1) What is php ?

- PHP is an acronym for "PHP: Hypertext Preprocessor".
- PHP is a widely-used, open source scripting language.
- PHP scripts are executed on the server.
- PHP is free to download and use.

2) Explain Get and Post method. (or Explain Getting HTML form data into PHP script).

There are two ways for form handling. 1) get& 2) post

1. GET

- When we use get method, value will be passed in the URL.
- This method does not provide any security to the user.
- Syntax : < form method = “get” >
- To fetch the data from get method $_GET variable is used.

2. POST

- When we use post method, value will not pass in the URL.
- This method will provide security to the user by hiding the value from URL.
- Syntax : <form method = “post”>
- To fetch the data from post method $_POST variable is used.

Example : welcome.php

```
<form action="welcome.php" method="post">
Name: <input type="text" name="name">
<input type="submit">
</form>
<?php echo $_POST["name"]; ?>
```

//for get method program, write ‘get’ instead of post.

3) Explain Cookie with example.

- A cookie is a small file that the server sends on the user's computer.
- A cookie is used to identify a user.
- Cookie always resides on client side.

Create Cookie :

To create cookie, setCookie() function is used.

Syntax :setCookie(name, value, expire time*)*

Example :setCookie(username, abc, time() + 3000);

Retrieve a Cookie :

To retrieve a cookie, $_COOKIE variable is used.

Syntax : $_COOKIE[cookie_name]l;

Example : $_COOKIE [‘username’];

Delete a Cookie :

To delete a cookie, use the setCookie() function with an expiration date in the past:

Example :setCookie("user", "abc", time() - 3000);

Example :

```
<?php
    setCookie(‘user’, ‘abc’, time() +3000);
    echo "Value is: " . $_COOKIE[user];
?>
```

4) Explain File uploading in php.

- Please note that if you want to upload the file then you have to use the POST method into the form.
- Also note that enctype is compulsory for file uploading.
- PHP provides the $_FILE variable for file uploading the files.

Example :

Process.html

```
<form action="process.php" method="post" enctype="multipart/form-data">
<input type="file" name="uploadfile"/>
<input name="Submit" type="submit" value="Submit">
</form>
```

- The above file will display a form to upload the file.

process.php

```
<?php
$dir = 'C:/upload/';
$uploadfile=$dir.$_FILES['uploadfile'] ['name'];
if (move_uploaded_file($_FILES['uploadfile'] ['tmp_name'], $uploadfile))
 {
echo "File is successfullyuploaded.";
 }
else
{
      echo "File Uploading Failed";
}
?>
```

5) Explain Server variable in detail.

- $_SERVER variable is used to display server information in the web browser.
- $_SERVER variable returns the data in array.

Variable	description	output
$_SERVER['PHP _SELF']	displays the file name	server.php
$_SERVER['REQUEST_METHOD']	displays method name	GET/POST
$_SERVER['SERVER_PROTOCOL']	displays name and version of protocol	HTTP/1.1
$_SERVER['SERVER_NAME']	displays name of the server host	Localhost

$ SERVER['SERVER PORT']	displays Port number of the Server	80
$_SERVER['DOCUMENT_ROOT']	displays current directory	C:\XAMMP\

Example :

```
<?php
        foreach ( $_SERVER as $key=>$value
        {
        echo " $_SERVER[ '$key' ] $value";
        }
        ?>
```

6) Sending mail in php

- mail() function is used for sending the mail.

Syntax :

bool mail ($to , $subject ,$message [, $additional_headers [, $additional Parameters]];

- $to : email address on which you want to send the mail.
- $subject : subject to set as the email subject.
- $message : email body which is sent into the mail.
- $additional_header : optional argument, Cc or Bcc email
- $addional_parameter :optional argument, add image or files

Example :

```
<?php
    $to= 'ketan_2533@yahoo.com';
    $subject= 'PHP Testing mail';
    $message= 'This is testing mail from the PHP.';
    mail($to, $subject, $message);
?>
```

7) **Web Page :** document written in HTML (hypertext markup language) and translated by your Web browser is known as web page.
   - Web pages can either be static or dynamic.

