

# FTP and Uploading your Files

(50 points)

## Objectives

- Use an FTP (File Transfer Protocol) Client
- Manage files using FTP
- Where to store files on your server
- Add index.php files to each directory

## Introduction

So far we have been working in our local development environment. Now it is time to upload your work to your web host. There are a couple of ways to do it

1. FTP or (SFTP - Secure Shell FTP)
2. By the file manager provided by your web host

## FTP

We use FTP to move files from your local computer to another computer. In this case it will be to your webhost. For this example we are going to use a free program called Filezilla. Note, that there are hundreds of FTP clients you can download if you don't like Filezilla.

First you will need to get your FTP login information. If you are using Siteground,

- Login to siteground.com
- Click on the My Accounts tab
- You can find your FTP information here. There is also a link to the right of that information with an FTP tutorial. It provides a lot of information about using FTP. For now we just want to upload to the part of the server that hosts the web files; however, keep in mind that you can use your web host to store files in other parts of the server. This would be similar in concept to having a Dropbox or Google Drive account

If you are not using SiteGround then please use a search engine and search on your web hosts' name + FTP

## Video Tutorials

If you are new to FTP, here are a couple of short video tutorial on setting up and managing files with Filezilla.

1. [Configuring your website using Filezilla](#)
2. [Managing files once you are connected.](#)
3. [Uploading file using Filezilla](#)

## Index files

Now for the drudgery.

When you are running XAMPP and have your browser open with localhost you are able to see **all** of the files in each of the directories. This is fine when you are working locally but it is a security risk when you are on a shared server so most web hosts turn off that feature leaving you wondering why you can't see your files. This forces you to create an index.html or index.php (there are variations on this) file that points to the files in your directory. For instance, in my web182 folder I have an index.php file that looks like

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>WEB 182</title>
</head>
<body>
<h1>WEB 182: PHP Programming</h1>

<ul>
  <li><a href="book-source">Book source</a></li>
  <li><a href="my-code">My code</a></li>
</ul>

</body>
</html>
```

Notice that it has two links that point to the Book source and My code.

The **index.php** file I have inside the **my-code** directory looks like

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>My WEB 182 Code</title>
</head>
<body>
```

```
<h1>Code for class</h1>
<ul>
  <li><a href="ch01">ch01</a></li>
  <li><a href="ch03">ch03</a> </li>
</ul>

</body>
</html>
```

[Here is a link to the web182 part of my site](#) (please ignore the fact that there is no style ).

## HTML vs. PHP

Even though I'm only using HTML code for each of my files I use the .php in case I want to add PHP code to my index file later. Even something simple like adding a date will force you to use PHP (or another language) so it is best to plan ahead.

## Assignment

Create an **index.php** file for each of your directories starting with your **web182** directory then upload the files so you have them available on your web host. You **do not** need to create an **index.php** file for the book's source code.

## Submit your URL

After you have uploaded your index.php files and any other code from the previous assignments, take the time to make sure your links work.

Enter the URL in the Moodle Notes section.