

### ❖ htmlentities()

- when we work with form at that time it can be happen that user might enter html code inside text fields and this can be dangerous for our program
- to avoid this kind of situation we can use htmlentities() in PHP
- it will convert all the html code that user enter in text field into simple text
- Example :

```
$first_name = htmlentities($_POST['first_name']);
```

### ❖ Sessions in PHP

- Session is way to store the information that can be used across multiple pages
- Sessions are generally stored on the server so user can not destroy it
- Sessions are generally used to keep track of user and user's actions
- There is a method called **session\_start()** which is used to start the session
- Session\_start() will be the first line of every php file where you want to use session
- So we can say that session\_start() will we used in every file of our application
- We can set session variables with the help of global variable **\$\_SESSION**
- We can display or access session variables using **\$\_SESSION**
- We just have to pass session variable name with **\$\_SESSION** like ,  
**\$\_SESSION['username'];**
- We can also destroy or clear session by some methods which are as follows
- **session\_destroy()** will destroy all the data registered to a session
- **session\_unset()** will free all the session variables
- we can also unset session variables using **unset()** method
- we can also display the list of sessions by printing the **\$\_SESSION** array
- Example :

```
// set session  
session_start();  
$_SESSION['username']='Malcom';  
$_SESSION['age']=23;
```

```
// display session variable  
session_start();  
echo $_SESSION['username'];  
echo $_SESSION['age'];
```

**// unset single session variable**

```
unset($_SESSION['username']);
```

**// free all the session variables**

```
session_unset();
```

**// destroy all the data registered to session variables**

```
session_destroy();
```

**// display all the session list**

```
print_r($_SESSION);
```

#### ❖ **Cookies in PHP**

- Cookies are useful to store small information about the user.
- We can use it like sessions but the cookies are stored on user's computer
- So each time when same user request to access the page at that time the stored cookie will also sent
- User can also disable it too. That's why it is not secure
- There are mainly two methods to create cookie
- **setcookie()** and **setrawcookie()**
- **setcookie()** will send the cookie
- **setrawcookie()** will send a cookie without urlencoding the cookie value
- setcookie() and setrawcookie() both will take total 7 arguments out of which only first one 'name' is required and all other six arguments are optional.
- Generally we pass three arguments to setcookie(). First is cookie name , second is cookie value and third is cookie expiration time
- We can also get the cookie value through **\$\_COOKIE** global variable
- If we want to delete cookie then we have to use setcookie() method and we have to set expiration time in past.
- We can see all the cookies list by printing the **\$\_COOKIE** array
- Example :

**// set cookie for 1 minute**

```
setcookie('username' , 'Malcom' , time() + 60);
```

```
// display cookie
```

```
echo $_COOKIE['username'];
```

```
// modify cookie
```

```
setcookie('username' , 'Smith' , time()+60);
```

```
// delete cookie or remove cookie
```

```
setcookie('username' , 'Malcom' , time() - 60);
```

```
// display list of cookies
```

```
print_r($_COOKIE);
```

```
// checking if cookie is set or not ?
```

```
if(!isset($_COOKIE['username'])){\
```

```
    echo 'cookie is not set';
```

```
}
```

```
else {\
```

```
    echo $_COOKIE['username'];
```

```
}
```

## ❖ File Handling

- Using some functions of file handling in php we can handle files easily.
- We can write , read , append data into files
- There are two main function that we have to use first to open a file and close the file which are **fopen()** and **fclose()**
- **fopen()** takes two arguments. First it takes file name and second it takes mode in which we have to open file
- example :

```
// open file
```

```
$handle = fopen('file.txt' , 'w');
```

- the modes are as follows :
  - **r** : open a file for read only. File pointer starts at the beginning of the file

- **w** : open a file for write only. Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file
  - **a** : open a file for write only. The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist
  - **x** : creates a new file for write only. Returns FALSE and an error if file already exists
  - **r+** : open a file for read/write. File pointer starts at the beginning of the file
  - **w+** : open a file for read/write. Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file
  - **a+** : open a file for read/write. The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist
  - **x+** : creates a new file for read/write. Returns FALSE and an error if file already exists
- after opening the file we can read data of file , write data into file and also can append data into file.
  - To write data into file there is a function called **fwrite()**.
  - fwrite() mainly take two arguments. First is name of file and second is data that we have to write
  - example :

**// write into file**

```
$handle = fopen('file.txt' , 'w');
fwrite($handle , 'this will be written');
```

- after writing file it's time to read the data written into file
- we can read the data using **fread()**
- it also takes two arguments. First is file name and second is file size.
- Example :

