language.
- b. It gives the users more control over the browsers.
- c. It is light weighted.
- d. Javascript is case sensitive.
- e. It offers validation of user's input.
- f. It offers event handling.
- g. It is supportable in several operating systems including, windows, MACOS, etc.

The Instances of Javascript are as follows:-

- a. Javascript are less complex and can be targeted for specific applications.
- b. Search engine, ecommerce, social media and phone apps would not be possible without Javascript.
- c. Javascript is only scripting language which is universally supported.
- d. It has opened many choices to the developers for Server-side.
- e. An important part of Javascript is the ability to create new function within scripts using function keyword.

5. Explain different Javascript operation with example.

→ Operator is a symbol that signifies the operations.

Operator is used in between two ~~operator~~ operands.

For e.g.: - $5+2$

where, 5 and 2 are operands and '+' is the operator.

The different types of javascript operations are as follows:

- a) **Arithmetic operators:** Arithmetic operators take numerical values as their operands and return a single numerical value. Arithmetic operators can perform mathematical operations. Some Arithmetic operators are +, -, *, /, %, ++, --, etc.
- b) **Comparison (Relational) operators:** Comparison operators are used to compare two operands. Its result is Boolean, either true or false. Some comparison operators are equal ($= =$), not equal ($!=$), greater than ($>$), less than ($<$), greater than equal (\geq), less than equal (\leq), etc.
- c) **Logical operators:** Logical operators are used for logical operators. It takes Boolean operands and gives Boolean result. The $\&\&$ and $||$ operators takes two operands and $!$ operator takes a single operand.

`&&` → logical AND

`||` → logical OR

`!` → logical NOT

d. **conditional Operators (Ternary operators)** : It is alternative to if- else conditions in javascript. Unlike other operators, it usually takes three expressions,

`Condition ? first expression : second expression;`

It has two operators, the "`?`" (question mark) and "`:`" (colon), which separates the condition from the result. If the condition is true, it will run and gives the first expression as result and if the condition is false, it runs and gives the second expression as result.

e. **Assignment Operators**: Assignment operator is a symbol used to assign a value or a ~~the~~ result of an expression to an identifier. Some assignment operators are (`=`, `+=`, `-=`, `*=`, `/=`, `&=`).

6. Describe different types of loop used in Javascript with examples.

→ The Javascript loops are used to iterate or repeat the execution of code using for, while, do-while loop.

There are three types of loops in Javascript.

a) for loop: for loop is used when the number of repetition are known. It consists of three expressions initialization, condition, and increment/decrement are the three expressions of for loop.

For eg. Javascript program to display the first 5 natural numbers using for loop.

<Script>

```
var i;
for(i=1;i<=5;i++)
```

{

```
document.write(i);
```

}

</Script>

Output:

12345

b) while loop: while loop is used when the number of repetition is unknown. It consists only one expression. i.e. condition. One condition becomes false, the loop will be terminated.

for eg. Javascript program to display the first 5 natural numbers using while loop.

```
<script>
```

```
var i=1;
```

```
while (i<=5)
```

```
{
```

```
document.write(i);
```

```
i++;
```

```
}
```

```
</script>
```

Output:

1 2 3 4 5

(1) do-while loop : do-while loop is similar to the while loop. The only difference is, it checks the condition at the end of the loop. This means that the loop will always be executed at least once.

for eg. Javascript program to display the first 5 natural numbers using do-while loop.

```
<script>
```

```
var i=1;
```

```
do
```

```
{
```

```
document.write(i);
```

```
i++;
```

```

    } while(i<=5);
</script>

```

Output:

1 2 3 4 5

7. What is variable? Explain different variables used in Javascript.

→ A variable is an identifier whose value may change during the execution of the program. It is simply a name of storage location. There are two types of variables in Javascript:

8. Local variable: A Javascript local variable is defined inside the block of function. It is accessible within the function or block only. For example:

~~function~~

<script>

function sum()

{

var x=5; //local variable

var y=10; //local variable

var sum=x+y;

document.write("The sum of "+x+" and
" +y+" is "+sum);

}

Sum(); //calling Javascript function

</script>

b. Global variable: A javascript variable is accessible from any function. A variable i.e. declared outside function or declared with a window object is known as a global variable. For example:

<script>

var x=5; //global variable

var y=10; //global variable

function sum()

{

var sum=x+y;

document.write("The sum of " + x + " and
" + y + " is " + sum);

}

sum(); //calling javascript function

</script>

8. How are events handled in javascript?

→ Javascript's interaction with HTML is handled through events that occur when the user or the browser manipulates a page. When the page loads, it is called an event. When the user clicks a button, that click is an event. Other examples include events like pressing any key, closing window, resizing a window.

