DWPD (3350702) Question-Bank

1. Write difference between HTML 4.0 and HTML 5.0

Sr. No	HTML 5	HTML 4
1	Syntax: It has Simplified and clear syntax.	Syntax: It's Syntax is little complex.
2	in HTML5 is extremely easy to get the user location	In HTML4, it was time consuming task to get the geographical locations of the visitors visiting the site.
3	HTML5 uses new structures such as drag, drop and much more	HTML 4 uses common structures like headers, footers using tag
4	HTML 5 can contain embedded video and audio without using flash player	HTML 4 cannot embed video or audio directly. HTML 4 makes use of flash player for it
5	It provides local storage in place of cookies.	Html4 use cookies.
6	HTML 5 is capable of handling inaccurate syntax	HTML 4 cannot handle inaccurate syntax
7	Using Html 5 you can draw shapes like circle, rectangle, triangle.	Using Html 4 it's not possible to draw shapes like circle, rectangle, triangle.

2. What is CSS? List Types of CSS and explain Inline CSS with example.

- **CSS stands for Cascading Style Sheets.**
- CSS describes **how HTML elements are to be displayed on screen, paper, or in other media**

❖ Use of CSS

1. It allows designer to separate HTML content from their design.
2. CSS provide Efficiency in Design & Updates.
3. CSS provides Faster page Downloads.
4. CSS provides consistency & uniformity.

❖ Type of CSS

Based on the location of css style there are three type of CSS available.

1. **Internal CSS/ Embedded CSS**
2. **External CSS**
3. **Inline CSS**

❖ Inline CSS

- An inline style may be used to apply a unique style for a single element.
- To use inline styles, add the style attribute to the relevant element. **The style attribute can contain any CSS property.**

Example

```
<html>
<head>
<title>HTML Inline CSS</title>
</head>
<body>
<p style="color:red;">This is red</p>
<p style="color:green;">This is green</p>
</body>
</html>
```

3. **Explain switch statement in PHP with example**

- Use the switch statement to **select one of many blocks of code to be executed**.
- Syntax

```
switch (n)
{
  case label1:
```

```
    code to be executed if n=label1;
    break;
  case label2:
    code to be executed if n=label2;
    break;
  ...
  default:
    code to be executed if n is different from all labels;
}
```

- **Example**

```
<?php
    $favcolor = "red";
switch ($favcolor)
{
case "red":
  echo "Your favorite color is red!";
  break;
case "blue":
  echo "Your favorite color is blue!";
  break;
case "green":
  echo "Your favorite color is green!";
  break;
default:
  echo "Your favorite color is neither red, blue, nor green!";
}
?>
```

4. What is variable? List rules for naming variable

- Variables are identifiers used to label storage in memory. In PHP, a variable starts with the $ sign, followed by the name of the variable. Variables are used to store data, like string of text, numbers, etc.
- **Example:**

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

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
- It 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)

5. What is Array? Explain types of array with example.

- An array is a special variable, which can hold more than one value at a time.
- If you have a list of items (a list of car names, for example), storing the cars in single variables could look like this:

```
$cars1 = "Volvo";
$cars2 = "BMW";
$cars3 = "Toyota";
```

- In PHP, there are three types of arrays:

  Indexed arrays

  Associative arrays

  Multidimensional arrays

- **Associative Arrays:**
- In associative array each elements having key associated with it.
- It can be either numeric or string.
- **Syntax:**

  $ArrayName = array(Key1=>Value1, Key2=>Value2, …, KeyN=>ValueN)

- In PHP, if we don't specify key value for each element then it will create a numeric array.
- **Example (Numeric Key)**

```
<?php
        $name = array(19=> "Aarav", 20=>"Kahaan");
         print_r($name);
?>
```

- **Output:**

```
Array ([19]=> Aarav [20]=>Kahaan)
```

6. Explain settype() and gettype() function with example

- **Gettype( ) :**

- In PHP, gettype( ) function is used to get the datatype of variable.
- **Syntax:**

```
gettype(variable name);
```

- **Example:**

```
<? php
$a=10;
echo gettype($a);
?>
```

- **Output:**

```
Integer
```

- **settype( ) function**

- In PHP, settype( ) function is used to set the datatype of variable.
- It returns Boolean value.
- If variable is set successfully to specific type then it returns 1 otherwise 0.
- **Syntax:**

```
settype(variable name, datatype);
```

- **Example:**

```
<?php
$a;
echo settype($a,"Integer");
echo gettype($a);
echo $a;
?>
```

- **Output:**

```
Integer
0
```

7. Explain mysql_connect() and mysql_query() functions

- **mysql_connect ( ) :**
- This function allows you to establish connection of PHP application with MySQL server.

- **Syntax :**

  $VariableName = mysql_connect (serverName, UserName, Password)
- **ServerName** : Indicates the name of the MySQL server with which you want to establish connection.
- **UserName** : Indicates name of the user using which you can logs on to MySQL server.
- **Password** : Indicates password of the user using which you can logs on to MySQL Server
- This Function returns a Boolean value TRUE or FALSE. If connection establish successfully with MySQL Server then this function returns true value otherwise it returns false.
- **Example :**

```
<?php
$con = mysql_connect ("localhost", "root") ;
if ($con)
{
econ "Connected with MySQL" ;
}
else
{
echo "Can not connect with MySQL" ;
}
?>
```

- **mysql_query ( ) :**
- This function allows you to specify and execute the MySQL command on MySQL Server.
- **Syntax :**

  Mysql_query ("Query", COnnectionName) ;
  - **Query** : Indicates the MySQL command to be executed.

