

# Tutorial 8

Editor – Brackets

## Goals

Introduction to PHP and MySQL.

- Set up and configuration of Xampp
- Learning Data flow

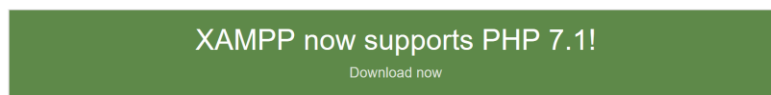
Things to note: Each week Xampp will need to be installed. Xampp is Windows software, similar software is available for Mac, called Mamp.

## Installing and configuring Xampp

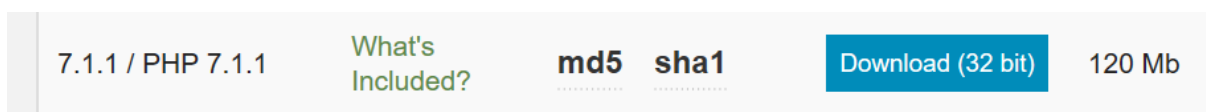
Go to the website: <https://www.apachefriends.org/index.html>



Download the Xampp with PHP 7.1, click on the following link



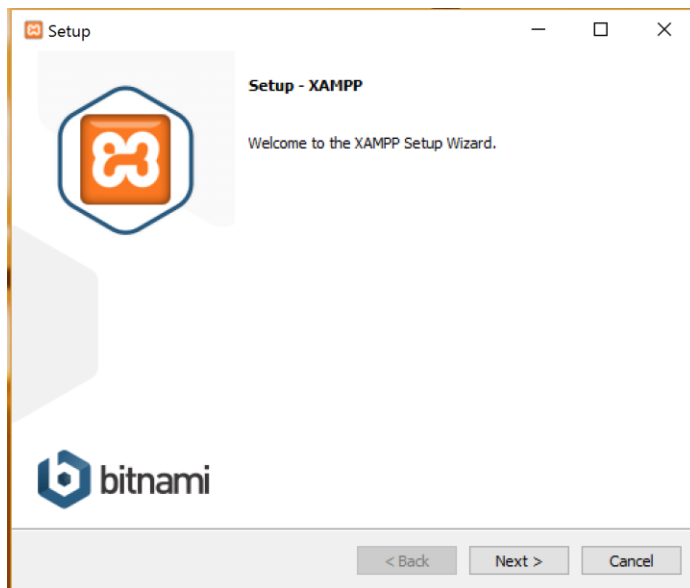
Then click on



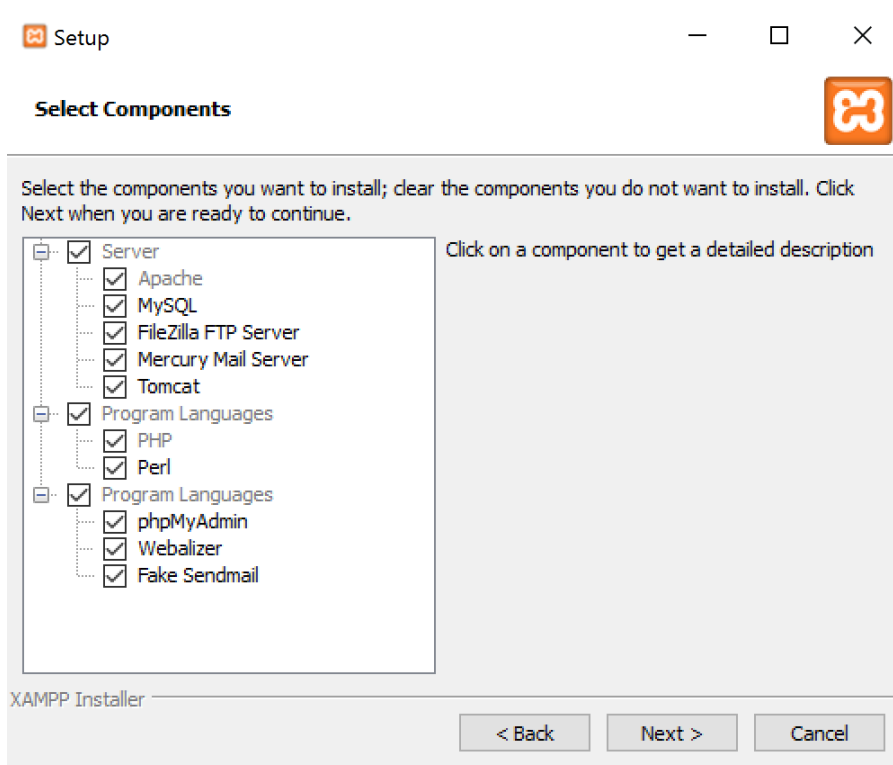
Download to the desktop, Once there, double click on the file and following the prompts