**// read data from file**

```
$handle = fopen('file.txt' , 'r');
fread($handle , filesize('file.txt'));
```

- We can also read one line from the file using **fgets()**
- It takes only one argument which is file name
- Example :

**// read one line from file**

```
$handle = fopen('file.txt' , 'r');
fgets($handle);
```

- We can also read data until end of file using **feof()**
- it just take one argument which is file name
- example :

```
// read one line each time till end of file
```

```
$handle = fopen('file.txt' , 'r');

while(!feof($handle)) {

    echo fgets($handle) . '<br>';

}
```

- after reading we can also append some data into existing file using append mode.
- Example :

```
// append the data into file
```

```
$handle = fopen('file.txt' , 'a');

fwrite($handle , 'Microsoft - Satya nadella' . "\n");
```

- After doing all these operations now it's time to close the file.
- If we have done working on the file the n at last we should close it.
- We can do this by a simple function called **fclose()**
- It just take a single argument which is file name.
- Example :

```
// close the file
```

```
fclose('file.txt');
```

#### ❖ **Explode() & implode()**

- Explode() is used to convert string into array
- We can simply convert any string into an array using explode()
- it just takes mainly two arguments. First is separator and second is string
- example :

```
<?php

$str = "Hello world. It's a beautiful day.";

print_r (explode(" ", $str));

?>
```

- implode() is totally reverse of explode()
- implode() converts an array into a string

- it also take two arguments. First is separator and second is array
- example :

```
<?php

$arr = array('Hello','World!','Beautiful','Day!');

echo implode(" ",$arr);

?>
```

#### ❖ opendir() & readdir()

- opendir() opens a directory handle
- it takes one required argument which is path of the directory.
- Example :

```
// open a directory handle
$directory = 'files';
opendir($directory.'/');
```

- readdir() function returns the name of the next entry in a directory.
- It takes one optional argument which is directory path
- Example :

```
// read a directory
$handle = opendir('files/');
readdir($handle);
```

#### ❖ file\_exists()

- file\_exists() function return true if the file is exist otherwise return false
- we can use this function to check if file is there or not
- it only takes one parameter which is file name
- example :

```
// check if file exist or not ?

$filename = 'CEOs.txt';

if(file_exists($filename)) {

    echo 'file exist';

} else {

    echo 'sorry , file don't exist';

}
```

```
}
```

#### ❖ **unlink() & rename()**

- unlink() function deletes a file
- it takes only one required argument which is file name
- example :

```
// check if file exist or not ?
```

```
$filename = 'CEOs.txt';  
if(unlink($filename)) {  
    echo 'file deleted';  
}
```

- rename() functions rename the file
- it takes generally two arguments. First is old file name and second is new file name
- example :

```
// rename the file
```

```
$filename = 'file_to_rename.txt';  
$newname = rand(1000 , 9999);  
  
if(@rename($filename , $newname.'.txt')) {  
    echo '<strong>'. $filename .'</strong> has been renamed to <strong>'.  
$newname .'</strong>';  
}  
else {  
    echo '<strong>'. $filename .'</strong> is either renamed already or does  
not exist';  
}
```

#### ❖ **upload files**

- we can upload files into php
- first when we upload any file it is stored in temp storage of xampp
- then we have to move that file at our preferred destination
- there is a function named **move\_uploaded\_file()** to move the file to destination where we want to upload all the files
- it takes two arguments. First is temporary storage location and second is our preferred storage location where we want to move the file.
- Example :

```
if(isset($_POST['submit'])){

    $name = $_FILES['photo']['name'];

    $max_size = 2048000;

    $extension = strtolower(substr($name , strpos($name , '.')+1));

    $type = $_FILES['photo']['type'];

    $file_size = $_FILES['photo']['size'];

    $temp_name = $_FILES['photo']['tmp_name'];


    if(isset($name)) {

        if(!empty($name)) {

            if(($extension == 'jpg' || $extension == 'jpeg') && $type ==
'image/jpeg' && $file_size < $max_size) {

                $location = 'uploads/';

                if(move_uploaded_file($temp_name , $location.$name)) {

                    echo 'uploaded....';

                }

                else {

                    echo 'there was an error';

                }

            }

            else {

                echo 'please select proper file with jpeg/jpg format and should be
less than 2 mb';

            }

        }

        else {

            echo 'please select any file';

        }

    }

}
```



}