- **ConnectionName** : Indicates the name of the variable that is used at the time of establish connection with MySQL server using mysql_connect ( ) function

- **Example** :

```
<?php
$con = mysql_connect ("localhost", "root") ;
$db = mysql_selectdb ("MyDatabase") ;
$cmd = mysql_query ("create table Test (ID integer, Name varchar (20))") ;
if ($cmd)
{
        echo "Table created Successfully" ;
}
else
{
        echo "Error while Executing query" ;
}
?>
```

8. What is Cookie? Explain with Example

- A cookie is a small piece of information that is stored either on the client computer's browser memory or in a file on the hard disk.
- A cookie having name and value.
- A cookie is useful for storing user specific information such as username, last visit etc.
- One can pass variables between pages by mean of cookies.
- **Syntax:** setcookie('Name', Value, ExpireTime)
- **Example:**

```
<?php
    $cookie_name = "user";
    $cookie_value = "Alex Porter";
    setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/");
```

```
?>
<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>
```

9. What is Session variable? Explain with Example

- 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 users computer.
- A session is started with the session_start() function.
- Session variables are set with the PHP global variable: **$_SESSION.**
- **Example:**

```
<?php
// Start the session
session_start();
?>
<html><body>
<?php
        // Set session variables
```

```
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>
</body></html>
```

- **Output:**

Session variables are set.

10. **Give difference between GET method and POST method**

GET Method	POST Method
Method data is sent from one page to other in the URL using query string format.	Using POST Method data is sent from one page to other within the body of the HTTP request
The GET method, appends name/value pairs to the URL.	POST method packages the name/value pairs inside the body of the HTTP request, which makes for a cleaner URL.
The length of a URL is limited so the GET method can send at most 1024 characters only.	POST method imposes no size limitations on the forms output.
Using GET method is insecure because parameters passed on the URL are visible in the address field of the browser.	Using POST method is more secure because no data is visible in the URL.
GET method can't be used to send binary data, like images or word documents, to the server.	The POST method can be used to send ASCII as well as binary data.
The data sent by GET method can be accessed using $_GET variable.	The data sent by POST method can be accessed using $_POST variable.

11. Explain passing by values and passing by reference to function

- **Call by Value**
- When we pass arguments to the function, the values of the passed arguments are copied into argument variables declared inside the argument list of function definition.
- So, called function works with copies of argument instead of original passed arguments.
- So any changes made to these variables in the body of the function are local to that function and are not reflected outside it.
- **Example**

```
<?php
        function addSix($num)
        {
                $num += 6;
        }
        $orignum = 10;
        addSix( $orignum );
        echo "Original Value is $orignum<br />";
?>
```

- **Output**

```
        Original Value is 10
```

- **Call by reference**
- It is possible to pass arguments to functions by reference.
- This means that a reference to the variable is manipulated by the function rather than a copy of the variable's value.
- Any changes made to an argument in these cases will change the value of the original variable.
- You can pass an argument by reference by adding an ampersand to the variable name in either the function call or the function definition.

- **Example**

```
<?php
        function addSix(&$num)
        {
                $num += 6;
        }
        $orignum = 10;
        addSix( $orignum );
        echo "Original Value is $orignum<br />";
?>
```

- **Output**

        Original Value is 16

12. Explain user defined function with default arguments

- A user defined function declaration starts with the word "function".
- **Syntax**

```
function functionName()
{
        code to be executed;
}
```

- **Default Arguments**
  - In PHP, default argument in function must be assign some value to the argument while defining it.
  - Default arguments must be written from right to left direction.
- **Syntax:**

```
funtion functionName( $argument1, $argument2=Value)
{
        Function Body
}
```

- **Example**

```
<?php
function total($a, $b = 50)
```

```
{
        echo "Total =  $a + $b<br>";
}
total(350);
total(10,65);
?>
```

13. **Explain passing of variable between pages using URL Method**

- The GET method passes arguments from in page to next page as a part of the URL Query String.
- When used for form handling, GET appends the indicated variable name and value to the URL designated in the ACTION attribute with a question mark separator.
- Each item submitted via GET method is accessed in the handler via $_GET array.
- **Example**

```
<html>
<body>
    <form action="welcome.php" method="GET">
        Enter your name: <input type="text" name="name" />
        Enter your age: <input type="text" name="age" />
        <input type="submit" value="OK" />
    </form>
</body>
</html>
```

The "welcome.php" file looks like this:

```
<html>
<body>
    Welcome <?php echo $_GET["name"]; ?>.<br/>
    You are <?php echo $_GET["age"]; ?> years old!
```

```
</body>
</html>
```

Output:

Welcome Sachin

You are 30 years old!

14. **Explain passing of variable between pages using Form collection Method.**

- POST method is the preferred method of form submission.
- The form data set is included in the body of the form when it is forwarded to the processing agent (web server).
- No visible change to the URL will result according to the different data submitted.
- Each item submitted vai POST method is accessed in the handler via the $_POST array.
- **Example**

Step1 : Create a web page named form1.php as shown below :

```
<html>
<head>
   <title> Processing form element </title>
</head>
<body>
   <form name="form1" method="post" action="form2.php" >
   Enter your name :
    <input type="text" name="username" />
    <input type="submit" name="submit" value="submit"  />
   </form>
</body>
</html>
```

Step 2 : Now create another webpage named form2.php that process the input entered in the form1.php as shown below :

```
?php
        $name=$_POST['username'];
        echo "Hello welcome ".$name;
?>
```