

Client side : HTML, CSS
Server side : PHP, JSP, ASP

PHP is used to develop Web Applications.

PHP (Personal Home Page) (Hypertext Preprocessor)

PHP: PHP is a server side Scripting language implemented by Rasmus Lerdorf in 1995 using 'c' and 'perl' Technologies. By using PHP we can develop Dynamic Web Application.

Static Webpage: Only information displays. Ex: Newspaper.

Dynamic Webpage: We can login Ex: Gmail, Facebook.

PHP stands for Personal Home page and also it's contains it's alias name HtP i.e. Hypertext Preprocessor.

Rasmus Lerdorf implemented PHP 1.0 to find out how many companies accessing his Resume through online. He used 'c' and 'perl' Technologies for this implementation.

Apache: Open Source free Server.

IIS: .NET used that server.

Note: MySQL is a Open source and only PHP can directly contact with MySQL, So there is a combination of MySQL & PHP.

PHP Application : Facebook

ASP Application : ORKUT

Note: PHP Maintained by Zend Organization.

Date: 14 AUG 2012

* Features of PHP:

1) Cross platform: It can run under any types of operating system.

2) Cross Server: It can run under diff. types of web servers, like IIS, Apache, Tomcat etc.

3) Cross Database: It supports any type of Database server.

4) PHP 5.0 MySQL: PHP 5.0 is providing MySQL library to interact with MySQL DB.

5) PHP supports no. of object Oriented programming concepts like Inheritance, Access Specifiers etc.

6) PHP is Open Source.

7) PHP supports diff. types of content management s/w like JOOMLA, DRUPAL, WORDPRESS etc. All CMS (Content Management System) are open source s/w.

- 8) PHP providing number of security fns to apply security to applicatn.
 1-Way Encrypt?, 2-Way Encrypt?, Authentication etc.
- 9) PHP supporting diff. types of Editors to develop prog; we can also use lightweight Editors (Notepad, Editplus Etc)
- 10) PHP execute is very fast Bcoz, of Zend Engine.
- 11) Zend organizatn is maintaining the PHP script. They introduced number of tools to work with PHP & also they are providing complete support to PHP programmers thr. online.
- 12) PHP is very easy to understand & Develop the Applicatn.

Date: 16 AUG 2012

* Versions of PHP :

- 1) PHP 1.0 (1995):
 i) Rasmus Lerdorf implemented PHP 1.0 using 'C' & 'perl' S/W.
 ii) It is not a server side scripting lang.
 iii) Implementation from 'C' & 'Perl' softwares.
- 2) PHP 2.0 (1997):
 i) Rasmus Lerdorf implemented PHP 2.0 as partially server-side scripting language.
 ii) PHP 2.0 supports different types of Database servers.
 iii) We can implement some types of modules using 2.0 those are registration, login, etc.
- 3) PHP 3.0 (1998):
 i) PHP 3.0 is fully server side scripting language.
 ii) Zeev Suraski & Andi Gutmans rewritten the functionalities of PHP from scratch. On released it 3.0 as fully server side Scripting lang.
 iii) PHP 3.0 supports different platforms (Operating syst.)
 iv) Personal Homepage is Renamed as Hypertext Preprocessor with this version.
- 4) PHP 4.0 (2000):
 i) PHP 4.0 is cross server, it supports IIS, APACHE, TOMCAT etc.
 ii) Zend Engine 1.0 is introduced in PHP 4.0, Zend Engine is Runtime Environment of PHP Applicatn.
 iii) Smarty templating syst. is introduced with this Version, we can simply develop the Applicatn with predefined templates.
- 5) PHP 5.0 (2004-05):
 i) The major upgrade of PHP 5.0 is Object Oriented programming support.
 ii) Zend Engine 2.0 is introduced with PHP 5.0

iii) XML and Webservices related classes are introduced with PHP 5.0⁽³⁾

6) PHP 6.0 :

- i) The main focus of PHP 6.0 is Unicode support means programmers can able to name their classes and methods with their native languages.
- ii) Some Advanced Object Oriented programming concepts introduced with PHP 6.0 Those are Namespaces, and Late Static Binding.



- * PHP is partially case sensitive lang. In ^{fn} point of view it is case ^{sensitive} and Variables point of view, it is case sensitive.
- * Every statement in PHP, we should end with semicolon.
- * PHP is loosely typed language. That's why no need to provide Datatypes at the time of variable declaration.
- * Variable Names in PHP should starts with \$'(Dollar) symbol.
- * Every PHP file extension should be .PHP.
- * PHP script we should include within the Script Declaration style tags.
`<?php ----- ?>`

Date: 17 AUG 2013

Webserver:

Webserver is a s/w used to run the Web Application. It handles the request from client and sends the response to the client.

Ex: IIS, Apache, Tomcat etc.

Web Browser:

It is a s/w used to open web application from Webserver.

Ex: Internet Explorer, Mozilla Firefox, etc.

Request:

A Request is a Trip of Webpage from Browser to server.

Ex: www.gmail.com.

Response:

Server sends the opf for client request in the form of Response.

Client Side Script:

The script which executes client system with the help of Webbrowser can be called as Client-side Script.

Ex: HTML, Javascript etc.

Server Side Script:

The script executes ⁱⁿ the webserver with the help of Server engines. User can't see the original script of server side scripting languages.

Ex: PHP, ASP, JSP etc.

* Tools To Work With PHP:

1) WAMP:

(Stands for Windows APACHE MySQL and PHP), These tool supports Windows OS.

2) LAMP:

(LINUX APACHE MySQL and PHP). It can run under LINUX OS.

3) XAMPP:

(XBPF For any operating syst, APACHE, MySQL PHP & PERL)

All these tools are open source tools available with diff. extensions.

How to Run a prog. in other Drive?

* Installatⁿ of PHP: (XAMPP 1.7.2) → C:drive → XAMPP → apache → Conf → httpd → Ctrl+F → ~~httpd.conf~~ → httpd.conf

After Installatⁿ of XAMPP, we can find out XAMPP folder in the ~~starting~~ Destinalⁿ locatⁿ contains number of subfolders. Press 9 Times → DocumentRoot Root "F:/PHP"

htdocs:

It is the root directory of APACHE server, All php programs we should save at this locatⁿ.

→ Press 9 Times
Directory

tmp:

It is temporary mem. locatⁿ of server. All uploaded files and sessions will store at this locatⁿ.

php:

This folder contains php configuratⁿ settings. php.ini is configuratⁿ settings file of php.

apache:

apache contains apache configuratⁿ settings httpd.conf is apache configuratⁿ settings filename.

mysql:

mysql contains mysql configuratⁿ settings my.ini is configuratⁿ settings file.

* Steps to Create and Execute PHP program:

1) Open Editor and ^{implement} PHP Script using Script Declaratⁿ style tags.

2) Save this file in the root directory with extension: .php!

3) Open XAMPP control panel & start apache server.

4) Open Browser and send the request to the server to get the O/P of PHP file.

* Declaration style Tag of php: Date: 21-AUG-2013 (5)
php is providing different types of declaration style Tags.

1) Universal style Tag:

This Tag supports all functionality.

<?php.....?>

2) Short open Tag:

This Tag supports very few functionality.

<?.....?>

3) Asp style Tag:

We can add php script within asp declaratⁿ syntax.

<%.....%>

4) Script style Tag:

It is similar to javascript declaration syntax.

```
<script language = "php">  
.....  
</script>
```

We have 2 Configuratⁿ setting in php.ini related to declaratⁿ style Tag. The default value is "OFF" for short open Tags & asp tags. By changing the value we can execute this Tag. we need to Restart the server to see the changes.

* Output Functions in PHP:

1) Print: This fn displays output on Browser & Returns a Boolean value "True", if output is printed successfully, otherwise False. we cannot print multiple statements with single print statement.

[Note: In PHP, The value of True is "1" and False means "No Value"]

Ex: <?php

```
$a = print "ABC";  
print "<br/>";  
print $a;  
?>
```

2) Echo:

It same as print function. But we can print multiple statements with a single echo function.

Then print.

```
<?php  
echo "ABC", "DEF";  
?>
```

This fn doesn't return any value, that's why faster

2) var_dump:

This fn displays variable value along with its datatype.

Ex: <?php \$sno=111; var_dump(\$sno); ?>

3) printf:

By using this fn, we can display o/p with the help of format specifier.

Ex: <?php \$a=10; \$user = "scott"; printf("%s has %d cars", \$a, \$user); ?>

4) print_r:

By using this fn, we can display all elements of array & property of object.

* Types of Errors in PHP *

There are 4 Types of Errors.

1) Notice:

It is nothing but a small info to user. If we are trying to access undefined variables, the o/p is "Notice". Ex: <?php
\$x=100;
echo \$a;
echo \$y;
Notice doesn't stop script execution.

By default we cannot see the notice msg on browser. Bcoz, the config setting i.e. error_reporting value is "E_ALL" & E_NOTICE means display all error msgs except Notice. By removing E_NOTICE, the notice are displayed by the Browser. We can also display notice within the prog. by using Error_Report(E_ALL);

```
<?php error_reporting(E_ALL);  
$x=100;  
echo $x;  
echo $y;  
echo "NEXT";  
?>
```

2) Warning:

It stops the execn of webpage, from line where the error occurred. If we try to call undefined function Fatal Error occurs.

Ex: <?php

```
function f1()  
{  
    echo "hi";  
}  
f1()  
f2()  
?>
```

3) Fatal Error:

It stops the Execn of complete script if there is syntax mismatch.

Ex: <?php

```
echo "line1";  
echo "12"  
echo "13";  
?>
```

Note:

If the last line doesn't contain a semicolon, No Error occurs.

Data types available in PHP:

Data type is used to specify the type of data

What variable can hold.

Basically, we have 3 types of data types in PHP.

- 1) scalar
- 2) compound
- 3) special

scalar Data types again divided into diff. types.

1) Boolean:

These data types represents either true or false. In PHP, value of True is 1, and the value of False is Nothing.

Ex: <?php

```
$x=true;
echo $x;
?>
```

O/P: 12) is_bool(variable):By using this fⁿ, we can check whether the variable is boolean variable or not.Ex: <?php

```
$x=false;
echo is_bool($x); } // O/P: 1
$y="false";
echo is_bool($y); } // False: O/P -
```

O/P: 13) (bool) variable, (boolean) variable:

To convert the data type of variable into boolean data type.

<?php

```
$y="False";
$y=(bool)$y;
echo is_bool($y);
?>
```

O/P: 1

Integer:

This datatype stores numeric values.

Ex: <? php

```
$x = 123;
echo $x;
?>
```

O/p : 123

Is_int, is_integer:

Integer or Not.

<? php

```
$x = 123;
echo is_int($x);
?>
```

O/p : 1

Int (int) Variable, (integer) variable, int val variable:

By using this conversion fⁿ, we can convert a variable datatype into integer datatype.

Ex: <? php

```
$x = "123";
$x = intval($x);
echo is_int($x);
?>
```

O/p : 1

Floating:

This datatype represents decimal values.

is_float:

By using this fⁿ, we can check the \$/p value is float or Not.

float (variable):

By using this fⁿ, we can convert a variable datatype into a floating pt. number.