- A web page can contain any type of information, and can include text, colour, graphics, animation and sound.

8) **Website :** A website is a collection of web pages (documents that are accessed through the Internet).
   - website can consist of one page, or of tens of thousands of pages, depending on what the site owner is trying to accomplish.

9) **Static website :** A website having static page & display the same information to all users is called Static web Site.
   - Static website is made up with simple HTML.
   - information remains the same, or static, for every viewer of the site.

➔ **Advantages of static websites :**
- Quick to develop
- Cheap to develop
- Cheap to host

➔ **Disadvantages of static websites :**
- Requires web development expertise to update site
- Site not as useful for the user
- Content can get stagnant

10) **Dynamic Website :** A· dynamic website is one that changes or customizes itself frequently andautomatically, based on certain criteria.
   - A website that have interaction with user and change information with each user is called Dynamic Website.
   - Dynamic Website is made with HTML, DHTML, CSS Client side Scripting, Server Side Scripting etc.

➔ **Advantages of dynamic websites :**
- Much more functional website
- Much easier to update
- New content brings people back to the site and helps in the search engines
- Can work as a system to allow staff or users to collaborate

➔ **Disadvantages of dynamic websites:**
- Slower *I* more expensive to develop
- Hosting costs a little more

11) Client side Scripting v/s Server Side Scripting

Server Side Scripting	Client side Scripting
Works in the back end which could not be visible at the client end.	Works at the front end and script are visible among the
Requires server interaction.	Does not need interaction with the server.
PHP, ASP.net, Ruby on Rails, ColdFusion, Python, etcetera.	HTML, CSS, JavaScript, etc.
Could effectively customize the web pages and provide dynamic websites.	Can reduce the load to the server.
Relatively secure.	Insecure

12) **Web Server :** A computer that delivers Web pages to browsers and other files via theHTTP protocol is known as web server.

- It executes serverside scripts (CGI scripts, JSPs, ASPs, PHP, ASP.NET etc.)
- There Two most widely Used Web Server

[1] Apache Server [2] IIS Server

[1] Apache Sever :

- Apache HTTP Server is an open source HTTP web server.
- It is used for Unix-like systems (BSD, Linux, and UNIX systems), Microsoft windows, and other platforms.
- The Apache HTTP Server is developed and maintained by Apache Software Foundation.

[2] IIS Sever :

- (Internet Information Services) Microsoft's Web server.
- IIS (Microsoft Internet Information Services or Server) is a set of Internet based services for Windows machines.
- IIS runs only under the Windows operating system.

13) **Explain HTTP & HTTPS Protocol.**

HTTP :

- (Hyper Text Transfer Protocol) The communications protocol used to connect to servers on the Web.
- Web browsers use HTTP protocol as default.
- HTTP is a "stateless" protocol.
- HTTP uses port 80, 8008, or 8080.

HTTPS:

- Short for Hypertext Transfer Protocol over Secure.
- HTTPS is a secure method of accessing or sending information across a web page.
- it is slower than HTTP.
- HTTPS uses port 443.

14) FTP:

- File Transfer Protocol.
- FTP is a standard network protocol used to transfer files fromone host to another host over a TCP-based network, such as the Internet.
- FTP is often secured with SSL/TLS.

15) ISP :

- Internet Service Provider.
- ISP is an organization that provides access to the Internet.
- An Internet service provider (ISP) is a business or organization that offers users access to the Internet and related services.
- They provide services such as Internet transit, domain name registration and hosting, dialup access, leased line access and colocation.
- Following is some of the service provided by ISP
  (1) Internet Access (2) E-mail (3) Web Hosting (4) Domain Registration

16) WEB HOSTING :

- Web hosting is a service that provides Internet users with online systems for storing information, images, video, or any content accessible via the web.
- Web hosts provide data space and connectivity to the Internet.
- Types of hosting:
    - Free hosting : provide free space on host.
    - Shared hosting : provide sharing space.
    - Reseller hosting : get large space and distribute to others.
    - Dedicated hosting : to get own server.
    - Collocated hosting : to install application in all others.
    - Application hosting :provide online purchasing.
    - Virtual Private Server hosting : to manage other users.

