

***BSc.(Information Technology)***  
***(Semester II)***  
***2018-19***

***Web Programming***  
***(USIT203 Core)***  
***University Paper Solution***

***By***  
***Mrs. Pallavi Tawde***

### **Question 1**

**Q1a. Explain different approaches of style sheets.**

**Ans**

#### **Inline Styles**

An inline style loses many of the advantages of style sheets by mixing content with presentation. Use this method sparingly, such as when a style is to be applied to a single occurrence of an element. To use inline styles you use the style attribute in the relevant tag. The style attribute can contain any CSS property. The example shows how to change the color and the left margin of a paragraph:

```
<p style= "color: sienna; margin-left: 20px">
```

This is a paragraph</p>

#### **Internal Style Sheet**

An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section by using the <style>tag, like this:

```
<head>
```

```
<style type= "text / css">
```

```
hr {color: sienna}
```

```
p {margin-left: 20 px}
```

```
body {background-image: url ("images/back40.gif")}
```

```
</style>
```

```
</head>
```

#### **External Style Sheet**

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing one file. Each page must link to the style sheet using the <link>tag. The <link> tag goes inside the head section:

```
<head>
```

```
<link rel= "stylesheet" type="text/css"
```

```
href= "mystyle.css"/>
```

```
</head>
```

An example of a style sheet file is shown below: hr {color: sienna}

```
p {margin-left: 20px}
```

```
body {background-image: ur1("images/back40.gif")}
```

**Q1b. Explain the structure of HTML5 file.**

**Ans:**

### **The Structure of HTML Documents**

The following is the general structure of HTML document

```
<!DOCTYPE HTML PUBLIC "html version">
<HTML>
<HEAD>
<TITLE>Document Title</TITLE>
...Other supplementary information goes here....
</HEAD>
<BODY>
...Marked-up text goes here....
</BODY>
</HTML>
```

- ☐ An HTML document begins with a **<!DOCTYPE>** declaration, indicating the version of HTML used by the document.
- ☐ Following this, the **<HTML>** element encloses the actual document. It contains two primary sections, the head and the body, enclosed by the **<HEAD>** and **<BODY>** elements, respectively.
- ☐ The head can contain identifying and other supplementary information about the document. The head always contains the document's title, enclosed by the **<TITLE>** element.
- ☐ The body contains the actual document content and the HTML markup used to structure the document.
- ☐ The **<HTML>** element delimits the beginning and the end of an HTML document. It contains only the **<HEAD>** element and the **<BODY>** element.

**Q1c. What is Ecommerce? Explain the advantages, disadvantages and types of ecommerce.**

**Ans:** E-Commerce or Electronic Commerce means buying and selling of goods, products, or services over the internet. E-commerce is also known as electronic commerce or internet commerce. These services provided online over the internet network. Transaction of money, funds, and data are also considered as E-commerce. The standard definition of E-commerce is a commercial transaction which is happened over the internet. Online stores like Amazon, Flipkart, Shopify, Myntra, Ebay, Quikr, Olx are examples of E-commerce websites.

**Advantages of E-Commerce**

- E-commerce provides the sellers with a global reach. They remove the barrier of place (geography). Now sellers and buyers can meet in the virtual world, without the hindrance of location.
- Electronic commerce will substantially lower the transaction cost.
- It provides quick delivery of goods with very little effort on part of the customer.
- One other great advantage is the convenience it offers. A customer can shop 24×7.
- Electronic commerce also allows the customer and the business to be in touch directly, without any intermediaries.

**Disadvantages of E-Commerce**

- The start-up costs of the e-commerce portal are very high.
- Although it may seem like a sure thing, the e-commerce industry has a high risk of failure.
- At times, e-commerce can feel impersonal. So it lacks the warmth of an interpersonal relationship which is important for many brands and products.
- Security is another area of concern. Only recently, we have witnessed many security breaches where the information of the customers was stolen. Credit card theft, identity theft etc. remain big concerns with the customers.
- Then there are also fulfillment problems. Even after the order is placed there can be problems with shipping, delivery, mix-ups etc.

**Types of E-Commerce Models**

**1. Business to Business:** This is Business to Business transactions. Here the companies are doing business with each other. The final consumer is not involved. So the online transactions only involve the manufacturers, wholesalers, retailers etc.