<? php

```
$x = 123.45;
echo is_float($x);
?>
```

O/p : 1

precision:

It is a configural⁹ setting in php.ini used to specify total no. of digits displayed in floating pt. no.

String:

String is collect⁹ of characters in php. In PHP, we can declare string variable in 3 ways.

- 1) Using single Quotat⁹s,
- 2) Using double Quotat⁹s,
- 3) Using Heredoc syntax.

Ex: <?php

```
$user = "scott";
$str = "Welcome to $user"; <br/>;
$str1 = 'Welcome to $user';
echo $str;
echo $str1;
?>
```

SAVE AS: String.php

O/P: Welcome to scott
Welcome to scott

Note: If we place a variable within double Quotat⁹, It returns the value of that variable.

Heredoc:

By using the syntax we can display the HTML elements in .php script. It avoids the problems what were getting with open quotat⁹s of string. This syntax doesn't require any open quotat⁹s that's why for internal values we can use Both single & double quotat⁹s.

The Syntax of Heredoc is : →

~~heredoc opening~~ <<< nameofstring
 /// content
 nameofstring

Ex: <?php

```
echo <<<mystring
<input type="button" value='Go'>
mystring;
?>
```

SAVE AS: Heredoc.php

O/P:

If we not write these 2 stat. It gives Error

parse error: syntax error,
unexpected '<' in
prog

* compound Datatypes:

1) Array: collectⁿ of Elements.

2) Object: Instance of a class.

* Special Datatypes:

Resource datatype:

It refers the external Resources like Database Connⁿ, FTP Connⁿ, File ptrs etc.

Ex: <?php

```
$con=mysql_connect("local host","root","");
echo $con<br/>;
echo
get_resource_type($con);
?>
```

SAYE AS:

ResourceDatatype.php

O/P:

Resource id #2
mysql link

Null Datatype:

In php, null is not a value we can consider a variable as Null variable based on 3 condⁿ.

- i) IF the variable is not set with any value.
- ii) IF the variable is set with Null value.
- iii) IF the value of variable is unset.

is_null:

By using this fⁿ, we can check whether the variable is Null or Not.

<?php

```
$x=100;
unset($x);
echo is_null($x);
?>
```

Note: We can unset the variable value by using unset fⁿ.

SAYE AS: ISNULL.php

O/P: 1

Array:

Array is collectⁿ of heterogeneous (dissimilar) datatypes.
php is loosely typed lang. That's why we can store any type of values in Arrays.

Normal variable can store single value, Array can store multiple values.

Array contains no. of elements, Each element is a combinatⁿ of element key & element value.

The key of 1st element is '0' and last element is 'total elements - 1'.

Syntax of Array Declaration:

11

```
variablename = array (ele1, ele2, ele3....)
```

Ex: `arr = array(10, 20, 30)`

1) <?php
 \$arr = array (10, 20, 30);
If we want to see output on the screen
then prog. will be <?php

?> OPP: Blank existing OPP print_r(\$arr)

2) <?php
 For printing use print
 \$arr = array (10,"scott",30);
 ?> SAVE AS: Array1.php

?) OPP: Blank
 For printing OPP
 use print_r(\$arr)

3) <?php
 for (\$i = 0; \$i < 10; \$i++) {
 echo "The value is " . \$i . "
 } // End of for loop.
 // Output:
 // The value is 0
 // The value is 1
 // The value is 2
 // The value is 3
 // The value is 4
 // The value is 5
 // The value is 6
 // The value is 7
 // The value is 8
 // The value is 9
 // End of script.

OIP: Scott.

We can create Array with Explicit Keys

23 AUGUST 2012

58 PHP

```
$ arr = array(10, 0 => "scott", 2 => 30);
print_r($arr);
```

SAVE AS: Array3.php

Op: Array ($\text{lo} \Rightarrow \text{lo[1]} \Rightarrow \text{sort}$)
 $\quad \quad \quad [\text{2}] \Rightarrow \text{lo[0]}$)

Note: If we don't provide an Explicit key to an Array Element, the new key is depending upon previous keys of that Element. First, It collects all previous keys finding \max^m key, then add it to the \max^m key & provide the resultant value as key to the new element.

L² ph

```
$ arr = array(10, 100 => "Scott", 30);  
print_r($arr);
```

3

```

obj: Array
([0] ⇒ 10
[100] ⇒ scott
[101] ⇒ 30
)

```

Ex: <?php

```
$arr=array(10,20,0⇒30);
print_r($arr);
?>
```

Save As: Array 4.php

Op: Array ([0]⇒30 [1]⇒20)

(12)

Ex: <?php

```
$arr=array();
$arr[0]=10;
$arr[1]=20;
print_r($arr);
?>
```

Save As: Array 5.php

Op: Array ([0]⇒10 [1]⇒20)

Ex: <?php

```
$arr=array();
$arr['Lang']='PHP';
$arr['Manager']='Scott';
print_r($arr);
?>
```

SAVE AS: AssociativeArray.php

Op: Array ([Lang]⇒PHP [Manager]⇒
Scott)

We can provide string as Array Element ^{key}, This concept is called as Associative Array.

COUNT()

This f() Returns total no. of Array Elements.

<?php

```
$arr=array(10,20,30);
echo count($arr);
?>
```

Save As: Count.php

Op: 3

SORT()

This f() arranges the Array Elements in Ascending order with new keys.

SAVE AS: sort.php

Op: ArraySort

[0] = 5

[1] = 10

[2] = 20

[3] = 30

)

Ex: <?php

```
$arr=array(10,5,20,30);
sort($arr);
print_r($arr);
?>
```

Ex: <?php

```
$arr=array(10,'a'⇒5,20,30);
sort($arr);
print_r($arr);
?>
```

⇒ same op as above

rsort()

rsort: Arranges Array Elements Descending order. With the new keys.

SAVE AS: SortDesc.php O/P: Array [0] ⇒ 30

[1] ⇒ 20

[2] ⇒ 10

[3] ⇒ 5

↑
Prog.

<?php
\$arr = array(10, 5, 20, 30);
rsort(\$arr);
print_r(\$arr);
?>

aSort()

```
<?php
$arr = array(10, 'a' ⇒ 5, 20, 30);
asort($arr);
print_r($arr);
?>
```

O/P: 9 ⇒ 5
0 ⇒ 10
1 ⇒ 20
2 ⇒ 30

arsort()

Descending order with original keys.

kSort()

Ascending order based on Element keys.

<?php
\$arr = array(10, 100 ⇒ 5, 20, 5 ⇒ 30),
ksort(\$arr);
print_r(\$arr);
?>

O/P: Array

krsort()

Descending order based on Array Keys.

array_sum()

To get the sum of Array Elements.

Ex: <?php

```
$arr = array(10, 20, 30);  
echo array_sum($arr);  
?>
```

O/P: 60

Ex: <?php

```
$arr = array(10, 20, 30, 'Scott');  
echo array_sum($arr);  
?>
```

array_product()

Get the product of elements.

Ex: <?php

```
$arr = array(10, 20, 30)  
echo array_product($arr);  
?>
```

array_push:

By using this fn, We can add new element at end pt. of Array & it returns total no. of Array Elements.

x: <? php

```
$arr = array(10, 20, 30);
echo array_push($arr, 40);
?>
```

O/P : 4

x: <? php

```
$arr = array(10, 20, 30);
echo array_push($arr, 40);
print_r($arr);
?>
```

array_pop:

Deletes the last element of Array and returns the value of the element.

x: <? php

```
$arr = array(10, 20, 30);
echo array_pop($arr);
?>
```

O/P : 30

x: <? php

```
$arr = array(10, 20, 30);
echo array_pop($arr);
print_r($arr);
?>
```

array_shift:

This fn removes the 1st element of an Array & Returns the value of that Element.

x: <? php

```
$arr = array(10, 20, 30);
echo array_shift($arr);
print_r($arr);
?>
```

array_unshift:

By using this fn we can add an element at the beginning of Array & Returns the total no. of Array Elements.

Ex: <? php
 \$arr = array(10, 20, 30);
 echo array_unshift(\$arr, 5);
 print_r(\$arr);
 ?>

array_combine:

This fn creates a new array by combining the elements of 2 arrays. New array keys are values of Ist Array. New Array values are values of IInd Array.

Ex: <? php
 \$arr1 = array(10, 20, 30);
 \$arr2 = array('a', 'b', 'c');
 \$narr = array_combine(\$arr1, \$arr2);
 print_r(\$narr);
 ?>

Op: Array
 ([10] = a
 [20] = b
 [30] = c)

array_diff:

This fn compares the values of 2 Array Elements & Returns the differences from Ist Array.

Ex: <? php
 \$arr1 = array(10, 20, 30);
 \$arr2 = array(10, 30, 40);
 print_r(array_diff(\$arr1, \$arr2));
 ?>

Op:
 Array ([1] => 20)

array_diff_assoc:

It compares key and values of 2 Arrays & Returns the differences from Ist Array.

Ex: <? php
 \$arr1 = array(10, 20, 30);
 \$arr2 = array(10, 30, 40);
 print_r(array_diff_assoc(\$arr1, \$arr2));
 ?>

Op:
 Array ([1] => 20)
 Array ([2] => 30)

array_diff_key:

compares 2 Array keys & Returns the differences from Ist Array

Ex: <? php
 \$arr1 = array(10, 20, 30);
 \$arr2 = array(10, 20, 40);
 print_r(array_diff_key(\$arr1, \$arr2));
 ?>

Op:
 Array ([1] => 20)

Explode:

(16)

This fn divides a string as Array Elements based on I/p value.

Ex: <?php

```
$str = "Welcome";
$arr = explode("e", $str);
print_r($arr);
```

?>

O/p:

```
Array
([0] => w
[1] => lcom
)
```

implode:

This fn combines array elements as a string.

Ex: <?php

```
$arr = array(10, 20, 30);
echo implode("@", $arr);
?>
```

O/p: 10@20@30
?> O/p: 10@/20@/30

extract:

This fn divides the associative array elements as variables from Array.

Ex: <?php

```
$arr = array('a' => 10, 'b' => 20, 'c' => 30);
extract($arr);
echo $a;
echo $b;
?>
```

O/p:
a = 10
b = 20

list:

24 AUGUST 2019

By using this fn we can ascend array elements into the variables.

Ex: <?php

```
list($x, $y) = array(10, 20);
echo $x;
echo $y;
?>
```

O/p:
10 20

in_array:

By using this fn we can check whether the s/p value is existed in the specified Array or not. If the value is existed it returns True, otherwise it returns False.

Ex: <?php

```
$arr = array(10, 20, 30);
echo in_array(20, $arr);
?>
```

O/p: 1

array_search:

This fn Searches the \$ip value is existed in specified Array or not & Returns key of that Element.

Ex: <?php

```
$arr = array(10, 20, 30);
echo array_search(20, $arr);
?>
```

Op: 1

array_rand:

By using this fn, We can get the keys of Arrays randomly.

Ex: <?php

```
$arr = array('b'=>10, 'a'=>20, 'c'=>30);
print_r(array_rand($arr, 2));
?>
```

array_slice:

By using this fn, We can get some part of an Array.

Ex: <?php

```
$arr = array(10, 20, 30, 40);
print_r(array_slice($arr, 0, 3));
?>
```

array_count_values:

This fn returns an Array with no. of occurrences for each value.