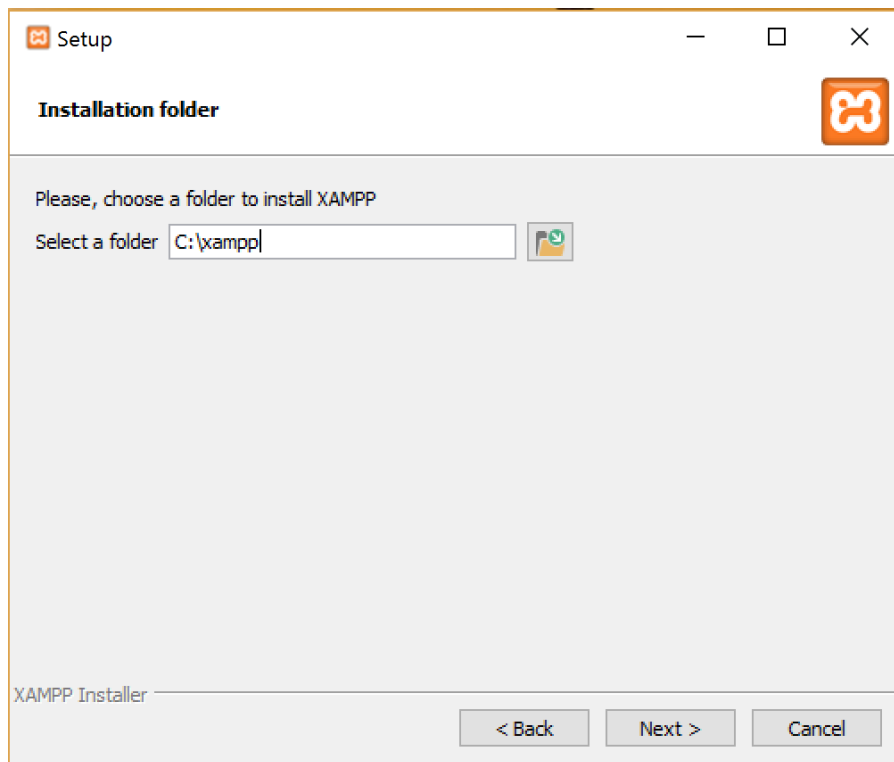
Click Next



Click Next

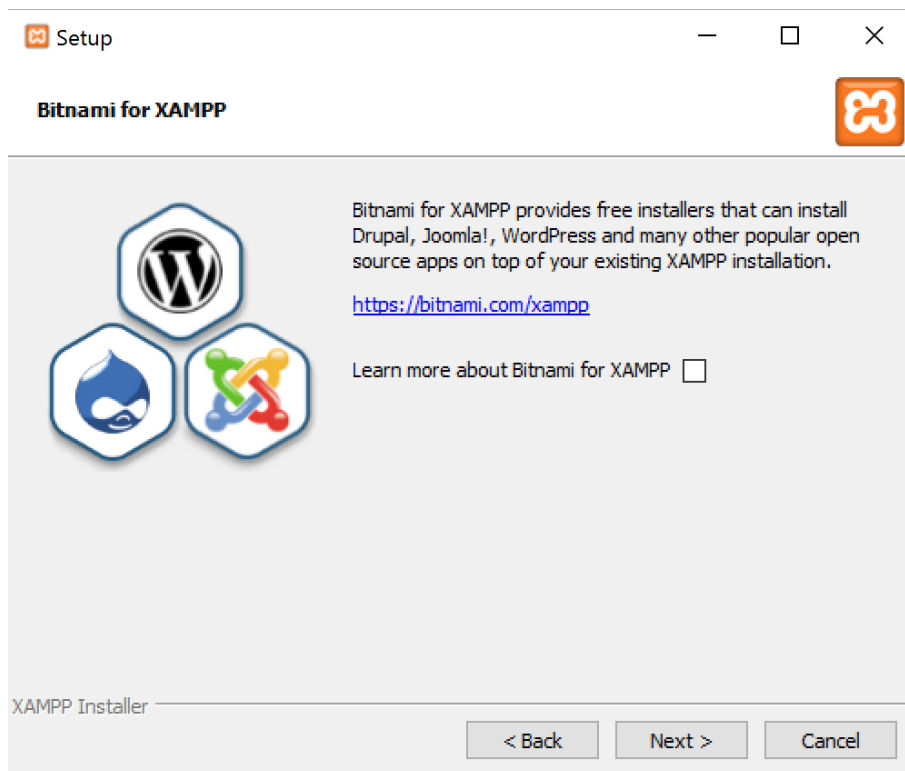


Click Next