17) Virtual Hosting :

- Virtual hosting is a method that web servers use to host more than one domain name on the same computer and IP address.
- Two or more domains may have same IP addresses.
- Advantage :Cheapest Way of Web Hosting for Small Website.
- Disadvantage : virtual hosting is the inability to host multiple secure websites.

18) Multihoming :

- Multihoming is a technique to increase the reliability of the internet connection of an IP network.
- Two apparently independent links, from completely different ISPs may actually share a common transmission line.
- A network must have a public IP address.
- Disadvantage : network may not be effective.

19) Distributed Web Server :

- Distributed Web-server systems are groups of Webservers connected over a LAN orWAN and operating as a single entity to serve requests for Web resources from clientsover the Internet.

- distributed Web-server systems composed by multiple server nodes.
- distribute client requests among the multiple server nodes.
- LAN and WAN based systems are defined as locally and globally distributed Webserver systems respectively.
- The main aim of distributed Webserver systems is to achieve scalability to respond increasing demands.

20) Document Root :

- Document root is special folder on server for website home.
- It is the start point of website.
- All webpage &other file are stored in this folder.
- The document root is a directory (a folder) that is stored on your host's servers and that is designated for holding web pages.
- Examples :htdocs , httpdocs , html , public_html, web

21) Jquery

What is jQuery ?

- jQuery is usedto use JavaScript on your website.
- JQuery is a lightweight.
- "write less, do more"

What are Events?

- different visitor's actions that a web page can respond are called events.
- Executes when something happens.

Examples:

- moving a mouse over an element
- selecting a radio button
- clicking on an element

jQuery Selectors

- jQuery selectors allow you to select and manipulate HTML element(s).
- All selectors in jQuery start with the dollar sign and parentheses: $().

1) Element Selector (tag selector)
2) Class Selector
3) id Selector

1. **The element Selector**

- The jQuery element selector selects elements based on the element name.
- Example : $("p")

2. **The #id Selector**

- The jQuery #id selector uses the id attribute of an HTML tag to find the specific element.
- Example :$("#test")

3. **The .class Selector**

- The jQuery class selector finds elements with a specific class.
- Example : $(".test")

Example :

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#test").hide();
  });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p id="test">This is another paragraph.</p>
<button>Click me</button>
</body>
</html>
```

JQuery Effects

1) fadeIn()
2) fadeOut()
3) fadeTo()
4) fadeToggle
5) slideUp
6) slideDown()
7) slideToggle
8) hide()
9) show()

[note : values for effcts ('slow', 'fast' or miliseconds)]

JQuery Events

1) click() when mouse is clicked
2) dbclick() when mouse is double clicked
3) keypress() when key is pressed completely / same as keyup()
4) keydown() when key is pressed down
5) keyup() when key is released
6) submit() when click on submit button
7) change() when edit text
8) focus() when mouse control is on the widgets
9) blur() when click outside the widgets
10) load()when page or widget loads
11) resize() when window is resized
12) scroll() when click on scroll button
13) unlode() when page or widget unloads
14) **animate()** animate() method is used to create custom animations.
15) The $(document).ready() method allows us to execute a function when the document is fully loaded.

Common Format for effect & event program.

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").dblclick(function(){                //change event name here only
```

```
   $(this).hide();                    // change effect name here only
  });
});
</script>
</head>
<body>
<p>If you double-click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>
```

22) MySql

general steps for connectivity

1) Create connection using mysql_connect().
2) Select the database using mysql_select_db().
3) Create a query (insert, update, delete)
4) Execute a query using mysql_query().
5) Display the data as required. (by using loop or array)

common program

```
$con=mysql_connect("localhost","root","");
mysql_select_db("test",$con);
$query="insert into
empvalues('".$_POST['num']."','".$_POST['nm']."','".$_POST['des']."','".$_PO
ST['sal']."','".$_POST['date']."','".$_POST['dnm']."')";

$r=mysql_query($query,$con);
```

mySql functions(any 5 in detail)

1) mysql_connect() – connects to the server.
2) mysql_close() – close the connection
3) mysql_error() – displays the error message
4) mysql_errno() – displays the error number only
5) mysql_select_db() – selects the database from the server
6) mysql_query() – executes query (insert, update, delete, select)