Ex: <?php

```
$arr = array(10, 20, 10, 40);
print_r(array_count_values($arr));
?>
```

array_key_exists:

By using this fn we check whether the \$ip key is existed in the specified array or not.

Ex:

```
<?php
$arr = array(10, 'a'=>20, 40);
echo array_key_exists('a', $arr);
?>
```

array_keys:

This fn returns all the keys of an array as a new array.

Ex: <?php

```
$arr = array('x'=>10, 'a'=>20, 'b'=>40);
print_r(array_keys($arr));
?>
```

array_merge:

To merge the elements of 2 Arrays.

Ex: <?php

```
$arr = array(10, 20, 30);
$arr1 = array(40, 50);
$na = array_merge($arr, $arr1);
print_r($na);
```

array_reverse:

This fn displays array elements in the Reverse dir.

Ex: <?php

```
$arr = array (10,5,20);
print_r(array_reverse($arr));
?>
```

array_values:

Returns the all values of array as a new array.

Ex:

```
<?php
$arr = array (10,'a'=>5,20);
print_r(array_values($arr));
?>
```

array_shuffle:

This fn shuffles the elements of an Array.

Ex: <?php

```
$arr = array (10,'a'=>5,20);
shuffle ($arr);
print_r ($arr);
?>
```

★ Some Server Configuration Settings:★) document_root directory:

By using this configuratⁿ setting we can change the document root path of Webserver. By default the root directory of Webserver is htdocs. We can change that locatⁿ using this configuratⁿ setting. This settings are available in server configuratⁿ setting file. i.e. httpd.conf.

★) listen,servername,localhost:

By using this configuratⁿ setting, we can change the portno. of Apache server.

By Default, Server runs with portno.80.

★) directory index:

By using this configuratⁿ setting we can specify the startup filenames.

By default, index, home & default comes under startup files.

★) isset:

By using this fn we can check whether the variable is set with any value or not.

```
<?php
$x=100;
echo isset($x);
$y;
echo isset($y);
?>
```

In php, Dot(.) is a concatenation operator used to join a string with another datatype value.

```
<?php
    $x=" Scott";
    $y=" John";
    $z=$x.$y;
    echo $z;
?>
```

<http://localhost>

25 - AUGUST - 2012

include :

By using this concept we can include a php script in another script. If we include an external script in the current php script we can access fn's, classes, variables and constants of external script from current php script. To include the script, different types of fn are available those are

- 1) include
- 2) include_once
- 3) require
- 4) require_once

1) include :

This fn includes a file no.of times, if the included file is not available, it returns a warning msg and executes rest of the statements.

Ex: <?php Page1

```
echo "from Page1";
include "Page2.php";
echo $sno;
echo constant("city");
fun1();
?>
```

Page2:

```
<?php
echo "Script from page2";
$sno=100;
define("city", "hyderabad");
function fun1()
{
    echo "This is from function";
}
```

2) include_once:

It is same as include But it includes external file only one time. First it will check whether the specified file is already included or not. If it is not included, then only includes external file.

Ex: <?php

```
echo "from page1";
include_once "page2.php";
include "page2.php";
?>
```

3) require:

It's same as include but it returns fatal errors if external file is not available & stops the execution of script.

Ex: <?php

```
echo "from Page1";
require "Page2.php";
echo "Next";
?>
```

4) require_once:

It is same as require, But only one time it includes the external file

?> <?php

```
echo "from Page1";
require_once "Page2.php";
?>
```

* Types of Variables In PHP *

? A Variable is the name of mem. locatⁿ used to store values at the time of program Executⁿ.

PHP is loosely typed lang. that's why we can store create variables without datatypes.

Types of Variables:

1) Local Variable:

Variable declaratⁿ within the fⁿ comes under Local variable declaratⁿ. Local Variables we can access within the fⁿ where we declared. We cannot access from outside the fⁿs.

Ex: <?php

```
error_reporting(E_ALL);
Function fun1()
{
    $x=200;
}
function fun2()
{
    echo $x;
}
fun1();
fun2();
?>
```

O/P : Undefined Variable x in
F:\PHP\Local Variable.php
on line 9.

2) Global Variable:

variable declared in global locatⁿ means outside the ass.

Functions. Global Variables we can access from any fⁿ within the script.

By default, we can not access Global Variables from the fⁿs directly.

If u want to access Global Variables, Use "`$GLOBALS`" keyword or

Redeclare the variable within the fⁿ as Global Variable.

Ex:

```
<?php
//error_reporting(E_ALL);
$sno=100;
function fun1()
{
    $sno=111;
    echo $GLOBALS['sno'];
    echo $sno;
}
function fun2()
{
    global $sno;
    $sno=200;
    echo $sno;
}
//fun1();
echo $sno;
fun2();
echo $sno;
?>
```

If we remove the
comment of fun1();
Then O/P will be

O/P: 100 200 200

Variables_Variable:

If we assign variable name as the value of another variable, comes under variables_variable fⁿ.

Ex: <?php

```
$x="scott";
$y='x';
echo $$y;
?> $y=$x
∴ $x=scott
```

O/P: scott

Static Variables:

22

Static Variables can maintain the previous values
We can assign the values only one time into static variable.

Ex: <?php

```
function fun1()
{
    Static $x=100; // * If we don't write static here, Then the
    $x++;
    echo $x;
}
fun1(); //101
fun1(); //102
fun1(); //103
?>
```

O/P: 101 102 103
So, we use static keyword. *

O/P: 101 101 101

Reference Variables:

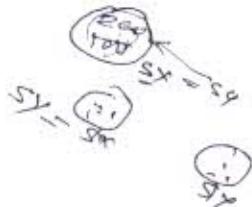
It refers the value of another variable.

Actual Variable and Reference Variable refers the value of same address locat?

Ex: <?php

```
$x=100;
$y=&$x; // O/P: 100
$y=200;
echo $x;
?>
```

O/P: 200



Super Global Variables:

PHP is providing no. of Super Global Variables
We can access this Super Global Variables from any locat?(from any webpage) within the project.

Different types of Super Global Variables are available. Those are
\$_GET, \$_POST, \$_REQUEST, \$_SERVER, \$_SESSION, \$_COOKIE, \$_FILES, \$_ENV.

All Super Global Variables are array datatypes

FORM

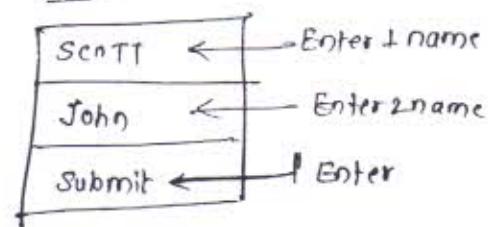
In HTML, Form is a container can hold S/I/P controls (Textbox, Radiobutton etc). If U want to x/ferr the values of S/I/P controls from 1 page to another page, When User click on SUBMIT Button we can go for FORM tag.

Attributes of Form Tag:

27-AUG-2012

- 1) ACTION: By using this attribute we can specify the pagename in which page we can redirect form values.
- 2) METHOD: By using this attribute we can specify the type of method what we are using to x/ferr the form values. The possible methods are "GET" and "Post".
- 3) ENC TYPE: By using this attribute we can specify the encoding type of form data.

Ex: `<Form Action="page2.html" method="POST">`
`<input type="text" name="t1">`
`
`
`<input type="text" name="t2">`
`
`
`<input type="submit" value="Send">`
`</form>`

Off:

We need to provide name property to send the values of form from one page to another page.

Ex: `<Form action="page2.html" method="get">`
`<input type="text" name="H1">`
`
`
`<select name='d1'>`
`<option> PHP </option>`
`<option> ASP </option>`
`<option> JSP </option>`
`</select>`
`<input type="Submit" value="Send" name="sub">`
`</form>`

* Difference Betⁿ GET and Post:

24

GET

- i) GET x'fers the informⁿ thr. http head locatⁿ & displays the data on URL address.
- ii) GET is unsecured.
- iii) GET x'fers limited amt of data.
- iv) ~~GET~~ Head can't upload the file.

POST

- i) Post x'fers the informⁿ thr. document body.
- ii) Post is highly secured.
- iii) Post x'fers ^{huge} amt of data.
- iv) Post can upload the files.

\$_GET:

By using \$_GET we can get the posted values of GET method. It is Array Variable the total no. of elements are equal to the total no. of posted values.

\$_POST:

To Get the posted values of Post method.

Ex: <FORM METHOD = "get" action = "page1.php">
<input type = "text" name = 't1'>

<input type = "text" name = 't2'>

<input type = "submit" name = 'sub' value = 'click'>
</form>

```
<?php  
 1 print_r($_GET);  
 1 echo $_GET['t2'];  
 echo $_GET['sub'];  
 ?>
```

```

<?php
if (isset($_POST['sub'])) // if (isset
{
    echo "This is form PHP";
    $t1 = $_POST['t1'];
    $t2 = $_POST['t2'];
    echo "Value of first is: $t1";
    echo "Value of second is: $t2";
}
?>

<FORM METHOD="POST" action="pro.php">
<input type="text" name='t1'>
<br>
<input type="text" name='t2'>
<br>
<input type="submit" name='sub' value='Click'>
</form>

```

28 Aug 2013

i) \$_request:

By using this super global variable we can ~~change~~ get the posted values of get method, and post Method. We can also get the values of COOKIES and query string

```

<form method="get" action="">
<input type="text">
<input type="text">
<input type="submit" value="click">
</form>

```

ii) Query string:

It is a small amt of data on URL address followed by "?"
If u want to xfer some xtra data along with the form data we can go for Origin. We can pass query string in 2 ways.

- 1) query string with name and value
- 2) query string with only value.

By using \$_request we can read the query string which contains name & value.

Ex: <?php

```

if(isset($_REQUEST['sub']))
{
    echo "value is";
    $_REQUEST['t1'];
    echo $_REQUEST['sno'];
} echo $_REQUEST['uname'];

```

```
<form method="post" action="abc.php? Sno=100 & uname=scott">
<input type="text" name="t1" value="Enter name">
<br>
<input type="submit" value="click" name="sub">
</form>
```

2) <?php

```
if (isset($_REQUEST['sub'])) {
    $val = $_POST['t1'];
    echo $val;
}
<form method="post" action="abc.php?
Sno=100 & uname=scott">
<input type="text" name="t1" value="<?php echo $val?>">
<br>
<input type="submit" value="click" name="sub">
</form>
```

3) <?php

```
if (isset($_POST['sub'])) {
    echo $_POST['sub'];
}
<form method="post" value="">
<input type="submit" name="sub" value="sub1">
<input type="submit" name="sub" value="sub2">
</form>
```

Assignment:ExpectedOp:

25

No1: No2: Res: + - *

```

<FORM Method = "POST" Value = " "
<input type = "text" name = 't1' >
<br>
<input type = "text" name = 't2' >
<br>

<form action = " " type = "post" >
<input type = "text" name = 't1' value = "<?php echo $val1 ?>" > <br>
<input type = "text" name = 't2' value = "<?php echo $val2 ?>" > <br>

<Form method = "POST" action = " " >
No1: <input name = 't1' value = "<?php
<br> echo $no1?>" >
No2: <input name = 't2' value = "<?php
<br> echo $no2?>" >
Res: <input type = 'Res' value = "<?php
<br> echo $res?>" >
<input type = "submit" name = 'sub'
      value = '+' >
<input type = "submit" name = 'sub'
      value = '-' >
<input type = "submit" name = 'sub'
      value = '*' >
<br>

```

?>

<?php
 if (isset(\$_POST['sub']))
 {
 \$no1 = \$_POST['t1'];
 \$no2 = \$_POST['t2'];
 if (\$_POST['sub'] == '+')
 \$res = \$no1 + \$no2;
 if (\$_POST['sub'] == '-')
 \$res = \$no1 - \$no2;
 if (\$_POST['sub'] == '*')
 \$res = \$no1 * \$no2;
 }

* $\$_FILES$:

By using this super global variable we can get the infoⁿ of uploaded files. It is 2 dimensional array variable providing 5 elements. Every element used the complete infoⁿ about uploaded file.