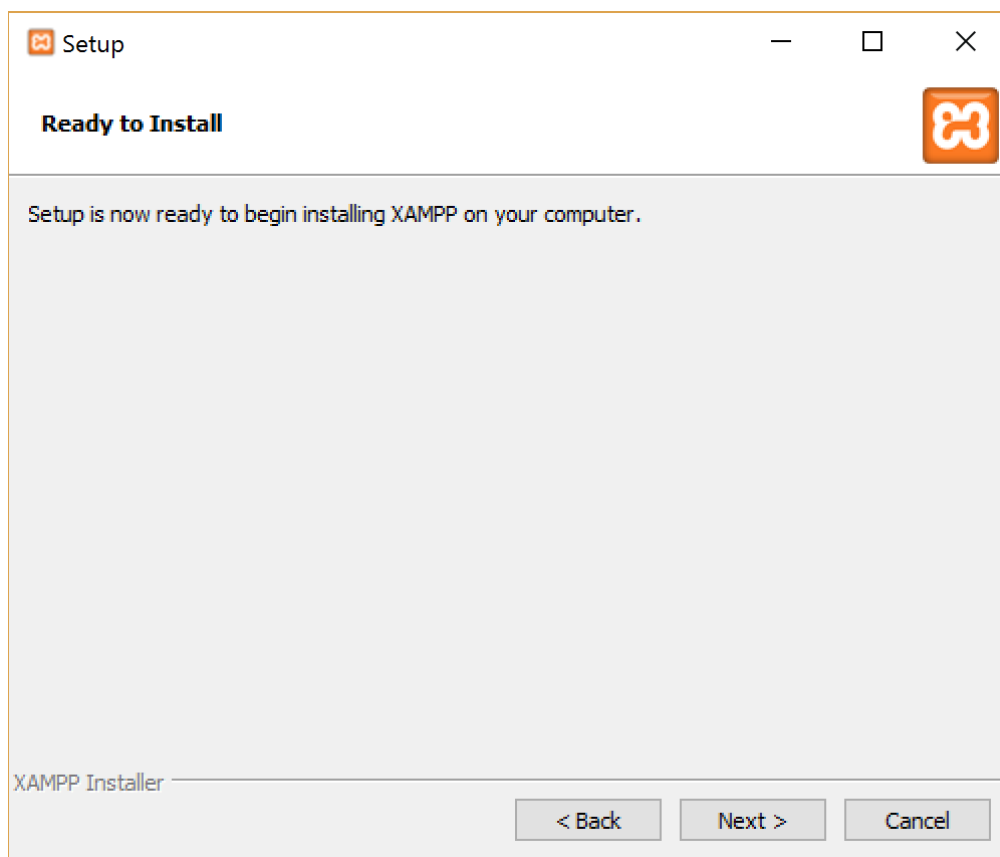


NB: This is the installation folder, all of your web pages will belong in c:\xampp\htdocs