**2. Business to Consumer:** Business to Consumer. Here the company will sell their goods and/or services directly to the consumer. The consumer can browse their websites and look at products, pictures, read reviews. Then they place their order and the company ships the goods directly to them. Examples: Amazon, Flipkart, Jabong

**3. Consumer to Consumer:** Consumer to consumer, where the consumers are in direct contact with each other. No company is involved. It helps people sell their personal goods and assets directly to an interested party. Usually, goods traded are cars, bikes, electronics etc. OLX, Quikr etc follow this model.

**4. Consumer to Business:** This is the reverse of B2C, it is a consumer to business. So the consumer provides a good or some service to the company. Say for example an IT freelancer who demos and sells his software to a company. This would be a C2B transaction.

**Q1d. Write a short note on File Transfer Protocol.**

**Ans:**

File Transfer Protocol service (known as FTP in short) enables an Internet user to move a file from one computer to another on the Internet. A file may contain any type of digital information—text document, image, artwork, movie, sound, software, etc. Moving a file from a remote computer to one's own computer is known as *downloading* the file, and moving a file from one's own computer to a remote computer is known as *uploading* the file.

By using FTP service, a file transfer takes place in the following manner:

- i. A user executes the *ftp* command on his/her local computer, specifying address of the remote computer as a parameter.
- ii. An FTP process running on user's computer (called FTP client process) establishes a connection with an FTP process running on remote computer (called FTP server process).
- iii. The system then asks the user to enter his/her login name and password on the remote computer to ensure that the user is authorized to access the remote computer.
- iv. After successful login, desired file(s) are downloaded or uploaded by using *get* (for downloading) and *put* (for uploading) commands. User can also list directories, or move between directories of the remote computer, before deciding which file(s) to transfer.

The user needs access rights for a remote computer to transfer files to/ from it. With this restriction, it is almost impossible to provide access rights to the vast number of users on the Internet to a computer that contains sharable information. The concept of an anonymous FTP site is used to overcome this problem. An anonymous FTP site is a computer allowing a user to log in with a username of *anonymous* and a password that is user's e-mail address. The user can then download files that have been stored on it for sharing. Such sites are called publicly accessible sites because any user on the Internet can access them.

Of course, FTP service is also used for more secure file transfer operations. In such cases, a user needs a valid username and password to access a particular computer. This is common; for example, for organizations that wish to let only certain people access their computer.

**Q1e. How are ordered lists created in HTML? Explain with the help of an example.**

**Ans:**

### **Ordered Lists**

An ordered list is also a list of items. The list items are marked with numbers.

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

```
<ol>
<li>Coffee</li>
<li>Milk</li>
</ol>
```

Here is how it looks in a browser:

1. Coffee
2. Milk

Inside a list item you can put paragraphs, line breaks, images, links, other lists, etc.

### **Optional Attributes**

Attribute	Value	Description
Start	<i>Number</i>	Specifies the start point in a list
Type	decimal (1, 2, 3, 4 (the default)) decimal-leading-zero (01, 02, 03, 04) lower-roman (i, ii, iii, iv) upper-roman (I, II, III, IV) lower-alpha (a, b, c, d) upper-alpha (A, B, C, D) none(nothing)	Specifies which kind of bullet points will be used.

**Q1f. Write HTML code to design a web page using any five formatting tags.**

**Ans:**

- I - Italics
- B - Bold
- U - Underline
- STRIKE - Strikeout
- BIG - Larger text
- SMALL - Smaller text
- SUB - Subscript
- SUP - Superscript

```
<html>
<body>
<b>This text is bold</b> <br>
<strong>This text is strong</strong> <br>
<big>This text is big</big> <br>
<em>This text is emphasized</em> <br>
<i>This text is italic</i> <br>
<small>This text is small</small> <br>
This text contains<sub>subscript</sub> <br>
This text contains<sup>superscript</sup>
</body>
</html>
```

## Question 2

**Q2a. What is Image map? Write the difference between client side and server side image mapping.**

**Ans:** An image map on the WWW permits the web document writer to include in its document the ability for a viewer to "click" on a portion of a GIF or JPEG (or any other image) file. There are three components in a Web page image map:

1. **An image:** a normal web image, usually stored in the GIF or JPEG format.
2. **A set of map data:** a description of the mapped regions within the image.
3. **An HTML host entry:** the HTML code that positions the image within the web page and designates the image as having map functionality.

The regions in a map include shape descriptions (rectangle, circle, polygon, or background), numeric data defining the coverage of the regions within the image, and link URLs. It is possible to define regions without links, using the **NOHREF tag**.