Each element first dimension is name of upload control

29-AUGUST-2012

Q: WAP to create 2-Dimensional Array.

⇒ <?php
 \$arr = array (100 => array (1,2,3), 20);
 echo \$arr [100][1];
 ?>

O/P : 2

If user upload any file from Browser to server first that file will upload into server temporary mem. locatⁿ. We need to implement server side script to move that file from temporary mem. locatⁿ to permanent locatⁿ.

In Temporary mem. locatⁿ it saves the file with another name.

* $\$_FILES$

⇒ $\$_FILES$ contains 5 elements. Each element's first dimⁿ is name of the upload control.

1) $\$_FILES['nameofuploadcon']['size']$: By using this element we can get the uploaded file size in Bytes.

2) $\$_FILES['nameofuploadcon']['name']$: To get the uploaded filename

3) $\$_FILES['nameofuploadcon']['type'][MIME]$: To get the uploaded filetype.

4) $\$_FILES['nameofuploadcon']['tmp_name']$:

5) $\$_FILES['nameofuploadcon']['error']$: To get the temporary filename.
 To get error noⁿ if any occurred at the time of file uploading.

MIME:

(29)

(MIME stands for Multipurpose Internet Mailing Extension)
It is a type of extension, used to x'ferr the files from one locat' to another locat'. User click on submit button.

By default FORM can x'ferr ~~bas~~ text format HTML data.
If you want to x'ferr another format file, we need to specify MIME type of that file.

Some of the MIME types are:

exe: application/octet-stream

jpg: image/jpeg

pdf: application/pdf

<form> Tag contains "enc" type attribute. By using this we can specify the MIME tag of input file.

multipart/form-data: This <MIME> Tag supports any type of file

* is_uploaded_file(\$FILES['Fileupcon']['tmp_name']):

By using this f?

We can check whether the file is uploaded from client syst. To Server Temporary locat' or not.

* move_uploaded_file:

By using this, We can move the uploaded file from temporary locat' to permanent locat'. It contains 2 Arguments temporary filename and permanent locat' path

Ex: → <form method="post" action="upload.php" enctype="multipart/form-data">
<input type="file" name="file1">

<input type="submit" value="send">
</form>

<?php

```
if(is_uploaded_file($FILES['file1file1']['tmp_name'])) save: upload.php
{
    $fname = $FILES['file1']['name'];
    if(move_uploaded_file($FILES['file1']['tmp_name'], "uploads/$fname"))
        echo "File is Moved";
    else
        echo "Not moved";
```

```
else  
echo "Not";  
?>
```

(30)

* Configuration Settings Related to files-uploads: C:\xampp\tmp\ini

1) File-uploads:

By using this configuration setting, we can allow and stop the file uploads. The default value is "ON". By changing this values as off we can stop the file-uploads.

2) upload_max_filesize:

By using this configuration setting, we can specify the max^m filesize to upload.

3) upload_tmp_dir:

By using this configuration setting, we can specify the temporary locatⁿ for uploaded files.

The default locatⁿ is tmp folder

== x ==

30-AUG-2012

\$_FILES['uploadfilename']['error']:

Super Global Variable

By using this configuration, we can get the

Error msg if any occurred at the time of file upload.

If the ^{Error} no. is zero, file is uploaded successfully. If Error is 1, one, file size is max^m than server configuration settings value if error is 2, file size is max^m than browser configuration settings value. if error is 3, there is some N/W problem at the time of file uploading. if error is 4, user selected the submit button without any file select.

\$_SERVER: (Get Server Configuration setting)

By using this Super Global Variable, we can get the informⁿ about server & browser if you want to get server S/W, server IP address, browser IP address etc, we can go for \$_SERVER.

(\$_SERVER['SERVER_SOFTWARE']):

By using this we can get the S/W informⁿ what is available in Webserver.

\$_SERVER['SERVER_NAME']:

To get the name of Webserver

\$_SERVER['SERVER_ADDR']:

To get the IP address of Webserver.

\$_SERVER['SERVER_PORT']:

To get the port no. of Web server.
(By default port no. is '80')

* `$_SERVER['REMOTE_ADDR']:`
To get the IP address of Browser.

* `$_SERVER['DOCUMENT_ROOT']:`
By using this super Global Variable, we can
get the document root of current applicatⁿ.
(c:/XAMPP/htdocs)

* ~~`$_SERVER['SERVER_ADMIN']:`~~
By using this we can get the Admin
name ~~@@~~@ server_name,

* `$_SERVER['SCRIPT_FILENAME']:`
To get the current Scriptname with
complete script path. (c:/xampp/htdocs/upload.php)

* `$_SERVER['REMOTE_PORT']:`
To get the port no. of Browser By default
(1317)

* `$_SERVER['SERVER_PROTOCOL']:`
To get the protocol informⁿ what we are
using to xiferr the data. (http/1.1)

* `$_SERVER['REQUEST_METHOD']:`
To get the method what we are using to xiferr
data with the Browser and Server. (Post)

* `$_SERVER['QUERY_STRING']:`
To get the query string value

* `$_SERVER['REQUEST_URI']:`
To get the current scriptname with query
string value.

* `$_SERVER['SCRIPT_NAME']` & `$_SERVER['PHP_SELF']:`
By using it's element
we can get the currently executing scriptname.

* `$_SERVER['REQUEST_TIME']:`
By using this Element we can get the ~~direct~~ time
informⁿ of last Request, as timestamp value.

~~`$_SERVER['HTTP_REFERER']`~~

* `$_SERVER['argc']:`
By using this element we can get the total no. of query
string arguments.

Ex: page1.php

```
<?php  
$sno=100;  
echo "Value is ". $sno;  
?>
```

Page 2.php

```
<?php
```

(32)

* Protocols:

A set of instructions to transfer the data between Browser & server.

Protocols are divided into 2 types.

- 1) Statefull protocols and
- 2) Stateless protocols.

1) Stateful Protocols:

These protocols can maintain the state of Application means we can access the all previous Request and Response values from current request.

In Windows Application we are using these protocols.

Ex: TCP/IP, FTP etc.

2) Stateless Protocols:

These protocols can't maintain the state of Application means we cannot get previous request and Response values from current Request.

In Web Application we are using these protocols Bcoz, They don't carry the previous page values that's why the performance is very fast.

Ex: http, https, etc.

The main Drawback of Stateless protocols is maintaining the state of Application to overcome this drawback ^{State Management} Cookie Concept was introduced.

* COOKIES *

(33)

The state management object using to maintain the state of Application. COOKIE stores the data in clients mem. locatⁿ, These data we can access from any webpage within the Application.

COOKIES stores the informⁿ in 2 locatⁿs. Either Hard Disk or RAM Memory locatⁿ of Client System.

COOKIES are divided into 2 Types:

- 1) In Memory cookie.
- 2) Persistence cookie.

1) Inmemory cookie: (Stores informⁿ in RAM)

If we create any cookie without Explicit Expiry tag comes under Inmemory cookie. Inmemory cookie stores the informⁿ in clients RAM mem. locatⁿ and destroys the data when user closed the Browser.

2) Persistence cookie: (Stores informⁿ in Hard Disk)

If we create any cookie with Explicit Expiry tag comes under Persistence cookie. Persistence cookie stores the data in Hard Disk and Deletes the informⁿ when the lifetime cookies completed.

SET-COOKIE:

By using this fⁿ we can create the cookies in php.

\$-COOKIE:

By using this Super Global Variable we can get the value of cookie

31-AUG-2019

Cookies stored in Browser mem. locatⁿ that's why we cannot access the cookies from one browser to another browser.

Cookies are browser dependent.

Ex: Page1.php

```
<?php  
Setcookie('x',100);  
echo "Cookie created";  
?  
<a href="page2.php"> Go </a>
```

Page2.php

```
<?php  
echo "Value is ".$_COOKIE['x'];  
?>
```

\$-COOKIE \Rightarrow To access Cookies

O/P: Cookie is created Go

Click on it, it displays
Value is

* Steps to create persistence cookie:

(34)

- i) Get the current Date and Time Inform^o when user send the request to access the file where we created cookies.
- ii) Add Lifetime to the current date and time to get Expiry time.
- iii) Create cookies with that Expiry tag.

Ex: <?php

```
setcookie ("un", "scott");
SETCOOKIE ("Sno", 1001, time() + 3600);
echo $_COOKIE ['un'];
echo $_COOKIE ['Sno'];           ↗ If we set time like 3600sec means 1 hour  
the o/p will be same upto 1 hour.
?>
```

O/P : Scott 1001

Persistence Cookies will store in Hard Disk in a file of Browser Mem. locat^o?
The name of that file is username@domain_name

We can Delete the cookie from client Syst. by Recreating the cookie with completed Mozilla → Tools → option → privacy → custom settings → show cookies for History

* Disadvantages of Cookies:

Internet Explorer → Tools → Internet → settings → options

- i) Cookies stores the inform^o in client Syst. that's why client can't delete the inform^o OR client can modify cookie data.
- ii) Cookie is storing limited amt of data.
- iii) cookie can store only Text data.

* SESSIONS *

(35)

Sessions can store the data in Webserver. This data we can access from any webpage in the Browser. By using `$_SESSION`, we can create & access the sessions.

* Differences Betⁿ COOKIES And SESSIONS.

→ COOKIES

- i) COOKIES stores the informⁿ in client-system.
- ii) COOKIES stores limited amt of data.
- iii) COOKIES ^{can} stores only Text data.
- iv) COOKIES are unsecured.

SESSIONS

- i) SESSIONS stores the data in server-system.
- ii) SESSIONS stores huge amt of data.
- iii) SESSIONS can store any type of data.
- iv) SESSIONS are Highly secured.

By Default, we cannot access the sessions of one page from another page. If you want to access the sessions we need to initialize the sessions when the Request is started.

By using SESSION.AUTO-START we can initialize sessions. The Default value is Zero. By changing its value as '1', we can start the sessions when the request is started. Otherwise use session_start function in the applicatⁿ from where you want to access SESSIONS.

```
<?php  
session_start();  
$_SESSION['x']=100;  
echo $_SESSION['x'];  
?  
<a href="page2.php">GotoNext</a>
```

```
<?php  
session_start();  
echo $_SESSION['x'];  
?>
```

O/P: 100 GotoNext ←
100

Session_id:

session_id is an unique value generated by the server when user sends the 1st request to the server. It is alphanumeric string.