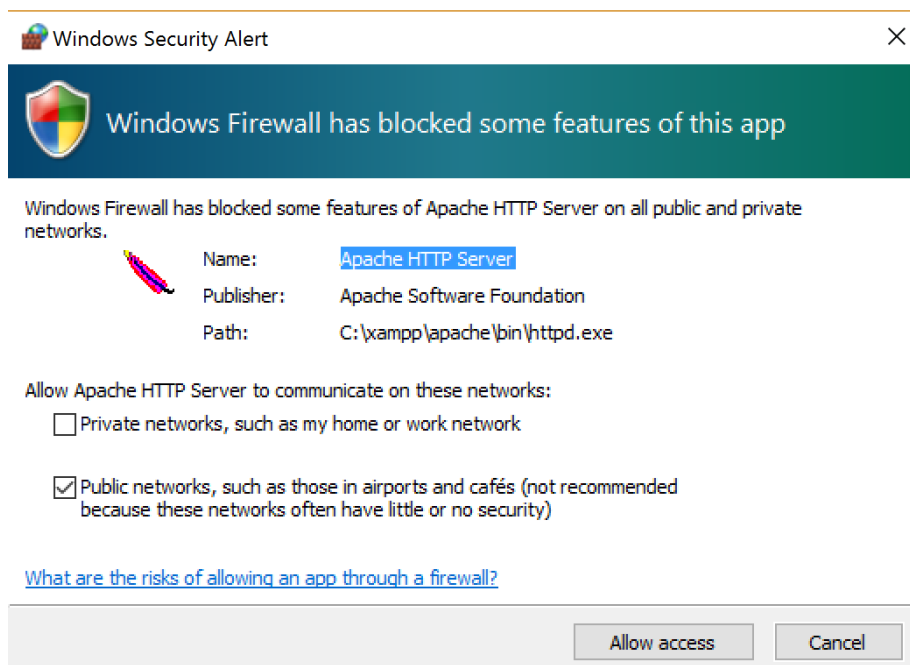
Untick Bitnami and click Next



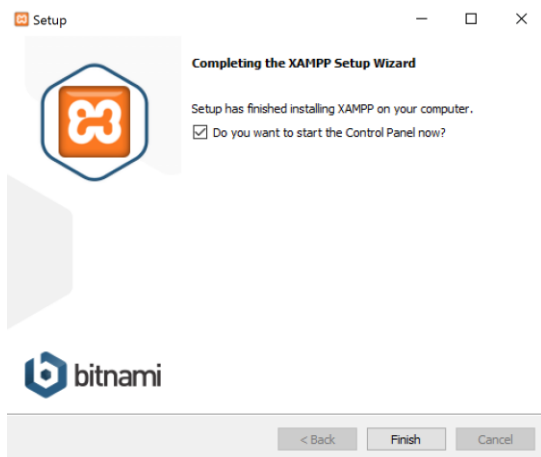
Click next to install



Allow any pop ups such as Apache or MySql

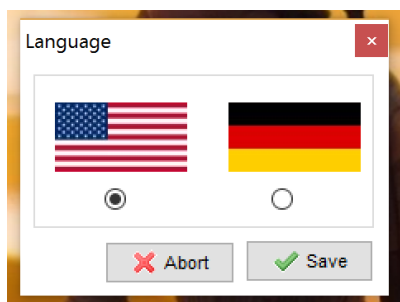


Then click finish



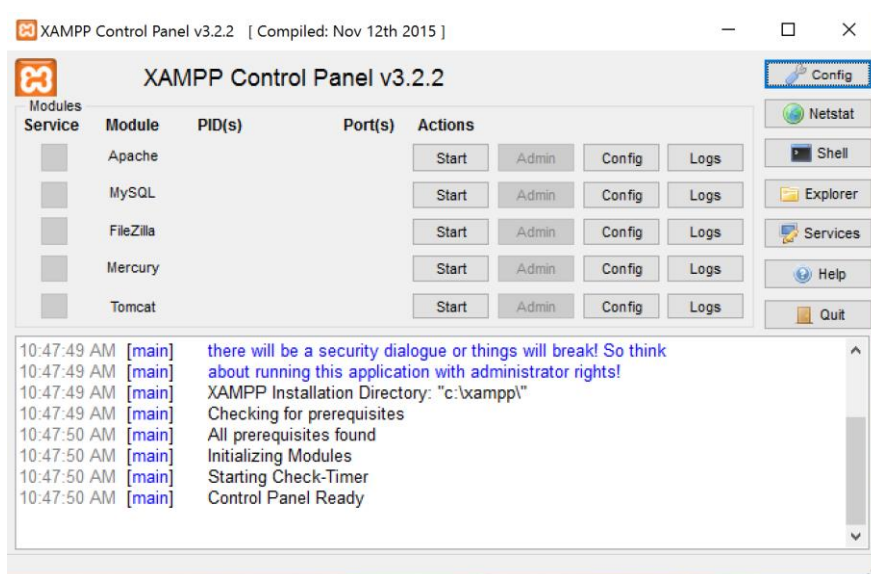
Now we configure Xampp.

Assuming you left the “Do you want to start the Control Panel” ticked, you should see the following on your screen.