### HTML <area> tag

The <area> tag defines an area inside an image-map (an image-map is an image with clickable areas).

The area element is always nested inside a <map> tag.

Attribute	Value	Description
Shape	rect, circle, poly	Specifies the shape of area
Cords	number	Specifies the co-ordinates of particular shape
Href	url	Specifies the page that will be displayed on the click of that area

```

  <map name="planetmap">
    <area shape="rect" coords="0,0,82,126" href="sun.htm" alt="Sun" />
    <area shape="circle" coords="90,58,3" href="mercur.htm" alt="Mercury" />
    <area shape="circle" coords="124,58,8" href="venus.htm" alt="Venus" />
  </map>
```

- **Server-side.** These maps are interpreted by the Web server. How server-side maps are stored depends on software installed on the Web server. When a user clicks within a server-side image map the browser program transfers the coordinates of the click to a program running on the Web server. This program then examines the map data and determines the link.

- **Client-side maps.** These maps are interpreted by the Web browser program - the map data is embedded in the host page. The Web browser processes a users' click without interacting with the server. This reduces the server load and provides better feedback to the page visitor. The crucial difference here from Server-side maps is that all the link and region instructions - the map files - are now stored directly in the HTML document. Hence, the browser handles all the work locally. The problem with this is that only a few browsers currently support this option.



**Q2b. Write HTML code for the following:**

Basic Fruit  
Comparison Chart

Fruit	Color
Apple	Red
Kiwi	Green
Watermelon	Pink

**Ans:**

```
<html> <head> <title>Table tag example</title> </head>
<body>
<table border="3" width="25%">
<caption> Basic Fruit<br>Comparison Fruit </caption>
<tr> <th>Fruit</th> <th>Color</th> </tr>
<tr> <td>Apple</td> <td>Red</td> </tr>
<tr> <td>Kiwi</td> <td>Green</td> </tr>
</table>
</body>
</html>
```

**Q2c. Write a HTML 5 program to display college admission form.**

**Ans:**

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="style2.css">
</head>
<body>
<div class="bodyright">
<p>Application Form</p>
<div class="registration">
<form id="login" action="" method="post">
<input type="text" placeholder="First Name" class="radius1" />
<input type="text" placeholder="Surname" class="radius1"/> <br/> <br/>
<input type="text" placeholder="Father's Name" value="" class="radius"
/> <br/> <br/>
<input type="text" placeholder="Mobile number or E-mail address" class="radius"
/> <br/> <br/>
<select name="year" class="radius"> <option>Select
Course</option> <option>B.Tech</option>
<option>B.Sc</option> <option>B.Arch</option> </select> <br/> <br/>
Select your skills<br/>
<input type="checkbox"> Programming<br/>
<input type="checkbox"> Creative Writing<br/>
<input type="checkbox"> Singing<br/>
<input type="checkbox"> Dancing<br/> <br/>
Birthday<br/>
<select name="day"> <option>Day</option> <option>1</option>
<option>2</option> <option>3</option> </select>
<select
name="month"> <option>January</option> <option>February</option> </select>
<select name="year"> <option>Year</option> <option>1997</option>
<option>1998</option> <option>1999</option> </select>
<a href="#" class="need"> Why do I need to provide my date of
birth?</a> <br/> <br/>
<input type="radio" name="gender"/> Female<input type="radio"
name="gender"/> Male<br/>
<p>By clicking Create an account, you agree to our
<a href="#" class="need"> Terms & Conditions</a> </p>
<button class="title" name="signup"> Submit</button>
</form>
</div> </div>
</body> </html>
```

**Q2d. Explain the following semantic tags with example.**

**i. <header> ii. <nav> iii. <figcaption>**

**Ans:**

A semantic element clearly describes its meaning to both the browser and the developer.

**i. <header> Element**

The <header> element specifies a header for a document or section.

The <header> element should be used as a container for introductory content.

You can have several <header> elements in one document.

The following example defines a header for an article:

Example

```
<article>
  <header>
    <h1>What Does WWF Do?</h1>
    <p>WWF's mission:</p>
  </header>
  <p>WWF's mission is to stop the degradation of our planet's natural environment,
  and build a future in which humans live in harmony with nature.</p>
</article>
```

**ii. <nav> Element**

The <nav> element defines a set of navigation links.

