

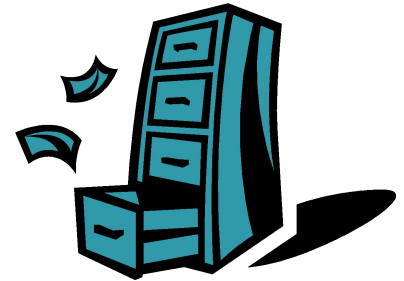
---

# File Handling with PHP



# Files and PHP

- File Handling
  - Data Storage
    - Though slower than a database
  - Manipulating uploaded files
    - From forms
  - Creating Files for download



# Open/Close a File

- A file is opened with `fopen()` as a “stream”, and PHP returns a ‘handle’ to the file that can be used to reference the open file in other functions.
- Each file is opened in a particular **mode**.
- A file is closed with `fclose()` or when your script ends.



# File Open Modes

<code>'r'</code>	Open for reading only. Start at beginning of file.
<code>'r+'</code>	Open for reading and writing. Start at beginning of file.
<code>'w'</code>	Open for writing only. Remove all previous content, if file doesn't exist, create it.
<code>'a'</code>	Open writing, but start at END of current content.
<code>'a+'</code>	Open for reading and writing, start at END and create file if necessary.



# File Open/Close Example

```
<?php
// open file to read
$storead = fopen('some/file.ext', 'r');
// open (possibly new) file to write
$towrite = fopen('some/file.ext', 'w');
// close both files
fclose($storead);
fclose($towrite);
?>
```



# Now what..?

- If you open a file to read, you can use more in-built PHP functions to read data..
- If you open the file to write, you can use more in-built PHP functions to write..



# Reading Data

- There are two main functions to read data:
- **fgets (\$handle , \$bytes)**
  - Reads up to \$bytes of data, stops at newline or end of file (EOF)
- **fread (\$handle , \$bytes)**
  - Reads up to \$bytes of data, stops at EOF.



# Reading Data

- We need to be aware of the End Of File (EOF) point..
- **feof (\$handle)**
  - Whether the file has reached the EOF point. Returns true if have reached EOF.





# Data Reading Example

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

```
while (!feof($handle)) {  
    echo fgets($handle, 1024);  
    echo '<br />';  
}
```

```
fclose($handle);
```



# Data Reading Example

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

```
while (!feof($handle)) {  
    echo fgets($handle, 1024);  
    echo '<br />';  
}
```

Open the file and assign the resource to \$handle

```
fclose($handle);
```



# Data Reading Example

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

```
while (!feof($handle)) {  
    echo fgets($handle, 1024);  
    echo '<br />';  
}
```

```
fclose($handle);
```

While NOT at the end of the file, pointed to by \$handle, get and echo the data line by line



# Data Reading Example

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

```
while (!feof($handle)) {  
    echo fgets($handle, 1024);  
    echo '<br />';  
}
```

Close the file

```
fclose($handle);
```



# File Open shortcuts..

- There are two 'shortcut' functions that don't require a file to be opened:
- `$lines = file($filename)`
  - Reads entire file into an array with each line a separate entry in the array.
- `$str = file_get_contents($filename)`
  - Reads entire file into a single string.



# Writing Data

- To write data to a file use:
- `fwrite ($handle, $data)`
  - Write \$data to the file.



# Data Writing Example

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

```
fwrite($handle, "\nFred:Male");
```

```
fclose($handle);
```



# Data Writing Example

Open file to append data (mode 'a')

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

```
fwrite($handle, "\nFred:Male");
```

```
fclose($handle);
```

Write new data (with line  
break after previous data)





# Other File Operations

- Delete file
  - `unlink ( 'filename' ) ;`
- Rename (file or directory)
  - `rename ( 'old name' , 'new name' ) ;`
- Copy file
  - `copy ( 'source' , 'destination' ) ;`
- And many, many more!
  - [www.php.net/manual/en/ref.filesystem.php](http://www.php.net/manual/en/ref.filesystem.php)



# Dealing With Directories

- Open a directory
  - `$handle = opendir('dirname');`
    - `$handle` 'points' to the directory
- Read contents of directory
  - `readdir($handle)`
    - Returns name of next file in directory
    - Files are sorted as on filesystem
- Close a directory
  - `closedir($handle)`
    - Closes directory 'stream'



# Directory Example

```
$handle = opendir ('./');
```

```
while (false !== ($file=readdir ($handle)))  
{  
    echo "$file<br />";  
}
```

```
closedir ($handle);
```



# Directory Example

```
$handle = opendir ('./');
```

Open current directory

```
while (false !== ($file=readdir ($handle)))  
{  
    echo "$file<br />";  
}  
  
closedir ($handle);
```



# Directory Example

```
$handle = opendir('./');
```

```
while(false !== ( $file=readdir($handle) ))  
{  
    echo "$file<br />";  
}
```

```
closedir($handle);
```

Whilst `readdir()` returns a name, loop through directory contents, echoing results



# Directory Example

```
$handle = opendir('./');
```

```
while(false !== ($file=readdir($handle)))  
{  
    echo "$file<br />";  
}
```

```
closedir($handle);
```

Close the directory stream



# Other Directory Operations

- Get current directory
  - `getcwd()`
- Change Directory
  - `chdir('dirname');`
- Create directory
  - `mkdir('dirname');`
- Delete directory (MUST be empty)
  - `rmdir('dirname');`
- And more!
  - [www.php.net/manual/en/ref.dir.php](http://www.php.net/manual/en/ref.dir.php)



# Review

- Can open and close files.
- Can read a file line by line or all at one go.
- Can write to files.
- Can open and cycle through the files in a directory.