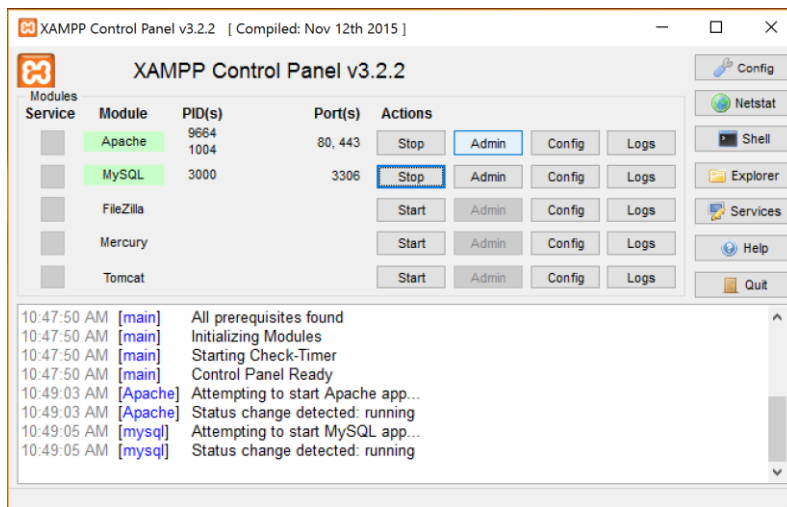
Click on Save

The following should appear:



From here we need to click Start on Apache and MySQL.

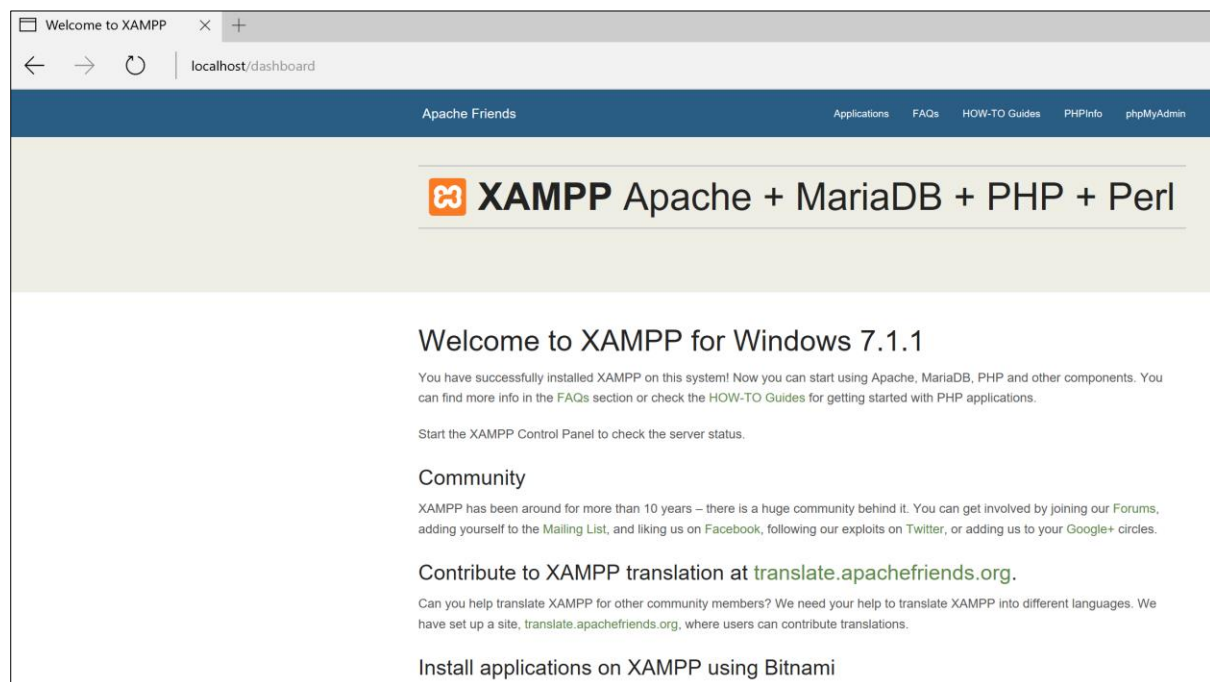
You should see the following.



We now have the Apache webserver up and running with MySQL.