```
<nav>
  <a href="/html/">HTML</a> |
  <a href="/css/">CSS</a> |
  <a href="/js/">JavaScript</a> |
  <a href="/jquery/">jQuery</a>
</nav>
```

**iii. <figcaption> Elements**

The purpose of a figure caption is to add a visual explanation to an image.

In HTML5, an image and a caption can be grouped together in a <figure> element:

```
<figure>
  
  <figcaption>Fig1. - Trulli, Puglia, Italy.</figcaption>
</figure>
```

**Q2e. Explain the concepts of redirection to another URL? Design a HTML page that automatically redirects the page to index.html after 4 seconds.**

**Ans:**

Page redirection is a situation where you clicked a URL to reach a page X but internally you were directed to another page Y. It happens due to page redirection. To redirect from an HTML page, use the META Tag. With this, use the http-equiv attribute to provide an HTTP header for the value of the content attribute. The value of the content is the number of seconds; you want the page to redirect after. Set the content attribute to 0, if you want the page to load the new URL immediately. The following is an example of redirecting current page to another website after 3 seconds.

**Example**

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Meta Tag</title>
    <meta http-equiv = "refresh" content = "4; url = index.html" />
  </head>
  <body>
    <p>Redirecting to another URL</p>
  </body>
</html>
```

**Q2f. Explain the <audio> tag with all its attribute.**

**Ans:**

### **<audio> tag**

The <audio> tag defines sound, such as music or other audio streams.

Currently, there are 3 supported file formats for the <audio> element: MP3, Wav, and Ogg:

### **Attributes**

Attribute	Value	Description
<u>autoplay</u>	autoplay	Specifies that the audio will start playing as soon as it is ready
<u>controls</u>	controls	Specifies that audio controls should be displayed (such as a play/pause button etc)
<u>loop</u>	loop	Specifies that the audio will start over again, every time it is finished
<u>muted</u>	muted	Specifies that the audio output should be muted
<u>preload</u>	auto metadata none	Specifies if and how the author thinks the audio should be loaded when the page loads
<u>src</u>	URL	Specifies the URL of the audio file

### **Example**

```
<html>
```

```
<body>
```

```
<audio controls>
```

```
  <source src="horse.ogg" type="audio/ogg">
```

```
  <source src="horse.mp3" type="audio/mpeg">
```

```
  Your browser does not support the audio element.
```

```
</audio>
```

```
<p><strong>Note:</strong> The audio tag is not supported in Internet Explorer 8  
and earlier versions.</p>
```

```
</body>
```

```
</html>
```

### Question 3

#### **Q3a. Explain the different conditional statements in JavaScript.**

**Ans:** In JavaScript we have the following conditional statements:

- Use **if** to specify a block of code to be executed, if a specified condition is true
- Use **else** to specify a block of code to be executed, if the same condition is false
- Use **else if** to specify a new condition to test, if the first condition is false
- Use **switch** to select one of many blocks of code to be executed

**Example :** If time is less than 10:00, create a "Good morning" greeting, if not, but time is less than 20:00, create a "Good day" greeting, otherwise a "Good evening":

```
var time = new Date().getHours();
if (time < 10) {
  greeting = "Good morning";
} else if (time < 20) {
  greeting = "Good day";
} else {
  greeting = "Good evening";
}
```

**Example :** Execute a block of code based on user input:

```
var text;
var fruits = document.getElementById("myInput").value;
```

```
switch(fruits) {
  case "Banana":
    text = "Banana is good!";
    break;
  case "Orange":
    text = "I am not a fan of orange.";
    break;
  case "Apple":
    text = "How you like them apples?";
    break;
  default:
    text = "I have never heard of that fruit...";
}
```