Developers can use these events to execute javascript coded responses, which cause buttons to close windows, messages to be displayed to users, data to be validated, and virtually any other types of response imaginable. Some events are onclick, onsubmit, onmouseover and onmouseout, onfocus and onblur, etc.

For example:

```
<html>
```

```
<head>
```

```
<script>
```

```
function sayHello()
```

```
{
```

```
    alert("Hello world");
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
<p> click the button below to see the result </p>
<button onclick = "sayHello()"> Click </button>
</body>
</html>
```

9. Write a function to print first five natural numbers in Javascript.

→

```
<html>
<head>
<title> Natural numbers </title>
</head>
<body>
<script>
function print()
{
    vari ;
    for(i=1 ; i<=5 ; i++)
    {
        document.write(i);
    }
    print(); // calling Javascript function
</script>
</body>
```

<html>

Output:

1 2 3 4 5

10. write a function to find area of a circle in javascript.

→

<html>

<head>

<title> Area of circle </title>

</head>

<body>

<script>

function AOC()

{

var pi = 3.14;

var r = parseInt(prompt("enter a radius"));

var area = pi * r * r;

alert("Area of circle is " + area);

}

AOC(); // calling Javascript function

</script>

</body>

</html>

11. Write a function to check whether the given number is odd or even.



```
<html>
<head>
    <title> check odd or even </title>
</head>
<body>
    <script>
        function check()
        {
            var n = parseInt(prompt("enter a number"));
            if (n % 2 == 0)
                alert ("even number");
            else
                alert ("odd number");
        }
        alert (check()); // calling Javascript function
    </script>
</body>
</html>
```

12. write a Javascript program to calculate the simple interest using principle, time & rate.



```
<html>
  <head>
    <title> simple interest </title>
  </head>
  <body>
    <script>
      function simple-interest()
      {
        var p = parseInt(prompt("Enter principle"));
        var t = parseInt(prompt("Enter time"));
        var r = parseInt(prompt("Enter rate"));
        var SI = (p * t * r) / 100;
        alert("The simple interest is " + SI);
      }
      Simple-interest(); // calling javascript function
    </script>
  </body>
</html>
```

13. Comparison between client-side scripting and server-side scripting.

→ The comparisons between client-side Scripting and servers-side scripting are given below:

→ Client-side scripting

- It is a scripting that run on a client web browser.
- It deals with the user interface and lighter functionality.
- Response from a client-side script is faster.
- It cannot connect to the databases on the web server.
- Examples of client-side scripting languages: Java script, VB script, etc.

→ Server side Scripting

- It is normally used to handle browser request.
- The server executes the script and returns to the browser.
- Response from a server-side script is slower because the scripts are processed remotely.
- Server-side scripting is used to connect to the database that is on web server.

- example of server-side scripting languages:
PHP, JSP, ASP.NET, ASP, etc.

- ii. write a Javascript function to find the surface area of football. [area = $4\pi r^2$]

→

```
<html>
  <head>
    <title>area of football</title>
  </head>
  <body>
    <script>
      function AOF()
      {
        var pi= 3.14;
        var r= parseInt(prompt("enter
          a radius"));
        var area= 4* pi * r* r;
        document.write ("The surface area
          of football is "+area);
      }
      AOF(); // calling javascript function
    </script>
  </body>
</html>
```

* Write a javascript function to check whether the given number is positive, negative or zero.

```
<html>
  <head>
    <title>Check positive, negative or zero </title>
  </head>
  <body>
    function check ()
    {
      var n = parseInt(prompt("Enter a number"));
      if (n > 0)
        {
          alert ("positive number");
        }
      else if (n < 0)
        {
          alert ("negative number");
        }
      else
        {
          alert ("zero");
        }
    }
    check();
  </script>
```

<body>
</body>

*Last year
re-exam
QSN*
write a javascript function to find the greatest numbers among three numbers.