7) mysql_fetch_array() – returns data in the array
8) mysql_num_Rows() - displays total number of rows in the table
9) mysql_affected_Rows() – Reports the number of rows modified by query
10) mysql_fetch_assoc() – returns data in the array
11) mysql_fetch_field() – fetches all the columns
12) mysql_fetch_object() – fetch only objects
13) mysql_fetch_row() – fetches selected rows by using select query
14) mysql_insert_id() – Gets the AUTO_INCREMENT value (if any) generated
15) mysql_num_fields() – returns total number of columns
16) mysql_result() – displays the result (true or false)
17) mysql_tablename() – displays name of the current table
18) mysql_list_tables() – displays all the table names of current database
19) mysql_list_fields() – displays all column's name
20) mysql_field_type() – displays datatype of the column
21) mysql_db_name() - displays current database name
22) mysql_db_query() – executes query of the database(create or drop database)
23) mysql_data_seek() - Jumps to a specific row.

23) **User Defined Functions**:

a) Argument function
b) Variable function
c) Return value function
d) Default argument function
e) Recursive function

1) argument function

```
<?php
functionsayHello($name,$age)
{
        echo "Hello $name, you are $age years old<br/>";
}
sayHello("aaa",27);
sayHello("bbb",29);
sayHello("ccc",23);
?>
```

2) default argument

```
<?php
functionsayHello($name="abc")
{
echo "Hello $name<br/>";
}
sayHello("aaa");
sayHello();//passing no value
sayHello("bbb");
?>
```

3) **variable function**

```
<?php
function add(...$numbers) //not supported in newer version
{
   $sum = 0;
foreach ($numbers as $n) {
      $sum += $n;
   }
return $sum;
}
echo add(1, 2, 3, 4);
?>
```

4) **return function**

```
<?php
function cube($n)
{
      return $n*$n*$n;
}
echo "Cube of 4 is: ".cube(4);
?>
```

5) recursive function

```
<?php
function display($number)
 {
if($number<=5){
echo "$number <br/>";
display($number+1);
   }
}

display(1);

?>
```

24) String Functions

Function Name	Syntax	Description	Example	Output
Chr()	String chr(intascii)	Return a character string	echo chr(97)	a
Ord()	Intord(string string)	Return the ascii value of the first character	echo ord(“A”)	65
strtolower()	strtolower(string str)	return string with lowercase	echo strtolower(“Welcome TO PHP”);	welcome to php
strtoupper()	strtoupper(string str)	return string with uppercase	echo strtolower(“Welcome TO PHP”);	WELCOME TO PHP
strlen()	Intstrlen(string string)	Return the length of the given string.	echo strlen(“welcome”)	7
ltrim()	string ltrim(string str[,string charlist])	Strip white space or other characters from the beginning of a string.	echo ltrim(“ welcome”)	welcome
rtrim()	string rtrim(string	Strip white space or other	echo rtrim(“ welcome ”)	welcome

	str[,string charlist])	characters from the end of a string.		
trim()	string ltrim(string str[,string charlist])	Strip white space or other characters from the beginning of a string.	echo trim(“ welcome ”)	welcome
substr()	string substr(stringstring,int start[,int length])	return part of a string	echo substr(“abcdef”, 2,-1)	cde
strcmp()	Intstrcmp(string str1, string str2)	return 0 if both are equal and <0 if str1 is less thanotherwise>0.	echo strcmp(“Hello”,”hello”)	-1
strcasecmp()	intstrcmp(string str1, string str2)	Binary safe case-insensitive string comparision. return 0 if both are equal and <0 if str1 is less thanotherwise>0.	echo strcmp(“Hello”,”hello”)	0
strpos()	intstrpos(string haystack,mixed needle [,int offset])	find the position of first occurrence	echo strpos(“abcdef abcdef”,’a’,1);	7
strrpos()	intstrpos(string haystack,mixed needle [,int offset])	find the position of last occurrence	echo strrpos(“abcdefabcdef”,’a’);	7
strstr()	string strstr(string haystack, string needle)	find first occurrence of a string	echo strstr(“abcdefabcdef”,’e’);	efabcdef
stristr()	string strstr(string haystack,	case-insensitive strstr()	echo strstr(“abcdefabcdef”,’E’);	efabcdef