When user ~~is~~ connected with server, without session_id, server creates new session_id for the users. This session_id ~~is~~ in client syst. as inmemory cookie. The name of cookie is PHPSESSID and value is the SESSION-ID. At the same time, server temporary mem.locatⁿ, a new file will create to store the session data. filename is same as SESSION-ID with prefix word SESSID.

this file creates in Temporary Mem. Locatⁿ of Server.

01-SEPT-2019

(36)

When user sending the 1st request without session_id, New session will create. The same session_id X'fers Betⁿ Browser and Server, which subsequent request and responses. If user closing the website, the Session cookie will destroy. Again ~~will~~ if we connect with server, the Request will go without session_id, then server creates the new session_id for the user again.

* SESSION_id :

By using this fⁿ, we can get the session id what is generated by the server.

```
<?php  
session_start();  
echo session_id();  
?  
<a href="page2.php">GOTONext</a>
```

```
<?php  
session_start();  
echo session_id();  
?>
```

Note: and starting
After closing Browser each time,
we can see diff. o/p. on Browser.

Ex:

```
<?php  
session_start();  
if(isset($_POST['sub']))  
{  
    $pro = $_POST['drpt'];  
    $qtn = $_POST['t1'];  
    $_SESSION[$pro] = $qtn;  
}  
<br>  
<form method="Post" action=" " >  
products:<select name='drpt'>  
<option> Nokia1100 </option>  
<option> Samsung1100 </option>  
</select>  
<br>  
Quantity:<input name='t1'>  
<br>  
<input type="submit" name="sub" value="submit" >  
</form>  
<a href="bill.php"> show Bill </a>
```

```
<?php  
session_start();  
foreach($_SESSION as  
$k=>$v)  
{  
    echo $k. " --- ". $v;  
    echo "<br>";  
}
```

O/p:

Products: Nokia1100

Quantity:

Submit

Show Bill

UNSET SESSION-DESTROY:

By using this fn, we can delete the data of session.

SESSION-DESTROY:

To destroy the sessions.

Ex: <?php
 SESSION_START();
 \$_SESSION['abc'] = "Scott";
 SESSION_UNSET();
 SESSION_DESTROY();
 ?>

Login page

```
<?php
session_start();
if(isset($_POST['sub']))
{
    $uname = $_POST['txtname'];
    $pwd = $_POST['txtpwd'];
    if($uname == "Scott" and $pwd == "Scott123")
    {
        $_SESSION['aut'] = true;
        echo "<script>location='Welcome.php'</script>";
    }
    else
        echo "Invalid";
}
?>

<form method="post" action=" " >
    username: <input type="text" name="txtname">
    <br>
    password: <input type="text" name="txtpwd">
    <br>
    <input type="submit" value="Login" name="sub">
</form>
```

Welcome page

```
<?php
session_start();
if(isset($_SESSION['aut']))
{
    echo "Welcome to user";
}
else
    echo "<script>location='login.php'</script>";
?>

<a href="logout.php"> logout </a>
```

Logout page

```
<?php
session_start();
session_destroy();
echo "Logged out successfully";
?>
```

* Configuration settings to work with sessions.

C:\xampp\php.ini

1) session.auto_start:

By using this configuration setting, we can start the sessions when the request is started.

The default value is zero. By changing this value as "1", we can start the sessions.

2) session.save_handler:

By using this configuration setting, we can specify the handler where you want to save sessions.

3) session.save_path:

By using this, we can specify the path local, where you want to save sessions.

The default local is temporary folder of Xampp folder. C:\xampp\temp

4) session.use_cookies:

By using this configuration setting, we can use cookies or not to store the sessions. The default value is '1'. If we change value to '0', then we can see diff. session at each refresh.

5) session.name:

By using this configuration setting, we can change the session_cookie name.

The Default name is ~~PHPSESSID~~ "PHPSESSID".

6) session.cookie_lifetime:

By using this configuration setting, we can increase or decrease the lifetime of cookie, what we're using to store sessions.

The default lifetime is '0' that's why it is creating inmemory cookie. By increasing this value we can create persistence cookie to store the sessions in client sys.

Ex:

```
<?php
session_start();
if (isset($_SESSION['aut']))
echo "<script>location='Welcome.php'</script>";
?>
<?php
```

```
<?php
session_start();
session_destroy();
Setcookie("PHPSESSID", " ", time() - 1);
echo "Logged out successfully";
?>
```

04-SEPT-2012

session.gc_maxlifetime :

By using this configuration setting, We can increase or decrease the lifetime to collect the unreferenced sessions by Garbage Collector.

The default value is 1440sec (24 min)

\$_ENV:

(Environment Variables) By using this we can get operating system variables.

```
<?php
```

```
print_r($_ENV);
?>
```

OpP : ~~Array~~

getenv():

By using this fⁿ, We can get the values of Environment variables.

```
<?php
```

```
echo getenv("Computer name");
phpinfo(16);
?>
```

OpP : ~~XP~~

~~Environment Variable~~

phpinfo():

By using this fⁿ, We can display all super global variables or Browser.

```
<?php
```

```
echo getenv("COMPUTER NAME");
phpinfo(16);
?>
```

OpP : Display all Environment variables.

ini_get():

By using this fⁿ, We can get a configuration settings value.

ini_set();

To change the configuration settings value.

```
<?php
```

```
echo ini_get("precision"); //16:OpP
ini_set("precision", 14); //14:OpP
echo ini_get("precision");
?>
```

* FILE HANDLING *

(40)

FILE HANDLING is the concept of reading the file contents as well as writing the file contents.

PHP is providing no. of fns, to read and write the file contents.
If we want to Read and Write the contents of file, First we need to open the file with the specific file mode.

Different Types of file modes are available:

1) r(read):

To Read the file contents. If U open the file with read mode, file ptr locates at the begining of file.

2) w(write):

To write the contents in a file. If U open any file, with Write mode it deletes the previous contents of files & locates the file ptr. at starting positt. If the file is not available, It creates a New file.

3) a(append):

To append some text with the existing content of file.

The file ptr. locates at the end of the file.

4) r+(read/write):

To Read and Write the contents of a file. It is same as Read mode. We can also write file contents as same as Read mode.

5) w+(write/Read):

It is same as Write mode, We can also read file contents.

6) a+(append /read):

It is same as Append mode, We can also read file contents.

7) fopen():

By using this fn, we can Open a file with specific file mode.
It Contains 2 Arguments. filename and filemode.

8) fread():

To Read the filecontents, arguments or file ptr. and file size

9) fwrite():

To write the contents in a file, arguments or file ptr. & ~~new~~ new content.

10) filesize():

To get the total no. of Bytes occupied by a file. Argument is filename.

Ex: <?php

```
$fp = fopen("myfile.txt", "r");
$size = filesize("myfile.txt");
$content = fread($fp, $size);
echo $content;
?>
```

Ex: <?php

```
$fp = fopen("myfile.txt", "w");
fwrite($fp, "John123");
?>
```

Ex: <?php

```
$fp = fopen("myfile.txt", "a");
fwrite($fp, "John123");
?>
```

Ex: <?php

```
$fp = fopen("myfile.txt", "w");
fwrite($fp, "smith");
echo
fread($fp, filesize("myfile.txt"));
?>
```

* **rewind()**:

By using this fⁿ, we can locates the file ptr. at the starting locatⁿ.

Ex: <?php

```
$fp = fopen("myfile.txt", "r+");
fwrite($fp, "smith");
rewind($fp);
echo fread($fp, filesize("myfile.txt"));
?>
```

* **fseek()**:

By using this fⁿ, we can locate the file ptr. ~~on~~ ^{on} specified locatⁿ.

Ex: <?php

```
$fp = fopen("myfile.txt", "w+");
fwrite($fp, "Hello Scott");
fseek($fp, 2);
echo fread($fp, filesize("myfile.txt"));
?>
```

* **fgets()**:

By using this fⁿ, we can read a line from a string. If it locates the file ptr. at the begining of Next line.

Ex: <?php

```
$fp = fopen("myfile.txt", "r");
$line = fgets($fp);
echo $line;
$line = fgets($fp);
echo $line;
?>
```

O/p :

Line1
Line2

NotePad

Line1
Line2
Line3
Line4
Line5

* fgetss():

It is same as fgets, But it ignores the html elements.

42

* fgetc():

By using this f^o, we can get a character from a file.

Ex:

```
<?php  
$fp = fopen ("myfile.txt", "r");  
$line = fgetc ($fp);  
echo $line;  
$line = fgetc ($fp);  
echo $line;  
?>
```

O/P : Li

* file_get_contents():

By using this f^o, we can read the contents of a file without file ptr.

05-SEPT-2019

```
<?php  
echo file_get_contents ("myfile.txt");  
?>
```

* file_put_contents():

By using this f^o, we can write new content in a file.

```
<?php  
echo file_put_contents ("myfile.txt", "abcd");  
?>
```

* readfile():

By using this f^o, we can read the contents of a file at the same time we can write those contents in current buffer locat?

```
<?php  
readfile ("myfile.txt");  
?>
```

O/P : abcd

* file:

This f^o reads each & every line of a file & returns every line as array elements.

```
<?php  
$arr = file ("myfile.txt");  
print_r ($arr);  
?>
```

O/P :
Array [0] = Line1
Array [1] = Line2
Array [2] = Line3

* filectime:

By using this f^o we can get the created date & time inform^o of a file. The o/p it returns as timestamp value.

Ex: <?php

```
echo filectime date('d/m/y', filectime ("myfile.txt"));
?>
```

O/P : 04/09/12

filectime:

To get the last access date & time info of a file.

filemtime:

To get the last modified date and time info of a file.

~~unlink~~ unlink:

By using this fⁿ, we can delete a file permanently from HardDisk.

<?php

```
unlink ("index.html");
?>
```

copy ():

To copy a file from one locatⁿ to another locatⁿ.

<?php

```
copy ("index.php", "C:/ind.php");
?>
```

rename ():

<?php

```
rename ("demo.php", "demoxyz.php");
?>
```

★ Directory functions: ★

1> mkdir:

By using this fⁿ, We can create new directory.

<?php

```
mkdir ("c:/abcd");
?>
```

2> rmdir:

To remove existing directory

<?php

```
rmdir ("C:/abcd");
?>
```

3> opendir:

To open a directory.

<?php

```
$dh = opendir ("E:/abcd");
?>
```

readdir:

To read the files of a directory.

closedir:

To close the opened directory.

Ex: <?php

```
$dh = opendir("C:/xampp");
while ($file = readdir($dh))
{
    echo $file;
    echo "<br>";
}
?>
```

scandir:

By using this fⁿ we can scan the all files of a directory, & Returns arr the arr in the form of an Array.

```
<?php
$arr = scandir("C:/xampp");
print_r($arr);
?>
```

getcwd:

This fⁿ used to get the current working directory.

Ex: <?php Op: c:/xampp/htdocs.
echo getcwd();
?>

chdir:

This fⁿ used to change the current working directory locatⁿ.

Ex: <?php
chdir("e:/");
\$fp = fopen("1.jpg", "r");
?>

FAQ: 6: How can we submit the form (44)

Without Submit Button?

→ By using Javascript submit

```
<form method="post" action="page.php">
<select onchange="submit()">
<option> PHP </option>
<option> ASP </option>
</select>
</form>
```

Q: What does a special set of tags <? = and ?> do in PHP?