<html>
<head>
 <title> greatest number </title>
</head>
<body>
 <script>

 Function greatest ()

 {

 var a = parseInt(prompt ("enter first number"));

 var b = parseInt(prompt ("enter second number"));

 var c = parseInt(prompt ("enter third number"));

 if (a > b && a > c)

 {

 alert ("The greatest number is "+a);

 }

 else if (b > a && b > c)

 {

 alert ("The greatest number is "+b);

 }

 else

{

alert ("The greatest number is "+c);

}

}

greatest ();

<script>

</body>

</html>

inst year
asn.

write a javascript function to find factorial
of a given number.

<html>

<head>

<title> factorial </title>

</head>

<body>

<script>

function factorial ()

{

var n = parseInt(prompt ("enter a number"));

var mul = 1;

for (i = 1; i <= n; i++)

{

mul = mul * i;

}

alert ("The factorial is " + mul);

3
Factorial();
<script>
<body>
</html>

Server-side scripting using PHP

PHP is a server-side scripting language designed to be used for web purposes. PHP is so popular because it's very simple to learn, code and deploy on the server, hence it has been the first choice for beginners for decades.

Benefits of PHP

- a. open source
- b. inexpensive hosting and setup
- c. cross-platform
- d. client applications don't need PHP installed
- e. lots of free libraries and packages available for use.

1. Write a PHP program to display "hello world".

→ <?php
echo "hello world";
?>

2. Write a PHP program to add two numbers.

→

```
<?php  
$x=5;  
$y=10;  
$sum=$x+$y;  
echo "The sum of $x and $y is $sum";  
?>
```

3. write a PHP program to check whether the mark is "pass" or "fail".

→

```
<?php  
$mark=20;  
if ($mark >= 40)  
{  
    echo "pass";  
}  
else  
{  
    echo "fail";  
?>
```

4. Write a PHP program to check whether the number is "odd" or "even".

→

```
< ?php
    $n = 25;
    if ($n % 2 == 0)
    {
        echo "even";
    }
    else
    {
        echo "odd";
    }
? >
```

5. Write a PHP program to find the factorial of a number.

→

```
< ?php
    $n = 25;
    if ($n % 2 == 0)
    {
        echo "even";
    }
    else
    {
```

```
< ?php
    $n = 4;
    $mul = 1;
    for ($i = 1; $i <= $n; $i++)
    {
        $mul = $mul * $i;
    }
    echo "The factorial of $n
is $mul";
```

echo "odd"; ?>

?>
?>

6. write a PHP program to find the sum of two numbers. (alternate method using HTML)

→

```
<html>
  <head>
    <title> sum </title>
  </head>
  <body>
    <form method="post">
      first number : <br>
      <input type="number" name="a"> <br>
      second number : <br>
      <input type="number" name="b"> <br>
      <input type="submit" name="submit">
    </form>
    <?php
      if (isset($_POST['submit']))
      {
        $a = $_POST['a'];
        $b = $_POST['b'];
        $sum = $a + $b;
        echo "the sum of $a and $b is $sum";
      }
    </?php>
  </body>
</html>
```

?>
 </body>
 </html>

- 7 write a PHP program to check whether the given number is "even" or "odd". (alternate method using html)

→

```

<html>
  <head>
    <title>Checking odd or even </title>
  </head>
  <body>
    <form method = "POST">
      Enter a number: <br>
      <input type = "number" name = "n" > <br>
      <input type = "submit" name = "submit" >
    </form>
    <?php
      if (isset($_POST['submit']))
        $n = $_POST['n'];
        if ($n % 2 == 0)
          {
            echo "even";
          }
    
```

Date :

Page:

```
    }  
else  
{  
    echo "odd";  
}  
?  
</body>  
</html>
```

8. write a PHP program to find the factorial of a given number. (alternate method using html)

→

```
<html>  
<head>  
    <title> sum </title>  
</head>  
<body>  
    <form method = "post">  
        Enter a number: <br>  
        <input type = "numbers" name = "n" > <br>  
        <input type = "submit" name = "submit" >  
    </form>  
    <?php  
        if(isset($_POST['submit']))  
    {
```

Date :

Page:

```
$n = $_POST['n'];
$mul = 1;
for ($i=1; $i <= $n; $i++)
{
    $mul = $mul * $i;
}
echo "The factorial of $n is $mul";

```

</body>

</html>

9. How do you connect MySQL database with PHP?
Demonstrate with example.

→

```
<?php
//connecting to the database
$servername = "localhost";
$username = "root";
$password = "";

//creating a connection
$conn = mysqli_connect($servername, $username,
                      $password);

//die if the connection was not successfull
```

```
if($conn)
{
    echo "The connection was successful";
}
else
{
    die("Sorry, the connection was not successful".
        mysqli_connect_error());
}
mysqli_close($conn);
```

10. Write a server-side scripting code to create a database "studentdb". Assume that `servername = "localhost"`, `username = "root"`, `password = ""`.