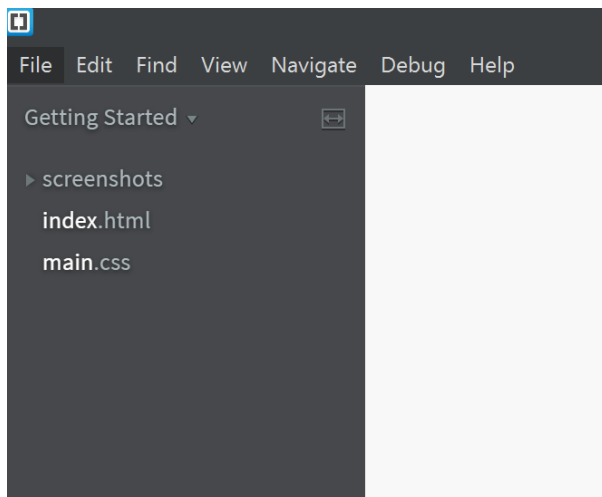
To check that it is working load up a browser and type localhost into the address bar.

You should see the following:

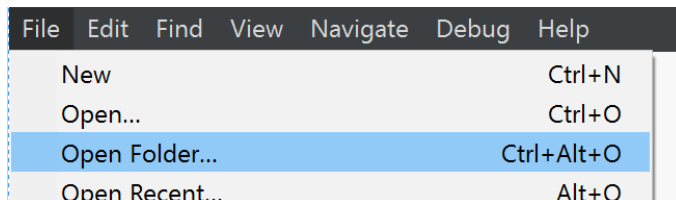


Now, this is all configuration information from Xampp, what we should do is clean out the folder htdocs, to put our own files. To do this, we will use brackets to clean out the folder.