→ <?
"hi"
?> Op: hi

06 SEPT 2012

Webserver:

Webserver is a s/w used to run web applicatⁿs.

Once, we taken the space in the webserver, we have to upload the project files from our local syst. to External servers.

By using the Control Panel, we can upload the files into external servers.

Dif^f. Types of External Servers are available to upload project. We can release registered the Space to the websites.

www.byethost.com.
www.godaddy.com
www.noads.biz
www.rackspace.com
www.ipage.com

* File Manager:

(45)

By using this option we can upload the files from our local syst. to external servers.

* Database Manager:

By using this option, we can create the DB tables in External Webservers.

* Email Manager:

To create the email-ids with our domain name.

Every External server contains Root Directory, We should upload a project files into ~~external~~ Root Directories.

* Index:

It is a startup of a file, Every server should contains start up filename with index.

www.byethost.com → Free hosting → sign up → Registration form → After activation → online File Manager → n/docs

http://nileshsatpute.byethost15.com.

* Header:

It is the small amt of data packet betⁿ Browser and Server.
Headers are divided into 2 Types.

- 1) Request header and
- 2) Response header.

1) Request Header:

Data transmission Betⁿ Browser & Server is called as Request Header.

2) Response Header:

Data transmission Betⁿ Server & Browser is called as Response Header.

In PHP, we are using header() to declare headers. The available headers are location, content-type, content-length, etc.

* Location:

07. SEPT. 2012

By using this header, We can redirect from one page to another page.

```
<?php Save!
if(isset($_POST['sub'])) {
    if($_POST['sub'] == "page1")
        header("location: page1.php");
    else
        header("location: page2.php");
}
```

```
<form method="post" action=" " >
<input type="submit" name="sub" value="page1">
<input type="submit" name="sub" value="page2">
</form>
```

O/P:
Blank

Save! Headerlocation.php
O/P: Page1 Page2

* Content-Disposition: attachment:

By

using this header we can display
the download dialogue box or Browser.

```
<?php
header("Content-Disposition: attachment;
filename=abc.txt")
?> Note: we see the window of downloading
as an O/p
```

```
<?php
header("Content-Disposition: attachment;
filename=abc.txt");
readfile("myfile.txt");
?>
```

Output Save: select.htm Query String
[Download](frm.php?fname=drupal-6.1.zip)

```
<br>
<a href="frm.php?fname=gel.php">Download2
<br>
<a href="frm.php?fname=myfile.txt">Download3
<br>
```

```
<?php
Form.php
$qs = $_REQUEST['fname'];
size = filesize($qs);
header("Content-length: $size");
header("Content-Disposition: attachment;
filename=$qs");
readfile($qs);
?>
```

* WWW-Authenticate: Basic Realm=<sitename>

→ By using this header, we can display the ~~Authenticat~~ dialogue Box.

Every Browser contains Authenticat Dialogue Box. To activate this dialogue box, we r using this header.

* \$_SERVER['PHP_AUTH_USER']:

By using this Super Global variable we can get the username textbox values.

* \$_SERVER['PHP_AUTH_PW']:

To get the

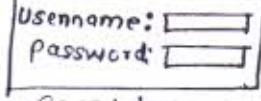
password TextBox value.

Authenticat?

Dialogue box xfers the inform? thr. headers that's why the data xmission performance is very fast.

If we r using Authenticat dialogue box, By default the TextBox values will store in Super Global variables. That's why no need to use session variables to get the control values from one page to another page.

Ex: <?php Save: WWWAuthBasicRealm.php
header("WWW-Authenticate: Basic
Realm=abcd");

echo
\$_SERVER['PHP_AUTH_USER'];
echo
\$_SERVER['PHP_AUTH_PW'];
?> O/P: 
Ex: <?php Save: Authusernamepwd.php

```
$auth=0;
if(isset($_SERVER['PHP_AUTH_USER'])  

and isset($_SERVER['PHP_AUTH_PW']))  

{
$uname=$_SERVER['PHP_AUTH_USER'];
$pwd=$_SERVER['PHP_AUTH_PW'];
// $fp=fopen("logins.txt","r");
while ($line=fgets($fp))
{
list ($un,$pw)=explode(":",$line);
$un=trim($un);
$pw=trim($pw);
if ($uname==$un and $pwd==$pw)
{
$auth=1;
break;
}
}
if($auth==0)
header("WWW-Authenticate: Basic Realm  

=MyWebsite");
```

```

    }  

    header("location:Welcome.php");  

}  

?
}

```

O/P:

47

Authentication Required	
Username :	<input type="text"/>
Password :	<input type="password"/>
<input type="button" value="Login"/> <input type="button" value="Cancel"/>	

```

<?php  

echo "Welcome to ".$_SERVER['PHP_AUTH_USER'];  

?>

```

↳ If we remove dot O/P: Welcome to
(.), Then it shows Error.

* MAIL *

10-SEPT-2012

By using this fn, we can send a mail from our Application to Destination mail id. It contains 4 arguments ~~2~~ address, subject, body, and collectⁿ of headers.

Mail fn Fourth argument is collectⁿ of headers, the separator betⁿ header and another header is "\r\n".

We need to use ~~some~~ SMTP server to send the mails from one locatⁿ to another locatⁿ.

Ex: <?php
if(mail('abc@gmail.com', 'hi', 'hello', 'From:xyz'))
echo "sent";
else
echo "Not sent";
?>

* Different Types of Mail Related Headers:

Diff types of Mail Related headers are available like Cc (Carbon Copy), Bcc (^{Blank} Carbon Copy), Content-type etc.

```

<?php  

if(mail("abc@gmail.com", "hi", "hello", "From: xyz\r\nCc: scott@gmail.com"))  

echo "sent";  

else  

echo "Not sent";  

?>

```

48

Ex:8

```
<?php  
$str=<<<abc  
<form method="post" action="http://swamyssoft  
solutions.com/geto.php">  
<span style='color: Red'> Username</span>  
<input type='text' name='t1'>  
<br>  
<input type="submit" value="click" name=  
"sub">  
  
abc;  
if (mail("abc@gmail.com", "HTML email",  
$str, "From:smith@gmail.com\r\nX-Mailer:  
type:text/html"))  
{  
echo "Mail sent";  
}  
else  
echo "Not sent";  
  
>  
form method="post" action=" ">  
to:<input name='to'>  
<br>  
sub:<input name='sub'>  
<br>  
body:<input name='body'>  
<br>  
from:<input name='from'>  
<br>  
input type="submit" value="click" name="sub1">  
</form>
```

* String fns: *

String is a collectⁿ of characters, no. of string fns are available to work with string.

1) strlen: By using this fn we can get the length of all characters of a string.

Ex: <?php

```
$str = "Welcome";
echo strlen($str);
?>
```

O/P: 7

4) chr:

By using this fn we can get the character of an ASCII value.

Ex: <?php
echo chr(97); O/P: a
?>

5) ord:

To get the ASCII value of \$/p character.

Ex: <?php
echo ord('A');
?>
O/P: 65

8) n2br:

By using this fn we can break new lines of a string.

```
<?php
$str = "Welcome
to
PHP";
echo n2br($str);
?>
```

O/P: Welcome
to
PHP.

2) strtoupper:

By using this fn, we can convert the all characters of a string into uppercase characters.

Ex: <?php

```
$str = "Welcome";
echo strtoupper($str);
?>
```

O/P: WELCOME

3) strtolower:

By using this functⁿ, we can convert the all characters of a string into lowercase characters.

<?php

```
$str = "WeLCOME";
echo strtolower($str);
?>
```

O/P: welcome.

6) ucfirst(): By using

this fn we can convert the 1st character of a string into uppercase character.

Ex: <?php

```
$str = "welcome";
echo ucfirst($str);
?>
```

O/P: Welcome.

7) ucwords:

Converts ~~all~~ First character of all words into uppercase.

<?php

```
$str = "Welcome To Scott";
echo ucwords($str);
?>
```

O/P: Welcome To Scott

9) addslashes:

By using this fn, we can add backslashes within the string where single quotations & double quotations are occurred.

Ex: <?php

```
$str = "Welcome";
$str1 = addslashes($str);
echo $str1;
?>
```

O/P: Wel\|come

10) stripslashes:

By using this fn, we can remove the slashes what we added with add slashes.

Ex: <?php

```
$str = "Wel\|come";
$str1 = stripslashes($str);
echo $str1;
echo stripslashes($str1);
?>
```

O/P:

11) addcslashes:

Adds the backslashes in front of Specified character.

12) stripcslashes:

To strip the slashes what we added with addcslashes

Ex: <?php

```
$str = "Wel'Come";
$str1 = addcslashes($str, "e");
echo $str1;
echo "<br>";
echo stripcslashes($str1);
?>
```

O/P :

13) str_word_count:

By using this fⁿ, we can get the total no. of words of a string. We need to pass mode value as second argument. If mode is zero (0), it returns total no. of words.

If mode is one (1), it returns each word as an array. Array values of words and keys are 0,1,2,...

If mode is 2, it returns the all words as array, values are the words of string & keys are the index no. of the string.

Ex: <?php

```
$str = "Welcome to scott";
print_r(str_word_count($str, 2));
```

?>

O/P:

14) similar_text:

By using this fⁿ, we can get the similarities Betw 2 ~~2~~ string. (50)

<?php

```
$str = "Welcome to scott";
$str1 = "Welcome to scott";
echo similar_text($str, $str1);
```

?>

O/P : 14

→ IF we write like that
similar_text(\$str, \$str1); then o/p will be 12.

15) join:

By using this fⁿ, we can join Array elements as a string. It is same as implode.

<?php
\$arr = array("scott", "smith");
echo join("/", \$arr);
?>

O/P :

16) trim:

Removes the LHS & RHS spaces of a string.

17) ltrim:

Removes the LHS spaces of a string.

18) rtrim:

Removes the RHS spaces of a string.

<?php

```
$str = "Welcome ";
$str1 = " scott";
echo trim($str), ltrim($str1);
?>
```

O/P :

19) chop:

It is an alias of rtrim.

20) str_shuffle:

This fⁿ randomly shuffles a string.

<?php

```
$str = "Welcome";
echo str_shuffle($str);
?>
```

(50)

21) str_replace:

Replaces a part of String with a New string.

22) str_ireplace:

It is same as str_replace, But it is case Insensitive.

<?php

```
$str = "Welcome";
echo str_ireplace("Come", "go", $str);
?>
```

23) str_repeat:

Repeats a string, with a Specified no. of Times.

<?php

```
$str = "Welcome";
echo str_repeat($str, 3);
```

O/P: WelcomeWelcomeWelcome

3 Times

24) split:

Converts a string as an Array.

<?php

```
$str = "Welcome to Scott";
$arr = str_split($str, 3);
print_r($arr);
?>
```

25) strcmp:

This fn compares 2 strings, and Returns '0', If Both are same,

If Returns > 0, If 1st string > 2nd string.

If Returns < 0, If 1st string < 2nd string.

Ex:
1) <?php

```
$str = "Smith";
$str1 = "smith";
echo strcmp($str, $str1);
```

O/P: 0

Ex: 2 => <?php

```
$str = "Smith";
$str1 = "smith";
echo strcmp($str, $str1);
```

O/P: -1

26) strcasecmp:

(5)

It is same as strcmp.