**Q3b. List and explain the methods of date object.**

**Ans: Date Methods**

<b>Sr.No.</b>	<b>Method &amp; Description</b>
1	Date() Returns today's date and time
2	getDate() Returns the day of the month for the specified date according to local time.
3	getDay() Returns the day of the week for the specified date according to local time.
4	getFullYear() Returns the year of the specified date according to local time.
5	getHours() Returns the hour in the specified date according to local time.
6	getMonth() Returns the month in the specified date according to local time.
7	getSeconds() Returns the seconds in the specified date according to local time.
8	setDate() Sets the day of the month for a specified date according to local time.
9	setFullYear() Sets the full year for a specified date according to local time.
10	setHours() Sets the hours for a specified date according to local time.
11	setMilliseconds() Sets the milliseconds for a specified date according to local time.
12	setMinutes() Sets the minutes for a specified date according to local time.

```
<html>
<body>
<script language="javascript">
    var d = new Date();
    document.write("Todays date and time"+d+"<br>");
    document.write("Day:-"+d.getDate()+"<br>");
    document.write("Day of week:-"+d.getDay()+"<br>");
    document.write("Year:-"+d.getFullYear()+"<br>");
    document.write("Hours:-"+d.getHours()+"<br>");
    document.write("Minutes:-"+d.getMinutes()+"<br>");
    document.write("Month:-"+d.getMonth()+"<br>");
    document.write("Seconds:-"+d.getSeconds()+"<br>");
    document.write("Time:-"+d.getTime()+"<br>");
    d.setDate(20)
    document.write("Set Date:-"+d+"<br>");
    d.setFullYear(2020);
    document.write("Set FullYear:-"+d+"<br>");
    d.setHours(12);
    d.setMinutes(46);
    d.setMonth(4);
    d.setSeconds(12);
    document.write("Modified Date:-"+d);
</script>
</body>
</html>
```

**Q3c. Write a JavaScript program to calculate the factorial of a number.**

**Ans:**

```
<html>
<head>
<script>
function show(){

var i, no, fact;
fact=1;
no=Number(document.getElementById("num").value);
for(i=1; i<=no; i++)
{
fact= fact*i;
}
document.getElementById("answer").value= fact;
}
</script>
</head>
<body>
Enter Num: <input id="num">
<button onclick="show()">Factorial</button>
<input id="answer">
</body>
</html>
```

Enter Num:



**3.d Explain the following methods of Math object with example:**

**i. ceil() ii. max() iii. random() iv. round() v. sqrt()**

**Ans:**

**i. ceil():** Math.ceil(x) returns the value of x rounded **up** to its nearest integer:

Math.ceil(4.4); // returns 5

**ii. max():** Math.min() and Math.max() can be used to find the lowest or highest value in a list of arguments:

Math.max(0, 150, 30, 20, -8, -200); // returns 150

**iii. random() :** Math.random() returns a random number between 0 (inclusive), and 1 (exclusive):

Math.random(); // returns a random number

**iv. round():** Math.round(x) returns the value of x rounded to its nearest integer:

Math.round(4.7); // returns 5

Math.round(4.4); // returns 4

**iv. sqrt():** Math.sqrt(x) returns the square root of x:

Math.sqrt(64); // returns 8

### **Q3 e. Explain following events:**

**i. onblur () ii. onchange() iii. ondblclick() iv. onkeydown() v. onmousedown()**

**Ans:**

**i. onblur ()** : Triggers when the window loses focus

**ii. onchange()** : Triggers when an element changes

**iii. ondblclick()** : Triggers on a mouse double-click

**iv. onkeydown()** : Triggers when a key is pressed

**v. onmousedown()** : Triggers when a mouse button is pressed

```
<html>
<head>
<script>
function myblur() {
    var x = document.getElementById("fname");
    x.value = x.value.toUpperCase();
}
function mychange() {
    var x = document.getElementById("mySelect").value;
    document.getElementById("demo").innerHTML = "You selected: " + x;
}
function mydbl() {
    document.getElementById("demo1").innerHTML = "Hello World";
}
function mydown() {
    alert("You pressed a key inside the input field");
}

function mouseDown() {
    document.getElementById("myP").style.color = "red";
}
</script>
</head>
<body>
Enter your name: <input type="text" id="fname" onblur="myblur()">
<p>Select a new car from the list.</p>
<select id="mySelect" onchange="mychange()">
    <option value="Audi">Audi</option>
    <option value="BMW">BMW</option>
    <option value="Mercedes">Mercedes</option>
    <option value="Volvo">Volvo</option>
</select>
<p id="demo"></p>
<p ondblclick="mydbl()">Double-click this paragraph to trigger a function.</p> <p
id="demo1"></p>
<input type="text" onkeydown="mydown()">
<p id="myP" onmousedown="mouseDown()">
Click the text! The mouseDown() function is triggered when the mouse button is
pressed down over this paragraph, and sets the color of the text to red. </p>
</body> </html>
```

