

## \* PHP strings & strings function

### 1. Add slashes

1. Addslashes()

2. Count\_chars()

3. Echo echo()

4. Trim ltrim rtrim()

5. Point()

6. Rtrim()

7. Str\_replace()

8. Str\_split()

9. Str\_word\_count()

10. Strcmp strcmp()

11. Strlen()

12. Strpos()

13. Strtolower()

14. Strtoupper()

15. Substr substr()

16. trim()

i



<?php

8

```
$con=mysqli_connect ("localhost", "root", "123", "student");
$name=$_POST ["txt Name"];
$contact=$_POST ["txtContact"];
$Email=$_POST ["txtEmail"];
$Gender=$_POST ["txtgender"];
$Aadhar Card no=$_POST ["txtaadhar card no"];
$Address=$_POST ["txtaddress"];
$query="insert into values ("
```

```
if (mysqli_query ($con, $query)) {
```

```
echo "Row inserted successfully";
```

3

else

```
echo "Error". mysqli_error ($con);
```

3

?

ale">  
ale"><br>  
>Card no"><br>  
>txt", name=

mysql\_connect (Host, username, password, DB)

Host - specifies server to connect

Username - mysql username

password - mysql password to get access

DB - database name to which we need to connect

<?php

```
$con = mysql_connect (Host, username, password, DB)  
if (!$con){
```

```
echo "Connection to Server Error".mysql_error($con);
```

}

?>

The default username for mysql database server is root and there is no password. However, to prevent your database from intrusion and unauthorised access use should set password from mysql accounts.

### \* Closing the mysql database server

- i. The connection to the mysql database server will be closed automatically as soon as the execution of the script end. However, if you want to close it earlier you can call mysqli\_close function.

e.g. <?php

```
$con = mysql_connect (Host, username, password, DB)  
if (!$con){
```

```
echo "Connection to Server Error".mysql_error($con);
```

}

```
mysql_close($con);
```

?>

## \* Super Global Arrays

Several predefined variables in php are "Super Globals" which means that they are always accessible regardless of scope & it can be accessed from any function, class or file without having to do anything special the php super global variables are: ~~\$\_GLOBALS~~

2. ~~\$\_SERVER~~
3. ~~\$\_REQUEST~~
4. ~~\$\_POST~~
5. ~~\$\_GET~~
6. ~~\$\_FILES~~
7. ~~\$\_COOKIE~~
8. ~~\$\_SESSION~~
9. ~~\$\_SESSION~~

### 1. ~~\$\_GLOBALS~~

→ ~~\$\_GLOBALS~~ is a php super global array which is used to access global variables from anywhere in the php script (also from within functions or methods)

→ Php stores all global variable in an array called ~~\$\_GLOBALS~~ [index]: the index holds the name of the variable.

e.g. <?php

~~\$x = 25;~~

~~\$y = 25;~~

~~function addition()~~ {

~~\$GLOBALS['z'] = \$GLOBALS['x'] + \$GLOBALS['y'];~~

~~}~~

~~addition();~~

~~echo \$z;~~

?>

Output: 100

In the above example since `Z` is a variable present within the `$GLOBALS` array it is also accessible from outside the function.

## 2. `$_SERVER`

- i. `$_SERVER` is a super global array which holds information about headers, path and script location.

`<?php`

```

echo $_SERVER[['PHP_SELF']];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER['HTTP_REFERER'];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?> 3

```

Write a program

A php session basis  
What is session?  
1. Although you can't security issue computers it can modify a data in your  
2. Also every server all automatically  
It means system to upload a page

`$_SERVER` [ ]  
Returns the

`$_SERVER` [ ]  
returns

`$_SERVER` [ ]  
returns

`$_SERVER` [ ]  
returns

`$_SERVER` [ ]  
returns

A php session is used to store on server on temporary basis

What is session?

1. Although you can store data using cookie but it has security issues since cookies are stored on user computers it is possible for an attacker to easily modify a cookie content to insert potential harmful data in your application.
2. Also everytime the browser requests a url to the server all the cookie data for a website is automatically sent to the server within the request. It means you have to store 5 cookies on user system each having 4kb in size the browser need to upload 20kb of data each time the user use a page which can affects your site performance.

**`$_SERVER['PHP_SELF']`**

Returns the filename of the currently executing script.

**`$_SERVER['ADDR']:`**

Returns the IP of the host server.

**`$_SERVER['SERVER_NAME']`**

Returns the name of the host server.

**`$_SERVER['SERVER_SOFTWARE']:`**

Returns the server identification string.

**`$_SERVER['SERVER_PROTOCOL']`**

Returns the name of the information protocol.

**`$_SERVER[REQUEST_METHOD]`**

returns the request method used to

**`$_SERVER[REQUEST_TIME]`**

request the time stamp.

**`$_SERVER[QUERYSTRING]`**

returns the query string if the page is accessed by a query string

**`$_SERVER[HTTP_SERVER]`**

returns the accept header from the

**`$_SERVER[HTTP_ACCEPT_CHARSET]`**

returns the accept charset header from the current request.