	string needle)			
strrev()	string strrev(string string)	return reverse a string	echo strrev(“Hello”);	olleH
strval()	string strval(mixed var)	get a string value of variable	$str=123.34the; echo strval(“$str”);	123.34the
echo()	void echo(String arg1[,string …])	output one or more string	echo “Hello world”;	Hello world
print()	int print(string arg)	output a string	print “escaping character is done\”Like this\”.”;	escaping character is done ”Like this”
explode()	array explode(string seprator,string string[,int limit])	split a string by string	$str=”A B C D E”; $arr=explode(“ “,$str); foreach ($arr as $val) echo “ ”.$val	A B C D E
implode()	string implode(string glue,array pieces)	join array element with a string	$array=array(‘A’,’B’,’C’,’D’,’E’); $str=implode(“,”,$array); echo $str;	A,B,C,D,E
Join()	string join(string glue,array pieces)	Alias of implode()	$array=array(‘A’,’B’,’C’,’D’,’E’); $str=join(“,”,$array); echo $str;	A,B,C,D,E
md5()	string md5(string str[,boolraw_output])	calculate the md5(message-digest algorithm) hash of a string	$str=”hello”; echo md5($str);	

str_split()	array str_split(string string [,intspilt_le	convert a string to an array	$str="Hello Friends"; $arr1=str_split($str); print_r($arr1);	Array ([0] => H [1] => e [2] => l [3] => l [4] => o [5] => [6] => F [7] => r [8] =>i [9] => e [10] => n [11] => d [12] => s)
str_shuffle()	string str_shuffle(string str)	Randomly shuffle a string	$str="abcdef"; $shuffled=str_shuffle($str); echo $shuffled;	ceabdf
strcspn()	intstrcspn(string str1, string str2)	The strcspn() function is used to get the number of characters that occur between the start of the main string and the first occurrence of any character listed in the second string. -	echo strcspn('W3School.com', '.');	8
strspn( )	intstrspn (string str1, string str2)	Returns the length of the initial segment of str 1 which consists entirely of characters in str2.	$var = strspn("42 is the answer, what is the question ... ", "1234567890"); echo $var;	2

substr_compare()	intsubstr_compare (string main_str, string str, int offset [,int length [,boolcase_sensitivity]])	Binary safe optionally case insensitive comparison of 2 strings from an offset, up to length characters	echo substr_compare("hello student","hello people",0);	1
substr_count( )	intsubstr_count (string haystack, string needle)	Count the number of substring occurrences	echo substr_count("this is test","is");	2
ucfirst( )	string ucfirst (string str)	Make a string's first character uppercase	echo ucfirst("welCome to PHP");	WelCome to PHP
ucwords( )	string ucwords (string str)	Uppercase the first character of each word in a string	echo ucwords("welCome to PHP");	WelCome To PHP
str_replace()	mixed str_replace (mixed search, mixed replace, mixed subject [, int&count])	Replace all occurrences of the search string with the replacement string	$vowels= array("a","e","i","o","u","A","E","I","O","U"); $x = str_replace($vowels, "*", "Hello World of PHP"); echo $x;	H*ll* W*rld *f PHP

25) Math Functions

Function Name	Syntax	Description	Example	Output
abs()	number abs(mixed number)	Absolute value	echo abs(-4.2);	4.2
ceil()	float ceil(float value)	Round Frections up	echo ceil(4.3); echo ceil(9.987);	5 10
floor()	float floor(value)	Round fractions down	echo floor(4.3); echo floor(9.987);	4 9
round()	float round(float val[,int precision])	Round a float	echo " ".round(90.987,0); echo " ",round(90.987,2); echo " ",round(90.987,-1);	91 90.99 90
fmod()	float fmod(float x, float y)	returns the floating point reminder of dividing the dividend of the arguments	echo fmod(5.9,2);	1.9
min()	mixed min(number arg1, number arg2[, number ….])	find lowest value	echo min(2,3,1,7,5);	1
max()	mixed max(number arg1, number arg2[, number ….])	find highest value	echo max(array(2,5,10));	10
pow()	number pow(number base,number exp)	Returns based raised to the power of expression.	echo pow(2,2); echo pow(2,-2);	4 0.25