But it is case Insensitive.

* * * * *

27) strchr():

By using this fn, we can get all characters of a string from specified character.

<?php

```
$str = "SmithHello";
echo strchr($str, "l");
?>
```

28) strrchr:

It is same as strchr,

But it gets the string from reverse dir.

Ex: <?php

```
$str = "SmithHello";
echo strrchr($str, "l");
?>
```

29) strrstr():

same as strchr.

30) stripos:

same as strpos, But

case Insensitive.

31) strpos:

Find the positⁿ of a character in a string.

32) stripos:

same as strpos, But case Insensitive

Ex: <?php

```
$str = "Welcome";
echo strpos($str, "e");
?>
```

O/P: 1

33) substr:

To get the substring of a string.

<?php

```
$str = "Welcome";
echo substr($str, 3, 4);
?>
```

O/P: e

34) `strip_tags`: By using this fn, we can strip the html tags.

```
<?php  
$str = "Welcome <input type='button'  
       value='Click'>";  
echo strip_tags($str);  
?>
```

35) `strrev`: By using this fn, we can convert the string into Reverse dirn.

```
<?php  
$str = "hello";  
echo strrev($str);  
?>
```

Encryption: Date : 12. SEPT. 2013

It is a concept of Encoding & Decoding the data. Basically, we have 2 Types of Encryption in PHP. Those are

- 1) One-way Encryption
- 2) Two-way Encryption

II) One-way Encryption:

By using

this, we can encode the data But, we can not decode encoded data.

1) `md5` : (Msg. Digest 5)

By using this fn, we can encode the data as 32 characters length, alphanumeric string.

(combination of Alphabets & Nos.)

Ex: <?php

```
$str = "Scott";  
echo md5($str);  
?>
```

O/P

2) `crc32` (Cyclic Redundancy Check) (52)

This fn converts \$/p string as Numeric encrypted data.

Ex: <?php

```
$str = "Scott";  
echo crc32($str);  
?>
```

3) `SHA1`:

converts \$/p string as 40 characters length alphanumeric string.

Ex: <?php

```
$str = "Scott";  
echo sha1($str);  
?>
```

4) `crypt`:

converts \$/p string as alphanumeric string with special characters

Ex: <?php

```
$str = "Scott";  
echo crypt($str);  
?>
```

II) Two-way Encryption:

By using this concept, we can Encode & Decode the data.

1) `base64_encode`:

By using this fn, we can encode a string as 64-bit encrypted data.

Ex: <?php

```
$str = "Welcome";
```

2) `base64_decode`:

To decode the encoded data.

Ex: <?php

```
$str = "Welcome";
$str1 = base64_encode($str);
echo $str1;
echo base64_decode($str1);
?>
```

Ex: <?php

```
echo "hi";
echo "hello";
sleep(10);
echo ("scott");
?>
```

O/P: hihello scott

Note:

scott display after 10 sec.

* characters what we can used

With date function are as follows: →

i) d: 2 digit day with leading 0.

ii) j: 2 digit day without 0.

iii) N: day no. from 0 to 6.

iv) Y: ISO representation of day number (1-7)

1) L: Full day name

2) D: Three letters day name.

3) %E: Day no. in the year

4) W: Week no. in the year.

5) m: month no. with 0.

6) n: month no. without 0.

7) F: full month name.

8) M: Three letters month name.

9) t: No. of days in the given month.

10) L: checks the given yr is Leap or not.

11) y: 2 digit year no.

12) Y: 4 digit year no.

13) a: Lower case antimeridiem/post
(am) meridiem
(pm)

14) A: Upper case antimeridiem/post
Meridiem.

15) j: 12 hour format without leading 0.

16) G: 24 hour format without leading 0.

17) h: 12 hour format with zero.

18) H: 24 hour format with zero.

19) s: seconds.

20) i: minutes.

Ex: <?php
echo date("i");
?>

* Date And Time Functions:

1) time:

By using this fⁿ, We can get the current date and time information as timestamp value.

A timestamp value is nothing but a total no. of seconds from ~~1970~~ 1970 Jan 1st, 12.00am to upto now.

Ex: <?php

```
echo time();
?>
```

2) date:

Converts the f/p timestamp value as date, month and year format.

Value as date, month and year format string.

Ex: <?php

```
$t = time();
echo date('d/m/y', $t);
?>
```

3) getdate:

By using this fⁿ, we can get the current date and time information as an Array.

Ex: <?php

```
print_r(getdate());
?>
```

4) sleep:

By using this fⁿ, we can stop the execution of script upto specified time intervals.

checkdate:

By using this fⁿ, we can
check the fⁿip date is existed or not.

Ex: <?php

```
echo checkdate("2", "29", "2012");  
?>
```

Jacoby

(56) *Trichocentrus*

- * JQuery is an Open Source JavaScript library providing no. of predefined functionalities.
- * JQuery is available with different extensions like JQuery.js, JQuery-1.js, etc.
- * JQuery functionalities we can apply on any type of element.
- * JQuery supports any type of Web browser.
- * JQuery methods we can apply on html elements using '\$' 21. SEPT. 2012

```

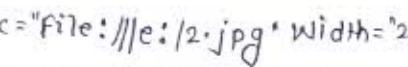
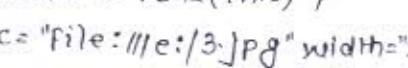
<Script src="jquery-1.js">
</script>
<style>
p{
    background-color: green;
    color: white;
}
</style>
<script>
function funshow()
{
    $("#P2").show("slow")
}
function funhide()
{
    $("#P2").hide ("slow")
}
<script>
<p id="P1">
    some content . . .
</p>
<p id="P2">
    some content . . .
</p>
<input type="button" value="show"
       onclick="funshow()">
<input type="button" value="hide"
       onclick="funhide()">

```

- * show(): Display elements in Horizontal dir?
- * Hide(): Hides the elements in Horizontal dir
- * slideUp(): Hides the elements in vertical dir?
- * slideDown(): Displays the elements in vertical dir?

SAVE AS: ~~FunShowHide.htm~~ FunShowHide.htm
OR:

Ex:

```
<script>src = "jquery-1.js">
</script>
<script>
function fun1(t)
{
$(t).fadeOut("slow")
$(t).fadeIn ("slow")
}
</script>

onmouseover = "fun1(this)">

onmouseover = "fun1 (this)">
```

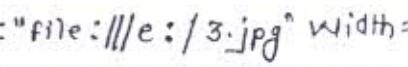
SAVE AS: ~~MyFile.htm~~
FadeInFadeOut.htm

(58)

* fadeTo:

To fade the element ^{upto} the specify value.

SAVE AS: ~~MyFile.htm~~
fadeTo.htm

```
<script>src = "jquery-1.js">
</script>
<script>
function fun2()
{
$(".div1").fadeTo("slow", 0.2)
}
</script>
<style>
.div1{
background-color: black;
width: 200;
height: 200;
position: absolute;
top: 0;
left: 0;
}
</style>

<div class = "div1" onmouseover="fun2()">
</div>
```

* corner()

(59)

By using this fn of JQuery, we can display contents with round corners.
It is available with "jquery.corner.js" library. The supporting library is jquery.js,
jquery-1.js etc.

```
<script> $rc = "jquery-1.js"
</script>
<script src="jquery.corner.js">
</script>
<script>
function fun1()
{
    $("*").corner();
}
</script>

<br>
<div id="div1" style="width:100; height: 100; background-color: blue"></div>
<button onclick="fun1()"> click </button>
```

* addClass():

By using this method, we can add the class selector properties on the elements.

* removeClass():

To Remove the class selector properties of an Elements.

```
<script> $rc = "jquery-1.js"
</script>
<style>
.c1s1 {
    background-color: yellow; color: red;
}
</style>
<script>
function fun2()
{
    $("#P1").removeClass("c1s1");
}
function fun1()
{
    $("#P1").addClass("c1s1");
}
</script>
<p id="P1" onmouseover="fun1()" onmouseout="fun2()>
Content . . .
</p>
```

A Cycle :

By using this f?, we can apply sliding effect on image correctly.
This f? is available in jquery.cycle.js library.com, jquery.cycle.min.js etc.

fx:

By using this property we can specify the type of sliding effect what u want to apply on image correctly.

(zoom, tools, wipe, curtain, curtainx, shuffle) etc f? s

Ex: <script src='jquery-1.js'>

</script>

<script src='jquery.cycle.all.js'>

</script>

<script>

function func()

{
\$("#div1").cycle({fx:'zoom'})}

}

</script>

<style>

#div1{

position: absolute;

top: 100;

left: 100;

}

</style>

<div id="div1">

</div>

<input type="button" value="click" onclick="func()">

i) blindx

ii) blindy

iii) blindz

iv) cover

v) curtain

vi) curtainx

vii) fade

viii) fadeZoom

ix) growx

x) growy

xi) none

xii) scrollUp

xiii) scrollDown

xiv) scrollLeft

xv) scrollRight

xvi) scrollHorz

xvii) scrollVert

xviii) shuffle

xix) slideX

xx) slideY

xxi) toss

xxii) turnUp

xxiii) turnDown

xxiv) turnLeft

xxv) turnRight

xxvi) UnCover

xxvii) wipe

xxviii) zoom

Validate.js :

By using this fn, we can validate the form controls. This fn is available validatejquery.validate.js library. This library is providing no. of properties to apply the validation those r required, email, number, etc.

By using errorClass property, we can apply the class selector on Validate library. When we r applying validation on the elements, we need to provide name property to the elements.

```

<script src = "jquery-1.js">
</script>
<script src = "jquery.validate.js">
</script>
<style>
.cls1{
    color: blue;
    border-color: red;
}
</style>
<script>
    function fun1()
    {
        $("#frm1").validate({errorClass : 'cls1'})
    }
</script>
<body onload = "fun1()">
<form id = "frm1" action = "page1.html">
    Username: <input type = "text" name = "txtnuser" class = "required">
    <br>
    Password: <input type = "password" name = "pwd" class = "required"
                minlength = 10 maxlength = 20>
    <br>
    Email: <input type = "text" class = "required email" name = "Em">
    <br>
    Age: <input class = "required number" min = 20 max = 99 name = "ag" id = "f9" >
    <br>
    URL: <input type = "text" name = "Ur" class = "Url" >
    <br>
    <input type = "submit" value = "click" name = "Sub" >
</form>
```

* Cookie:

By using this f(), we can create the cookies in a client-syst.
It is available in cookie.js library. To create the persistence cookie we can use ~~a~~ expires properties.