**`$_SERVER[HTTP_HOST]`**

returns the host header from the current request.

**`$_SERVER[HTTP_REFERER]`**

returns the complete URL of the current page.

**`$_SERVER[`**

is the script required through a secure http protocol.

**`$_SERVER[Remote_addr]`**

returns the IP address from where the viewing the current page.

**`$_SERVER[Script_filename]`**

returns the absolute path of the currently executing script.

## Exception handling.

- i When creating scripts and web applications error handling is an important part.
- ii Error handling is a process of catching errors raised by a program & and then taking appropriate actions.
- iii There are number of reasons that may cause errors.
  - 1. The webserver may run out of disk.
  - 2. The user might have entered invalid form data.
  - 3. The file or database record that you were trying to access may not ~~access~~ exists.
  - 4. The application might not have permission to write to a file on the disk.
  - 5. A service that a application needs to access might be temporarily unavailable.
  - 6. This type of errors are unknown as runtime errors because they occur at the time when the script runs.
  - 7. They are distinct from Syntax errors that needs to be fixed before the script will run.
- iv There are two ways to handle exception.
  - i. Using ~~die function~~
  - ii. Using custom Error handlers.

### 1. Basic error handling using die():

e.g. <?php

```
$file=fopen("abc.txt","r");
```

?>

If the file does not exists you might get an error as below:

Warning: fopen("abc.txt") [function.fopen]: failed to open stream: No such file or directory in "C:\wamp\www\web\test.php" on line 2.

If we follow some simple steps to prevent the user from getting an error message in the above example we test whether the file exists before we try to access it.

```
<?php
if (file_exists("abc.txt")){
    $file=fopen ("abc.txt", "r");
}
else {
    die ("Error: the file you are trying to access does not exist.");
}
```

output:

Error: the file you are trying to access does not exist.

description:

By implementing a simple condition to check whether the file exists or not before trying to access it we can generate an error message that is more meaningful to user.

Creating a

1. You can use error handling to deal with errors.
  2. The custom error handling is better than the default error handling as it might display more meaningful error messages in a file to help the user attempt to ignore the error.
  3. The custom error handling handles all errors and displays a meaningful error message.
- Syntax: error\_reporting();

Parameters

+ error\_reporting();

errors

error

error

1. You can create your own ~~be~~ error handler function to deal with the runtime errors generated by PHP engine.
2. The custom error handler provides you greater flexibility and better control over the errors, it can inspect the errors and decide what to do with the errors, it might display message to the user, log the error in a file to the database or send an email, attempt to fix problem, exit the execution of script, ignore the errors.
3. The custom error handler function must be able to handle at least two parameters (errno and error message) However it can optionally accept 3 additional parameters (error\_file, error\_line, error\_context) syntax:

error-function (errno, error, error\_file, error\_line, error\_context)

#### Parameters

- errno - error number - specify error report level for the user defined errors, must be a value number.  
(e.g. E\_ERROR, E\_WARNING, E\_ALL)

error - error string - specifies error message for user defined errors

error\_file - specifies the file name in which the error occurs

error\_line - specifies the line number in which the error occurs.

error\_occurs: specifies an array containing every variable and their values in use when the error occurs.

e.g. <?php  
\$file = fopen ("abc.txt", "w");  
e.g. <?php  
fopen ("filename", "emode")  
syntax:  
in which file needs to be opened.  
is to be opened and 2nd parameter tells about mode  
of this function contains the name of the file in which  
: open: this function is used to open a file. It's parameter  
function is file handling

some task to be done needs file processing  
file handling is important for any application as  
File handling in PHP  
output:  
echo "File";  
set\_error\_handler ("custom\_error");  
function custom\_error (\$error)  
{  
echo "File";  
echo "Error";  
echo "Custom Error";  
e.g. PHP

\$ filedata = fread(\$file, filesize());

\$ file = fopen(\$filename, "r");  
\$ filename = abc.txt;

e.g. in php

parameters. It filename file size in bytes

data are read using fread. It takes two  
lines of code to open the file using fopen then  
read the contents.

9. x = new file is created for write only

the file is preserved.

points to the end of the file existing data is  
written to file provided for read. File pointer

5. a+: file is open for read or write.

4. ut: opens file for read & write. If file does not  
exist then content of the file is erased.  
exists then file is created and if file already  
exists then content of the file is erased.

3. a: file is opened for write only. File pointer  
points to the end of the file existing data  
in the file is preserved.

2. a: opens file in read only.

1. w: opens the file for write only. If file does  
not exist then new file is created and if file  
exists already then content of the file is  
erased.

			date
			Page No.

6

file close (\$file);

file = fopen ("abc.txt", "w");

close

eg.

argument is file which needs to be closed  
file is closed using close function its

iv. close :

function close (\$file, \$text);

\$text = "fyscs";