sqrt()	float sqrt(float arg)	return square root	echo " ".sqrt(9);	3
exp()	float exp(float arg)	calculates the exponent of e(base of the natural system of logarithms)	echo exp(5.7);	298.87
rand()	int rand(int min, int max)	Genrate a random integer	echo rand(); echo rand(5,20);	7713 14
cos()	float cos (float arg)	returns the cosine of the arg parameter. The arg pa1·ameter is in radians.	echo cos(M_PI);	-1
acos()	float acos (float arg)	Returns the arc cosine of arg in radians. acosO is the complementary function of cosO	echo " ".acos(-1);	3.14159 26535898
sin()	float sin (float arg)	returns the sine of the arg parameter.	echo " ".sin(60);	-0.30481 062110 222
asin()	float asin (float arg)	Returns the arc sine of arg in radians. asinO is the complementary functionqfsinO	echo " ".asin(0.60);	0.64350 110879 328
tan()	float tan (float arg)	tan() returns the tangent of the arg parameter.	echo” ”.tan(1);	1.55740 772465 49

atan()	float atan (float arg)	Returns the arc tangent of arg in radians. atanO is the complementary function of tanQ,	echo “ ”.atan(1);	0.78539816339745
bindec()	intbindec(string binary_string)	return Binary to decimal	echo bindec(“1111”);	15
decbin()	string decbin(int number)	return Decimal to Binary	echo decbin(248);	11111000
hexdec()	inthexdec(string hex_string	return hexadecimal to decimal	echo hexdec(“2f”);	47
dechex()	string dexhex(int number)	return decimal to hexadecimal	echo " ".dechex("248");	f8
is_finite()	boolis_finite (float val)	finds whether a value is a legal finite number	echois_finite(2) . " "; echo is_finite(log(0)) . " ";	1
is_infinite()	boolis_infinite (float val)	finds whether a value is an infinite number	echois_infinite(2) . " "; echo is_infinite(log(0)) . " ";	1
log()	float log(float arg[,float base])	return Natural logarithem	echo log(10);	2.302585092994
base_convert()	string base_covert(string number,intfrombase,inttobase)	convert a number between arbitrary bases	$x='A3'; echo base_convert($x,16,2);	10100011

deg2rad( )	float deg2rad(float number)	convert a number in degrees to the radian equivalent	echo deg2rad(90);	1.57079 632679 49

26) Miscellaneous Funtions

Function Name	Syntax	Description	Example	Output
Define()	bool define(string $name, mixed $value[,bool $case_insensitive=false])	Define a named constant at runrime.	define("CONSTANT","HELLO",true); echo CONSTANT;	HELLO
constant()	mixed constant(string $name)	Return the value of the constant indicated by name	define(“MAXSIZE”, 100); echo MAXSIZE; echo constant(“MAXSIZE”);	100 100
include()	include(string ‘file-name’)	includes and evaluates the specified file.	var.php: <?php $color="green"; $fruit="apple"; ?> echo "A $color $fruit"; echo " "; include("var.php"); echo "A $color $fruit";	A A green apple
require()	require(string ‘file-name’)	indentical to include except upon failure it will produce a fatal	echo "A $color $fruit"; echo " "; require("var.php");	A A green apple

		E_ERROR LEVEL error	echo "A $color $fruit";	
header()	void header(string [,bool$replace=true[,int $http_response_code]])	used to send a raw HTTP header.	header("location:file_fun.php");	
die()	die()	die() is equivalent to exit()	die()	exits from program