If u want to destroy the cookie before lifetime, Recreate that cookie with null value.

```
<script src="jquery-1.js"></script>
<script src="cookie.js"></script>
<script>
function fun1()
{
    $.cookie("un", "scott")
    $.cookie("city", "hyd", {expires:1})
    alert($.cookie("un"))
    alert($.cookie("city"))
}
</script>
<input type="button" value="create" onclick="fun1()">
```

* Strong or not:

```
<script>
function fun1(val)
{
    x=false;
    for (i=0; i<val.length; i++)
    {
        ch=val.charAt(i);
        if (ch=="@" || ch=="#")
        {
            x=true;
            break;
        }
    }
    if (x==false)
        alert("Not strong");
    else
        alert("strong")
}
</script>
<input onblur="fun1(this.value)">
```

* Webcam.js : (library)

By using this library we can display the video capture control on Browser. we can activate camera control using this library. It is providing no. of methods.

i) Webcam.get_html:

By using this method, we can get HTML WebCam control on Browser. It contains 4 Arguments width & height of video Recorder control and width & Height of Image control.

ii) Webcam.snap:

By using this, We can take a snap from Webcam control.

iii) Webcam.reset:

To Reset the Webcam control.

iv) Webcam.configure:

To Configure the cam control.

v) Webcam.set_api_url:

This fn is used to execute php script when user take a snap from a Webcam control.

vi) Webcam.set_quality:

(1-100) By using this fn, We can change the quality of jpeg image.

vii) Webcam.set_shutter_sound:

By using this fn, We can enable and disable sound

Ex:

```
<script src="Webcam.js"></script>
<script>
document.write(Webcam.get_html(400,400,500,500));
Webcam.set_api_url("store.php");
Webcam.set_quality(100);
Webcam.set_shutter_sound(true);
</script>
<body>
<br> <input type="button" value="Snap" onclick="Webcam.snap()">
<input type="button" value="Reset" onclick="Webcam.reset()">
<input type="button" value="Configure" onclick="Webcam.configure()">
</body>
```

```

<?php
$filename = time(). ".imagez.jpg";
$content = file_get_contents("php://input");
file_put_contents($filename, $content);
?>

```

(64)

* date picker:

* By using this fn, we can display the calendar control on browser.
 It is available in datepicker.js library. It requires some supporting libraries those are jquery.ui.core.js, jquery-1.6.2.js.

* It requires a supporting css library i.e. jquery.ui.all.css

* This library is providing no. of methods ~~etc dateFormat~~

dateFormat():

By using this method, we can change the format of Date Time.

minDate():

To specify minimum Date.

maxDate():

To specify the Max^m Date.

onSelect():

To execute a javascript, when a date is changed.

Ex:

26. SEPT. 2012

```

<head>
<link rel="stylesheet" href="jquery.ui.all.css">
<script src="jquery-1.6.2.js"></script>
<script src="jquery.ui.core.js"></script>
<script src="jquery-ui-datepicker.js"></script>
<script>
function fun1()
{
    $("#txt1").datepicker({dateFormat:"yy/mm/dd", minDate:0, maxDate:03,
        onSelect: function(data)
        {
            alert(data)
        }
    })
}
</script>
<head>
<body onload="fun1()">
From Date: <input type="text" id="txt1">
</body>

```

Ex:

```

<head>
  <link rel="stylesheet" href="jquery.ui.all.css">
  <script src="jquery-1.6.2.js"></script>
  <script src="jquery.ui.core.js"></script>
  <script src="jquery.ui.datepicker.js"></script>
  <script>
    function fun1()
    {
      $("#txt1").datepicker({dateFormat:"yy/mm/dd", minDate=0, maxDate=3,
        onSelect: function(data){$("#txt2").datepicker({dateFormat:"yy/mm/dd"})}
      })
    }
  </script>
</head>
<body onload="fun1()"> FromDate:<input type="text" id="txt1" /> <br>
  ToDate:<input type="text" id="txt2" />
</body>

```

* draggable:

By using this library we can drag the elements on webpage.
The library name is jquery.ui.draggable.js

The supporting libraries are jquery-1.6.2.js, jquery.ui.core.js,
jquery.ui.widget.js, jquery.ui.mouse.js

```

<script src="jquery-1.6.2.js"></script>
<script src="jquery.ui.core.js"></script>
<script src="jquery.ui.widget.js"></script>
<script src="jquery.ui.mouse.js"></script>
<script src="jquery.ui.draggable.js"></script>
<script>
  $(document).ready(function(){
    $("#img1").draggable()
    $("#div1").draggable()
  })
</script>
</head>
<body>
  
  <div id="div1" style="border:1px solid silver; width:100; height:200; background-color: #green">

```

Hello

Hi
</div>
</body>

(66)

Ex:

```
<script src="jquery-1.6.2.js"></script>
<script src="jquery.ui.core.js"></script>
<script src="jquery.ui.widget.js"></script>
<script src="jquery.ui.mouse.js"></script>
<script src="jquery.ui.draggable.js"></script>
<script>
    sec=60;
    min=2;
    var t;
    function start()
    {
        t=setInterval("ctime()",50);
    }
    function ctime()
    {
        sec--;
        if(min==0 && sec==0)
        {
            document.getElementById('div1').innerHTML=min+":"+sec
            alert("Game over")
            $("img").hide("slow")
            clearInterval(t)
        }
        if(sec<0)
        {
            sec=59
            min--;
        }
        document.getElementById('div1').innerHTML=min+":"+sec
    }
    function fun1()
    {
        $("img").draggable()
        start();
    }
</script>
```

```

<body>
  <div id="div1" style="color:blue; font-size:40">
    3:00
  </div>
  
  
  
  
  
  <input type="button" value="play" onclick="fun1()">
</body>

```

27. SEPT. 2012

toggle():

By using ~~effect~~ f(), we can execute 2 f()'s parallelly, one after another. It is available in jquery.effects.core.js library. The supporting library is jquery-1.6.2.js library.

```

<script src="jquery-1.6.2.js"></script>
<script src="jquery.effects.core.js"></script>
<script>
  $(function()
  {
    $('#button').toggle(function()
    {
      $('#img1').hide("slow") $('#button').val("show")
    })
    function()
    {
      $('#img1').show("slow")
      $('#button').val("Hide")
    });
  });
</script>
<body>
  <input type='button' value='Hide' id="button">
  <br>
  
</body>

```

jquery effects library is providing diff. types of Effects, like pulse, slide, etc. we can apply this effects on Elements ~~to hide them~~. To hide them. The available effects are blind, bounce, fold etc. For Every type of Effect ~~there~~ 1 library is available. We need to include the library in our webpage.

The supporting library is jquery-1.6.2.js

Ex:

```

<Script src = "jquery - 1.6.2.js" ></Script>
<Script src = "jquery.effects.core.js" ></Script>
<Script src = "jquery.effects.blind.js" ></Script>
<Script src = "jquery.effects.bounce.js" ></Script>
<Script src = "jquery.effects.clip.js" ></Script>
<Script src = "jquery.effects.drop.js" ></Script>
<Script src = "jquery.effects.explode.js" ></Script>
<Script src = "jquery.effects.fold.js" ></Script>
<Script src = "jquery.effects.highlight.js" ></Script>
<Script src = "jquery.effects.pulse.js" ></Script>
<Script src = "jquery.effects.scale.js" ></Script>
<Script src = "jquery.effects.shake.js" ></Script>
<Script src = "jquery.effects.slide.js" ></Script>

<Script>
    Function fun1()
    {
        var = document.getElementById('drop1').value $( "#img1" ).hide( val )
    }
</Script>
<Head>
<Select id = "drop1" >
    <Option value = "blind" > blind </Option>
    <Option value = "bounce" > bounce </Option>
    <Option value = "clip" > clip </Option>
    <Option value = "drop" > drop </Option>
    <Option value = "explode" > explode </Option>
    <Option value = "fold" > fold </Option>
    <Option value = "highlight" > highlight </Option>
    <Option value = "pulse" > pulse </Option>
    <Option value = "scale" > scale </Option>
    <Option value = "shake" > shake </Option>
    <Option value = "slide" > slide </Option>
</Select>
```

```
<input type="button" value="Click" onClick="fun1()"><br/>

```

* \$.get():

By using this fⁿ, we can execute a Webpage ^{from server} without submitting a current page. It contains 3 Arguments.

1) Server side script name

2) Query String collectⁿ

3) Function Defⁿ what u want to execute when the server side script executⁿ is completed.

It is using get() method to x'ferr the data.

~~* \$.post()~~:

It is same as \$.get, But it is using post method to x'ferr the data Betⁿ Browser and Server.

```
<script src="jquery-1.3.js"></script>
```

```
<script>
```

```
function fun1()
```

```
{
  $.get("page2.php", {qs:'scott', qsl:'smith'}, function(x){alert(x)})} </script>
```

```
<input type="button" value="Click" onClick="fun1()"><br/>
```

```

```

```
<?php save:page2.php
```

```
$un = $_REQUEST['qs'];
```

```
$sq = $_REQUEST['qsl'];
```

```
echo $un, $sq;
```

```
?
```

Ex:

28. SEPT. 2012

```
<script src="jquery-1.3.js"></script>
```

```
<script>
```

```
function fun1(pname)
```

```
{
  $.get(pname, {}, function(data){document.getElementById('div1').innerHTML = data});}
```

```
<table border='1' width="100%" height="100%">
```

```
<tr><td style='width:100' onmouseover="fun1('page1.html')">Page1.html <td rowspan="3"
```

```
valign="top"><div id="div1"> Hi </div>
```

```
<tr><td onmouseover="fun1('page2.html')"> Page2.html
```

```
<tr><td onmouseover="fun1('page3.html')"> Page3.html
```

```
</table>
```

```
<h1 style='color:green'> Page1 </h1>
<img src='file:///e:/5.jpg' width='200'>
```

~~for style~~

```
<h1 style='color:red'> Page2 </h1>
```

<img src=''

```
<h1 style='color:yellow'> Page3 </h1>
```

<img src=''

```
<h1 style='color:pink'> Page4 </h1>
```

<img src=''

```
<h1 style='color:purple'> Page5 </h1>
```

<img src=''

★ jquery is providing jquery.ui.tabs.js library used to display tab control on the webpage, requires some supporting libraries those are jquery.ui.core.js, jquery.ui.widget.js and jquery-1.6.2.js, it requires the css file i.e. jquery.ui.all.css

★ To display the Tab control, first we need to take a container like div, span, etc. Tabs we can display by using list items.

```
<link rel="stylesheet" href="jquery.ui.all.css">
<script src="jquery-1.6.2.js"></script>
<script src="jquery.ui.core.js"></script>
<script src="jquery.ui.widget.js"></script>
<script src="jquery.ui.tabs.js"></script>
<script>
$(function(){$("#tabs").tabs();});
</script>
<body>
<div id="tabs">
<ul>
<li><a href="#tabs-1">Content1</a></li>
<li><a href="#tabs-2">Content2</a></li>
<li><a href="#abc">MyPage</a>
</li>
</ul>
```

71

```
<div id="tabs-1">
<h2> Content Heading 1 </h2>
<p> This is a Paragraph </p>
</div>
<div id="tabs-2">
<h2> Content Heading 2 </h2>
<p> This is a Paragraph </p>
</div>
<div id="tabs-3">
<h2> Content Heading 3 </h2>
<p> This is a Paragraph </p>
</div>
```

* size :

By using this fn we can get the total no. of elements of a type.

```
<script src="jquery-1.js"></script>
<script>
    function func()
    {
        alert($("#div").size())
    }
</script>
<div> Hi </div>
<div> Nilesh </div>
<input type="button" value="click" onclick="func()"/>
```

* html () :

By using this fn, We can display some html content on ~~inner~~ container.

Ex:

```
<script src="jquery-1.js"></script>
<script>
    function func()
    {
        $("#div1").html("<img src='file:///e:/3.jpg' width='100'>")
    }
</script>
<div id="div1"> Hi </div>
<input type="button" value="click" onclick="func()"/>
```