Make sure you can log into the server via an FTP client like Filezilla!

1. Write the bare XHTML markup

Start with just a plain file with the following markup:

```
<!DOCTYPE html>
<html>
<head>
     <title>MyProject</title>
</head>

<body>

</body>
</html>
```

2. Saving a Plaintext File

Save your file as “save-plaintext.php5”. Note that for some servers, they run PHP4 by default, but you can force PHP5 by adding the 5 to the extention.

Within the <body> tags, open a new php section with <? and ?>.
Create an array, such as:

```
$students = array();
$students[0] = array( "name" => "Abby Adams",         "age" => 25 );
$students[1] = array( "name" => "Bailee Bakalov",     "age" => 34 );
$students[2] = array( "name" => "Cara Chen",          "age" => 16 );
```

Next, we'll create a file path variable, as well as the file handler variable.

```
$path = "subfolder/students.txt";
$fileHandler = fopen( $path, "w" );
```

Note that, if the *subfolder* does not exist on the server, you will get PHP errors. We can make folders with PHP, but for now just create that subfolder via Filezilla.

Then we can write the contents to the file with a **for each** loop:

```
fwrite( $fileHandler, "STUDENTS:\n" );
foreach ( $students as $student )
{
     fwrite( $fileHandler, $student["name"] );
     fwrite( $fileHandler, "\t" );
     fwrite( $fileHandler, $student["age"] );
     fwrite( $fileHandler, "\n" );
}
```

And make sure to close the file when we're done:

```
fclose( $fileHandler );
```

Now if you upload the file and navigate to the appropriate URL
(http://www.alketo.info/inclass/*yourfolder*/save-plaintext.php5)
You will get a blank page. However, in the *subfolder* directory, you should see your text file.

You can move that text file to your hard-drive by clicking & dragging it from the right side to the left size. Open the file and make sure everything was written to the file.

3. Writing a CSV File

To write and read CSV files, we can use PHP5's **fputcsv** and **fgetcsv** functions.

Make a copy of the “save-plaintext.php5” file and name it “save-csv.php5” file. Saving the CSV file will be similar to what we were doing before, since we have to write one row at a time. However, we can pass an array into the **fputcsv** function and it will handle the formatting for us.

Remember to change the filename to .csv instead of .txt:

```
$path = "subfolder/students.csv";
```

Then change up the for loop:

```
fwrite( $fileHandler, "NAME,AGE\n" );
foreach ( $students as $student )
{
     fputcsv( $fileHandler, $student );
}
```

When you open this .php5 file in the web browser, afterwards you should be able to see your .csv file in the subfolder.

4. Writing a JSON File

With PHP's **json_decode** and **json_encode** functions, you can easily convert between JSON Strings and PHP Arrays. Sometimes, you'll see RSS feeds in the form of JSON instead of XML, JSON can be useful for passing data between client and server (via AJAX), and it might even be useful for storing information for your webpage.

Using the previous program, the main thing that is going to change is how we output the array. Make a copy of “save-plaintext.php5” and name it “save-json.php5”.

Erase this code:

```
fwrite( $fileHandler, "STUDENTS:\n" );
foreach ( $students as $student )
{
     fwrite( $fileHandler, $student["name"] );
     fwrite( $fileHandler, "\t" );
     fwrite( $fileHandler, $student["age"] );
     fwrite( $fileHandler, "\n" );
}
```

And replace it with:

```
fwrite( $fileHandler, json_encode( $students ) );
```

Also, make sure to change the output file from “output.txt” to “output.json”:

```
$path = "subfolder/students.json";
```

Now, when you navigate to that file, within the subfolder you should now see *output.json*. It should look something like:

```
[{"name":"Abby Adams","age":25},{"name":"Bailee Bakalov","age":34},{"name":"Cara
Chen","age":16}]
```

Note that the square brackets denote **array**, while the curly braces denote **key-value pair**. We have an array of key-value pairs.

Fill code for write-plaintext.php5:

```
<!DOCTYPE html>
<html>
<head>
     <title>PHP Sample</title>
</head>

<body>
     <?
     /* Open a file and write out plaintext */
     $filepath = "files/output-plaintext.txt";

     $fileHandler = fopen( $filepath, "w" ); // Open for write
     if ( $fileHandler == false )
     {
          print_r( error_get_last() );  // Error message (PHP5)
          exit();   // Stop script
     }

     fwrite( $fileHandler, "This is plaintext \n Hello world" );

     fclose( $fileHandler );
     ?>

</body>
</html>
```