**Q3f. Write a JavaScript program to accept a number from the user and display sum of its digits.**

**Ans:**

```
<html>
<head>
<script>
function sumDigits() {
    var value = document.getElementById("thenumber").value,
        sum = 0;

    while (value) {
        sum += value % 10;
        value = Math.floor(value / 10);
    }

    var rezultat = document.getElementById("result");
    rezultat.value = sum;
}
</head>
<body>
<input type="text" placeholder="number" id="thenumber"/> <br/> <br/>
<button onclick="sumDigits()">Calculate</button> <br/> <br/>
<input type="text" readonly="true" placeholder="the result" id="result"/>
</body>
</html>
```

#### Question 4

**Q4a. . Write a short note on variables in php.**

**Ans:** A variable can have a short name (like x and y) or a more descriptive name (age, carname, total\_volume).

Rules for PHP variables:

- A variable starts with the \$ sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_ )
- Variable names are case-sensitive (\$age and \$AGE are two different variables)

**Example:**

```
<html>
<body>

<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;

echo $txt;
echo "<br>";
echo $x;
echo "<br>";
echo $y;
?>

</body>
</html>
```

**Output:**

```
Hello world!
5
10.5
```

**Q4b. Explain different types of arrays available in php**

**Ans:**

An array is a special variable, which can hold more than one value at a time.

In PHP, the array() function is used to create an array:

```
array();
```

In PHP, there are three types of arrays:

**Indexed arrays - Arrays with a numeric index**

```
<?php
```

```
$cars = array("Volvo", "BMW", "Toyota");
```

```
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
```

```
?>
```

**Associative arrays - Arrays with named keys**

```
<?php
```

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
```

```
echo "Peter is " . $age['Peter'] . " years old.";
```

```
?>
```

**Multidimensional arrays - Arrays containing one or more arrays**

```
<?php
```

```
echo $cars[0][0].": In stock: ".$cars[0][1].", sold: ".$cars[0][2]."<br>";
```

```
echo $cars[1][0].": In stock: ".$cars[1][1].", sold: ".$cars[1][2]."<br>";
```

```
echo $cars[2][0].": In stock: ".$cars[2][1].", sold: ".$cars[2][2]."<br>";
```

```
echo $cars[3][0].": In stock: ".$cars[3][1].", sold: ".$cars[3][2]."<br>";
```

```
?>
```

**Q4c. Explain difference between break statement and exit statement with the help of an example.**

**Ans:**

**Break:**

- A break statement can be used to terminate or to come out from the loop or conditional statement unconditionally.
- It can be used in switch statement to break and come out from the switch statement after each case expression.
- Whenever, break statement is encountered within the program then it will break the current loop or block.
- A break statement is normally used with if statement.
- When certain condition becomes true to terminate the loop then break statement can be used.

The following program demonstrates the use of break statement. Loop will be terminated as soon as the counter value becomes greater than 5.

<?php

```
for( $i = 1; $i <= 10 ; $i++ )
{
    if ($i > 5)
        break; // terminate loop
    echo "$i."</br>" ;
}
```

?>

**Exit:**

- An exit statement is used to terminate the current execution flow.
- As soon as exit statement is found, it will terminate the program.
- It can be used to output a message and terminate the current script: for example exit("Good Bye!");
- It can also be used with error code. For example: exit(1), exit(0376).
- the following program demonstrates the use of exit statements.

<?php

```
$filename = 'sample.txt' ;
$file = fopen($filename, 'r') // open file for reading
or exit("unable to open file ($filename)");
```

?>

**Q4d. Explain the following string functions:**

**i) strlen() ii) strpos() iii) strrev() iv) str\_replace() v) str\_word\_count().**

**Ans:**

**i. strlen():** The strlen() function returns the length of a string.

Syntax: strlen(string)

Example: <?php echo strlen("hi"); ?>

Output:2

**ii. strpos():** The strpos() function finds the position of the first occurrence of a string inside another string.

Syntax: strpos(string,find,start)

-string :Specifies the string to search.

-find : specifies the string to find.

-start : specifies where to begin the search(optional).

Example: <?php echo strpos("i love php, i love php too!","php"); ?>

Output : 7

**iii. strrev():** The strrev() function is used to reverse a string.

Syntax: strrev(string)

Example: <?php echo strrev("hi"); ?>

Output : ih

**iv. str\_replace( ):** The str\_replace() function replaces a part of a string with another string.

Syntax: str\_replace(string,string)

Example: <?php echo str\_replace("world","everyone","hello world!"); ?>

Output: hello everyone!

**v. str\_word\_count():** The str\_word\_count() function counts the number of words in a string.

Syntax: str\_word\_count(string)

Example: <?php echo str\_word\_count("Good Morning"); ?>

Output: 2

**Q4e. Write a program in PHP to create a function to calculate average of three numbers and display it.**

**Ans:**

**Index.html : HTML page to take 3 input numbers**

