Working with File and Directories

File

- A file is an ordered **sequence of bytes.**
- **File handling** is an important part of any web application. You often need to open and process a file for different tasks.
- Different operations we perform on the file
  - Opening a file
  - Closing a file
  - Reading a file
  - Writing into a file
  - Detect end of a file
  - Deleting a file
  - Renaming a file
  - Copy contents of a file

Functions related to File

- fopen() : This is used to **open a file** ,returning a file handle associated with opened file .It takes two arguments as fname, mode. Example : $fp = fopen("a.txt","r");

Mode	Purpose
r	Open for reading only; place the file pointer at the beginning of the file
r+	Open for reading and writing; place the file pointer at the beginning of the file.
w	Open for writing only; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.
w+	Open for reading and writing; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.
a	Open for writing only; place the file pointer at the end of the file. If the file does not exist, attempt to create it.
a+	Open for reading and writing; place the file pointer at the end of the file. If the file does not exist, attempt to create it.

Functions related to File

- fclose() : This is used to **close file**, using its associated file handle as a single argument. Example : fclose($fp);
- fread() : This function is used **to extract a character string** from a file and takes two arguments, a file handle and optional parameter a integer length.

  Example: fread($fp, filesize("a.txt"));
- fgets() : This function is used **to read set of characters** it takes two arguments, a file handle and optional parameter a integer length. Example: fgets($fp);
- fgetc() : Function can be used **to read one character from file at a time**. It takes a single argument ,a file handle and return just one character from the file .It returns false when it reached to the end of file. Example : fgetc($fp);
- feof() : This function returns TRUE if end of **file is reached** else returns FALSE. Example :feof($fp);

Functions related to File

- readfile() : This function **prints content of file** without having a call to fopen(). It takes a filename as its argument, reads a file and then write it to standard output returning the number of bytes read. Exmaple: readfile("test.txt");
- file() : This function will return **entire contents of file in array**. This function will automatically opens, reads, and closes the file. It has one argument : a string containing the name of the file. Exmaple: file("test.txt");
- fwrite() : This function is used **to write data to a file** and takes two arguments, a file handle and a string. Example : fwrite($fp,"PHP");
- fputs() : This is simply an alias for **fwrite()**
- ftell(): It takes file handle as an argument and returns the current offset (in bytes) of the corresponding **file position indicator**. Example: ftell($fp);
- rewind(): It accepts a file handle as an argument and reset the corresponding file position indicator to the **beginning of file**. Example : rewind($fp);

Functions related to File

- fseek() : It takes file handle and integer offset(starting index), offset type as arguments .It will **move file position indicator** associated with file pointer to a position determined by offset. By default this offset is measured in bytes from the beginning of the file. The third argument is optional, can be specified as:
  - SEEK_SET:- Moves the position pointer offset bytes from **beginning of file**
  - SEEK_CUR:- Moves the position pointer offset bytes from its **current position**(default)
  - SEEK_END:- Moves the position pointer offset bytes from **end of the file**.

  Example: fseek($fp,0,SEEK_END);
- file_exists() : It takes file name with detail path as an argument and returns true if **file is there** otherwise it returns false. Example: file_exists("test.txt");
- filesize() : It takes file name as an argument and returns **total size** of file(in bytes)
- rename(): It takes two argument as old name and new name and renames the file with new name. Exmaple: rename("test.txt", "abc.txt");
- unlink(): It takes a single argument referring to the name of file we want **to delete**.

Code to read file contents

Method 1 : Using fread()

```
<?php
    $fp = fopen(“test.txt", "r") or die("Unable to open file!");
    echo fread($fp,filesize(“test.txt"));
    fclose($fp);
?>
```

Method 2: Using fgets()

```
<?php
    $fp = fopen(“test.txt", "r") or die("Unable to open file!");
    while(!feof($fp))
    {
            echo fgets($fp);
            echo "<br>";
    }
    fclose($fp);
?>
```

Code to read file contents

Method 3 : Using fgetc()

```
<?php
   $fp = fopen(“test.txt", "r") or die("Unable to open file!");
   while(!feof($fp))
   {
         echo fgetc($fp);
         echo "<br>";
   }
   fclose($fp);
?>
```

Method 4 : Using readfile()

```
<?php
   echo readfile(“test.txt”);
?>
```

Code to write contents into the file

Method 1 : Using fwrite()

```
<?php
   $fp = fopen(“test.txt", “w") or die("Unable to open file!");
   $txt = “Server side Scripting\n";
   fwrite($fp, $txt);
   fwrite($myfile, $txt);
   fclose($fp);
?>
```