27) Array functions

- array_key_exists — Checks if the given key or index exists in the array
- array_keys — Return all the keys or a subset of the keys of an array
- array_merge_recursive — Merge one or more arrays recursively
- array_merge — Merge one or more arrays
- array_multisort — Sort multiple or multi-dimensional arrays
- array_push — Push one or more elements onto the end of array
- array_replace — Replaces elements from passed arrays into the first array
- array_reverse — Return an array with elements in reverse order
- array_search — Searches the array for a given value and returns the first corresponding key if successful
- array_shift — Shift an element off the beginning of array
- array_slice — Extract a slice of the array
- array_unique — Removes duplicate values from an array
- array_values — Return all the values of an array
- array — Create an array
- arsort — Sort an array in reverse order and maintain index association
- asort — Sort an array and maintain index association
- compact — Create array containing variables and their values
- count — Count all elements in an array, or something in an object
- current — Return the current element in an array
- each — Return the current key and value pair from an array and advance the array cursor
- end — Set the internal pointer of an array to its last element
- in_array — Checks if a value exists in an array

- key_exists — Alias of array_key_exists
- key — Fetch a key from an array
- list — Assign variables as if they were an array
- next — Advance the internal pointer of an array
- pos — Alias of current
- prev — Rewind the internal array pointer
- rsort — Sort an array in reverse order
- sizeof — Alias of count
- sort — Sort an array

28) File Handling Functions

- copy — Copies file
- delete — See unlink or unset
- fclose — Closes an open file pointer
- fgetc — Gets character from file pointer
- fgetcsv — Gets line from file pointer and parse for CSV fields
- fgets — Gets line from file pointer
- file_exists — Checks whether a file or directory exists
- file_put_contents — Write data to a file
- file — Reads entire file into an array
- filesize — Gets file size
- fopen — Opens file or URL
- fputcsv — Format line as CSV and write to file pointer
- fputs — Alias of fwrite
- fread — Binary-safe file read
- fseek — Seeks on a file pointer
- ftell — Returns the current position of the file read/write pointer
- fwrite — Binary-safe file write
- is_readable — Tells whether a file exists and is readable
- is_writable — Tells whether the filename is writable
- move_uploaded_file — Moves an uploaded file to a new location
- readfile — Outputs a file
- rename — Renames a file or directory
- rewind — Rewind the position of a file pointer
- unlink — Deletes a file

29) Date &Time Functions

- checkdate — Validate a Gregorian date
- date_add — Alias of DateTime::add
- date_create — Alias of DateTime::__construct
- date_date_set — Alias of DateTime::setDate
- date_format — Alias of DateTime::format
- date — Format a local time/date
- getdate — Get date/time information
- gmdate — Format a GMT/UTC date/time
- mktime — Get Unix timestamp for a date
- strftime — Format a local time/date according to locale settings
- strptime — Parse a time/date generated with strftime
- strtotime — Parse about any English textual datetime description into a Unix timestamp
- time — Return current Unix timestamp

30) Variable Functions

1) Gettype() – returns the datatype of variable
2) Settype() – sets the datatype of the variable
3) Isset() – returns true if variable exists
4) Unset() – returns true if variable does not exist
5) Strval() – converts the value into string
6) Floatval()- converts the value into float
7) Intval() - converts the value into integer
8) print_r() – displays the values of array with index

31) foreach loop

- The foreach loop is mainly used for looping through the values of an array.
- **Syntax :**

```
foreach (array as $value)
{
        //code to be executed;
}
```

- **Example :**

```
<?php
        $salary[]=2000;
        $salary[]=3000;
        $salary[]=5000;

                foreach($salary as $value)
                {
                        echo "Salary: $value<br>";
                }
?>
```

32) Gd_library (Graphics Design Library)

- gd_info — Retrieve information about the currently installed GD library
- getimagesize — Get the size of an image
- imagecolorallocate — Allocate a color for an image
- imagecreate — Create a new palette based image
- imagecreatetruecolor — Create a new true color image
- imageellipse — Draw an ellipse
- imagefill — Flood fill
- imagefilledarc — Draw a partial arc and fill it
- imagefilledellipse — Draw a filled ellipse
- imagefilledpolygon — Draw a filled polygon
- imagefilledrectangle — Draw a filled rectangle
- imageline — Draw a line
- imagerectangle — Draw a rectangle
- imagestring — Draw a string horizontally
- imagesx — Get image width
- imagejpeg — Output image to browser or file
- imagepng — Output a PNG image to either the browser or a file