```
<html>
  <head>
    <title>   Average of three numbers      </title>
  </head>
  <body>
    <form action="avg.php" method="post">
      Enter your numbers:<br>
      <input type="text" name="txtno1"> <br>
      <input type="text" name=" txtno2"> <br>
      <input type="text" name=" txtno3">
      <input type="submit" value="Calculate average">
    </form>
  </body>
</html>
```

**avg.php : php page to calculate average**

```
<?php
$a=$_POST["txtno1"];
$b=$_POST["txtno1"];
$c=$_POST["txtno1"];
$avgno=($a+$b+$c)/3;
echo "Average of numbers is: ".$avgno;
?>
```



**Q4f. Write the difference between GET and POST methods in PHP.**

**Ans:**

**The GET method**

It sends encoded user information append to the page request.

- The GET method produces a long string that appears in your server logs, in the browser's Location: box.
- The GET method is restricted to send upto 1024 characters only.
- Never use GET method if you have password or other sensitive information to be sent to the server.
- GET can't be used to send binary data, like images or word documents, to the server.
- The data sent by GET method can be accessed using QUERY\_STRING environment variable.
- The PHP provides **\$\_GET** associative array to access all the sent information using GET method.

**The POST Method**

The POST method transfers information via HTTP headers. The information is encoded as described in case of GET method and put into a header called QUERY\_STRING.

- The POST method does not have any restriction on data size to be sent.
- The POST method can be used to send ASCII as well as binary data.
- The data sent by POST method goes through HTTP header so security depends on HTTP protocol. By using Secure HTTP you can make sure that your information is secure.
- The PHP provides **\$\_POST** associative array to access all the sent information using POST method.

## Question 5

**Q5a. What is a cookie? How to store and retrieve the values in cookie in PHP.**  
**Ans:**

A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.

### Create Cookies With PHP

A cookie is created with the `setcookie()` function.

#### Syntax

`setcookie(name, value, expire, path, domain, secure, httponly);`

Only the name parameter is required. All other parameters are optional.

```
<?php
$cookie_name = "user";
$cookie_value = "Pallavi Tawde";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); // 86400 = 1 day
?>
```

### Retrive Cookies With PHP

```
<html>
<body>
<?php
if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
    echo "Cookie '" . $cookie_name . "' is set!<br>";
    echo "Value is: " . $_COOKIE[$cookie_name];
}
?>
</body>
</html>
```

**Q5b. Write a short note on regular expressions in PHP.**

**Ans:**

Regular expressions are nothing more than a sequence or pattern of characters itself. They provide the foundation for pattern-matching functionality.

Using regular expression you can search a particular string inside a another string, you can replace one string by another string and you can split a string into many chunks.

PHP offers functions specific to two sets of regular expression functions, each corresponding to a certain type of regular expression. You can use any of them based on your comfort.

- POSIX Regular Expressions
- PERL Style Regular Expressions

**PHP's Regexp POSIX Functions**

PHP currently offers seven functions for searching strings using POSIX-style regular expressions –

Sr.No	Function & Description
1	<code>ereg()</code> The <code>ereg()</code> function searches a string specified by string for a string specified by pattern, returning true if the pattern is found, and false otherwise.
2	<code>ereg_replace()</code> The <code>ereg_replace()</code> function searches for string specified by pattern and replaces pattern with replacement if found.

**PHP's Regexp PERL Compatible Functions**

PHP offers following functions for searching strings using Perl-compatible regular expressions –

Sr.No	Function & Description
1	<code>preg_match()</code> The <code>preg_match()</code> function searches string for pattern, returning true if pattern exists, and false otherwise.
2	<code>preg_match_all()</code> The <code>preg_match_all()</code> function matches all occurrences of pattern in string.

**Q5c. Explain the following PHP/MySQL functions with example**

**i) `mysql_query()` ii) `mysql_error()`**

**Ans:**

**i) `mysql_query()`** : To create new table in any existing database you would need to use PHP function **`mysql_query()`**. You will pass its second argument with a proper SQL command to create a table.

`mysql_query(query, connection)` - Query handle for successful CREATE, SELECT, SHOW, DESCRIBE, EXPLAIN and other statements or FALSE on error.

**ii) `mysql_error()` : `mysql_error()`**:Returns the text of the error message from previous MySQL operation

**Example:**