Code to write contents into the file

Method 2 : Using fputs()

```
<?php
   $fp = fopen(“test.txt", “w") or die("Unable to open file!");
   $txt = “Server side Scripting\n";
    fputs($fp, $txt);
   $txt = " Cliend side Scripting ";
    fputs($myfile, $txt);
   fclose($fp);
?>
```

Code to copy contents from one file to another file

```
<?php
   $fp1 = fopen(“abc.txt", “r") or die("Unable to open file!");
   $fp2 = fopen(“pqr.txt", “w+") or die("Unable to open file!");
   while(!feof($fp1))
   {
                $ch=fgets($fp1);
                fputs($f2, $ch);
   }
   rewind($fp2);
   readfile($fp2);
   fclose($fp1);
?>
```

Working with Directories

Function	Description
chdir()	Changes the current directory
chroot()	Changes the root directory
closedir()	Closes a directory handle
dir()	Returns an instance of the Directory class
getcwd()	Returns the current working directory
opendir()	Opens a directory handle
readdir()	Returns an entry from a directory handle
rewinddir()	Resets a directory handle
scandir()	Returns an array of files and directories of a specified directory

Open a directory, read its contents, then close

```
<?php
$dir = "C:/xampp/htdocs/SYBCA/uploads/";
if (is_dir($dir)){
  if ($dh = opendir($dir)){
    while (($file = readdir($dh)) !== false){
      echo "filename:" . $file . "<br>";
    }
    closedir($dh);
  }
}
?>
```

Use of rewinddir()

```
<?php
$dir = "C:/xampp/htdocs/SYBCA/uploads/";
if (is_dir($dir)){
  if ($dh = opendir($dir)){
    while (($file = readdir($dh)) !== false){
      echo "filename:" . $file . "<br>";
    }
    rewinddir($dir);
    while (($file = readdir($dh)) !== false){
      echo "filename:" . $file . "<br>";
    }
    closedir($dh);
  }
}
?>
```

Use of scandir()

```
<?php
   $dir = "C:/xampp/htdocs/SYBCA/uploads/";
   // Sort in ascending order
   $a = scandir($dir);

   // Sort in descending order
   $b = scandir($dir,1);

   print_r($a);
   print_r($b);
?>
```

File Uploading

HTML File

```
<html>
<body>

<form action="upload.php" method="post" enctype="multipart/f
orm-data">
  Select image to upload:
  <input type="file" name="fileToUpload">
  <input type="submit" value="UploadImage" name="submit">
</form>

</body>
</html>
```

File Uploading

PHP File

```php
<?php
$target_dir = "uploads/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
$imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));

// Check if image file is a actual image or fake image
if(isset($_POST["submit"])) {
  $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
  if($check == true) {
    echo "File is an image - " . $check["mime"] . ".";
    $uploadOk = 1;
  } else {
    echo "File is not an image.";
    $uploadOk = 0;
  }
}
```

File Uploading

PHP File

```
// Check if file already exists
if (file_exists($target_file)) {
  echo "Sorry, file already exists.";
  $uploadOk = 0;
}

// Check file size
if ($_FILES["fileToUpload"]["size"] > 500000) {
  echo "Sorry, your file is too large.";
  $uploadOk = 0;
}

// Allow certain file formats
if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType
!= ”jpeg” && $imageFileType != "gif" )
{
  echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
  $uploadOk = 0;
}
```

File Uploading

PHP File

```
if ($uploadOk == 0)
{
  echo "Sorry, your file was not uploaded.";
}
else {
  if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file))
  {
    echo "The file ". htmlspecialchars(basename( $_FILES["fileToUpload"]["name"])). "
has been uploaded.";
  }
  else {
    echo "Sorry, there was an error uploading your file.";
  }
}
?>
```

File downloading

```php
<?php

    // Initialize a file URL to the variable
    $url = 'C:\MIT-Logo.jpg';

    // Use basename() function to return the base name of file
    $file_name = basename($url);

    // Use file_get_contents() function to get the file
   // from url and use file_put_contents() function to
   // save the file by using base name

    if (file_put_contents($file_name, file_get_contents($url)))
    {
        echo "File downloaded successfully";
    }
    else
    {
        echo "File downloading failed.";
    }
?>
```

Thank you