33) Regular Expressions

a) ereg_replace — Replace regular expression
b) ereg — Regular expression match
c) eregi_replace — Replace regular expression case insensitive
d) eregi — Case insensitive regular expression match

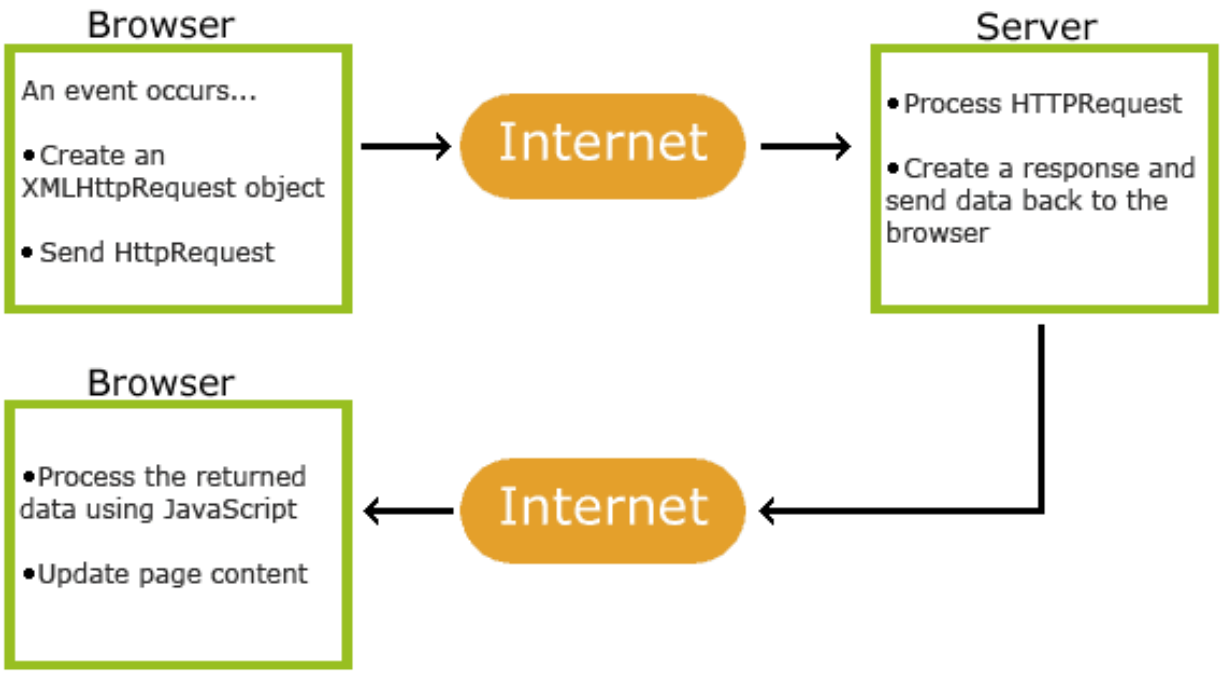
e) split — Split string into array by regular expression
f) spliti — Split string into array by regular expression case insensitive
g) sql_regcase — Make regular expression for case insensitive match

34) Ajax

What is AJAX?

- AJAX = Asynchronous JavaScript and XML.
- AJAX is a technique for creating fast and dynamic web pages.
- Not a stand-alone language or technology.
- It is a technique that combines a set of known technologies in order to create faster and more user friendly web pages.
- It is a client side technology.
- Examples of applications using AJAX: Google Maps, Gmail, Youtube, and Facebook tabs.

How Ajax Works ?



35) Session

- 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
- Four functions of session
    1) session_start() – starts the session
    2) session_id()- displays session id
    3) session_name() - displays session name
    4) session_destroy() – destroys the session

- **Example :**

```
<?php
        session_start();

        $_SESSION['username']="abc";
        $_SESSION ['type'] ="admin";

        echo "<br>Your session id is :- ".session_id();
        echo "<br>Your session name is :- ".session_name();
        echo "<a href=38-viewSession.php>Click to view Session
data</a>";
        session_destroy();
?>
```