```
<?php
$con=mysql_connect("localhost","root","");
if(!$con)
    die('could not connect:'.mysql_error());
if(mysql_query("create database college ",$con))
    echo"Database Created successfully";
else
    echo "Error creating database:".mysql_error();
?>
```

**Q5d. Create a "Student.php" page to insert the data in the table Student\_Detail(Rollno, Name, Marks)**

**Ans:**

```
<?php
$con=mysql_connect("localhost","root","");
if(!$con)
    die('could not connect:'.mysql_error());
if(mysql_query("create database college ",$con))
    echo"Database Created successfully";
else
    echo "Error creating database:".mysql_error();
mysql_select_db("college",$con);
$query="create table student (rollno smallint, name varchar(50), marks
decimal(5,2))";
if(mysql_query($query,$con))
    echo"Table Created successfully";
else
    echo "Error creating table:".mysql_error();
$query1="insert into student values(101,'pallavi',88)";
if(mysql_query($query1,$con))
    echo"Record 1 inserted successfully";
else
    echo "Error inserting record 1:".mysql_error();
mysql_close($con);
?>
```

**Q5e. What are superglobal arrays? Explain with example the use of \$GLOBALS and \$\_SERVER.**

**Ans: PHP Global Variables – Superglobals**

Some predefined variables in PHP are "superglobals", which means that they are always accessible, regardless of scope - and you can access them from any function, class or file without having to do anything special.

### **PHP \$GLOBALS**

\$GLOBALS is a PHP super global variable which is used to access global variables from anywhere in the PHP script (also from within functions or methods). PHP stores all global variables in an array called \$GLOBALS[index]. The index holds the name of the variable. The example below shows how to use the super global variable \$GLOBALS:

```
<html>
<body>
<?php
$x = 75;
$y = 25;
function addition() {
    $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}
addition();
echo $z;
?>
</body>
</html>
```

**Output:** 100

### **PHP \$\_SERVER**

\$\_SERVER is a PHP super global variable which holds information about headers, paths, and script locations. The example below shows how to use some of the elements in \$\_SERVER:

```
<html> <body>
<?php
echo $_SERVER['PHP_SELF']; echo "<br>";
echo $_SERVER['SERVER_NAME']; echo "<br>";
echo $_SERVER['HTTP_HOST']; echo "<br>";
echo $_SERVER['HTTP_REFERER']; echo "<br>";
echo $_SERVER['HTTP_USER_AGENT']; echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?> </body> </html>
```

**Output:**

/demo/demo\_global\_server.php

35.194.26.41

35.194.26.41

http://localhost/showphp.php?filename=demo\_global\_server

Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0

/demo/demo\_global\_server.php

### **Q5f. Write a short note on Sessions in PHP.**

#### **Ans: Sessions**

- A session is a way to store information (in variables) to be used across multiple pages.
- Unlike a cookie, the information is not stored on the user's computer.
- Session variables hold information about one single user, and are available to all pages in one application.
- A session creates a file in a temporary directory on the server.
- PHP first creates a unique ID for each session which is a random string of 32 hexadecimal numbers.
- A cookie called PHPSESSID is automatically sent to the user's computer to store unique session ID.
- Information stored in session variables will be deleted once the browser is closed.

#### **Starting a PHP Session**

A PHP session is easily started by making a call to the **session\_start()** function. This function first checks if a session is already started and if none is started then it starts one. It is recommended to put the call to **session\_start()** at the beginning of the page.

Session variables are stored in associative array called **\$\_SESSION[]**. These variables can be accessed during lifetime of a session.

Make use of **isset()** function to check if session variable is already set or not.

#### **Destroying a PHP Session**

A PHP session can be destroyed by **session\_destroy()** function. This function does not need any argument and a single call can destroy all the session variables. If you want to destroy a single session variable then you can use **unset()** function to unset a session variable.

#### **Example:**

```
<?php
    session_start();
    if( isset( $_SESSION['counter'] ) ) {
        $_SESSION['counter'] += 1;
    }else {
        $_SESSION['counter'] = 1;
    }
    $msg = "You have visited this page ". $_SESSION['counter'];
    $msg .= "in this session.";
?>

<html> <head>    <title>Setting up a PHP session</title>    </head>
    <body>
        <?php echo ( $msg ); ?>
    </body>
</html>
```