→

```
<?php
// Connecting to the database
$servername = "localhost";
$username = "root";
$password = "";

// Creating a connection
$conn = mysqli_connect($servername, $username,
    $password);

// Die if the connection was not successful
```

```
if ($conn)
{
    echo "The connection was successful<br>";
}
else
{
    die ("sorry, the connection was not successful:" .
        mysqli_connect_error ());
}

// creating a database
$sql = "CREATE DATABASE student DB";
$result = mysqli_query ($conn, $sql);

// checking database creation
if ($result)
{
    echo "Database created successfully";
}
else
{
    echo "Database is not created successfully".
        mysqli_error ($conn);
}

mysqli_close ($conn);
?>
```

11. write a server - side scripting code to create table "student" with fields (firstname, lastname, mark and email). Assume thatservername = "localhost", username = "root", password = "", database = "studentdb".

→

```
<?php  
    //connecting to the database.  
    $servername = "localhost";  
    $username = "root";  
    $password = "";  
    $database = "studentdb";  
  
    // creating a connection  
    $conn = mysqli_connect($servername, $username,  
                           $password, $database);  
  
    //die if the connection was not successful  
    if (!$conn)  
    {  
        echo "The connection was successful<br>";  
    }  
    else  
    {  
        die("Sorry, the connection was not success-  
            full: ". mysqli_connect_error());  
    }
```

//creating a table in the database
\$sql = "CREATE TABLE student (firstname varchar(55),
lastname varchar(55), mark int, email varchar(55));
\$result = mysqli_query(\$conn, \$sql);

//checking table creation
if(\$result)
{
 echo "table created successfull";
}
else
{
 echo "table is not created successfull". mysqli_error(\$conn);
}
mysqli_close(\$conn);
?>

12. write a server-side scripting code to insert data into the table "student" having fields (firstname, lastname, mark and email). Assume that servername = "localhost", username = "root", password = "", database name = "studentdb". [2080]

→

```
<?php
    //connecting to the database
    $servername = "localhost";
    $username = "root";
    $password = "";
    $database = "studentDB";

    //creating a connection
    $conn = mysqli_connect($servername, $username,
        $password, $database);

    //die if the connection was not successful
    if ($conn)
    {
        echo "The connection was successful<br>";
    }
    else
    {
        die("sorry, the connection was not success-
            ful:".mysqli_connect_error());
    }
}
```

Date:

Page:

}

// inserting data in the table

```
$sql = "INSERT INTO student (firstname, lastname,  
mark, email) VALUES ('anil', 'thapa', '99', 'anil@-  
gmail.com')";
```

```
$result = mysqli_query($conn, $sql);
```

// checking whether the data is inserted or not.

```
if ($result)
```

{

```
    echo "Data inserted successfully";
```

}

```
else
```

{

```
    echo "Data is not inserted successfully". mysqli_error  
        ($conn);
```

}

```
    mysqli_close($conn);
```

```
?>
```

13. write a server-side scripting code to select data from table "student" having fields (firstname, lastname, mark and email). Assume thatservername = "localhost", username = "root", password = "", database name = "studentdb".

→

< ?php

// connecting to the database

\$servername = "localhost";

\$username = "root";

\$password = " ";

\$database = "student DB";

// creating a connection

\$conn = mysqli_connect(\$servername, \$username, \$password, \$database);

// die if the connection was not successful

if (\$conn)

{

echo "The connection was successful
";

}

else

{

die("Sorry, the connection was not successful: ". mysqli_connect_error());

{

//selecting all data from the table

\$sql = "SELECT * FROM Student";

\$result = mysqli_query(\$conn, \$sql);

//total number of retrieved records

\$num = mysqli_num_rows(\$result);

//Displaying all the retrieved records

echo "The records found in the database are:

";

if (\$num > 0)

{

while (\$row = mysqli_fetch_assoc(\$result))

{

echo "firstname = ". \$row['firstname']. "
";

echo "lastname = ". \$row['lastname']. "
";

echo "mark = ". \$row['mark']. "
";

echo "email = ". \$row['email']. "
";

echo "

";

}

{

mysqli_close(\$conn);

?>