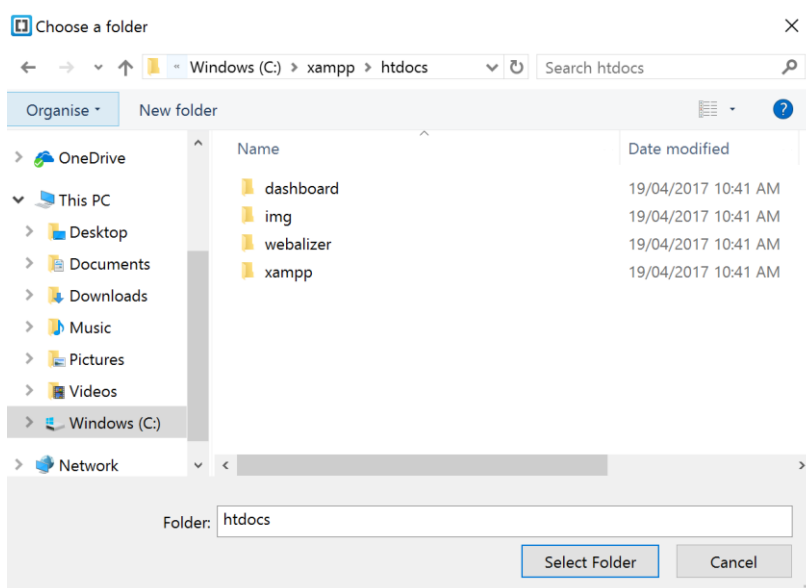
Load up brackets



From here, go to open folder

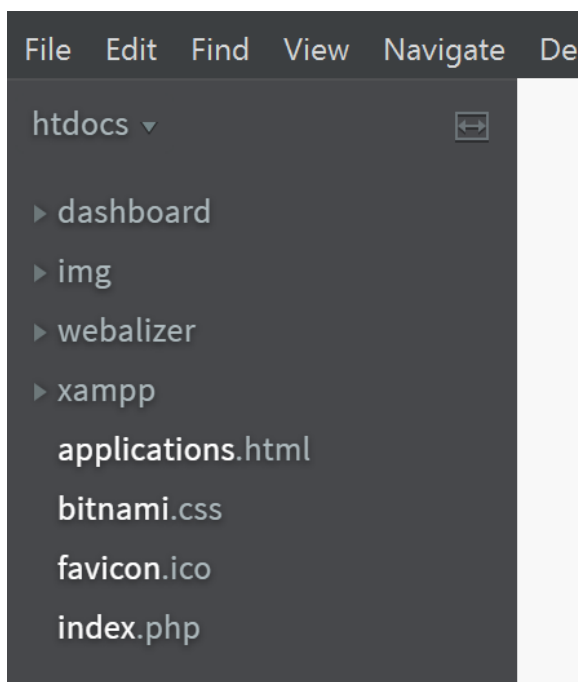


Navigate to c:\xampp\htdocs

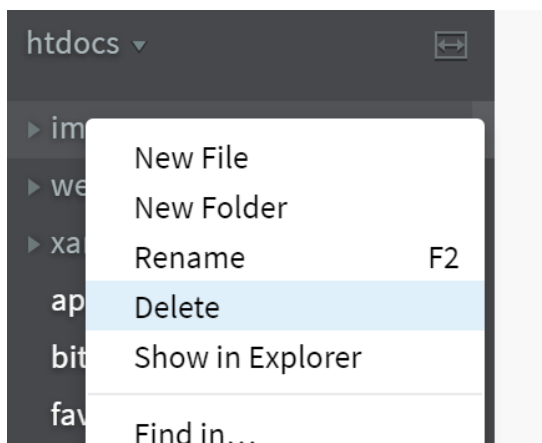


Click Select

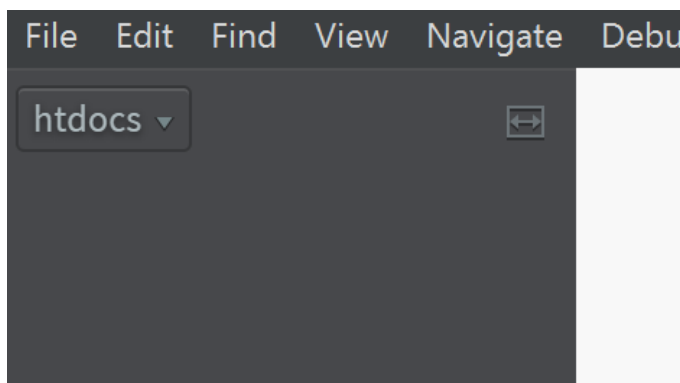
This should give you the following set of files in the navigation pane of brackets:



Highlight and right click on the navigation elements and delete everything in the folder

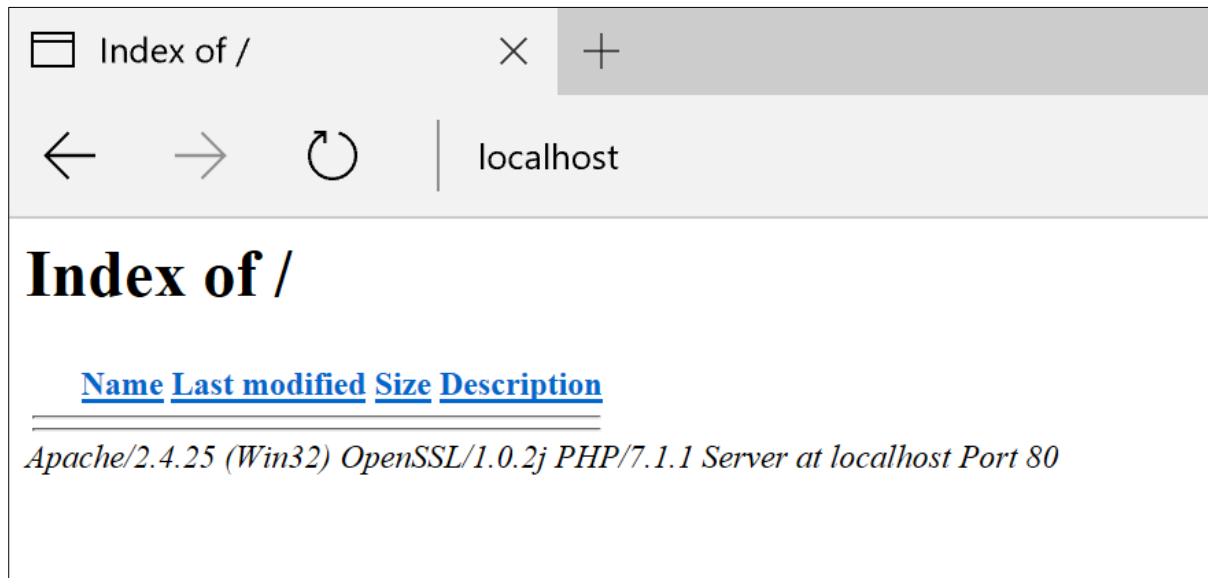


Once you have emptied the folder it should look like





Now, we check to ensure that the system is clean, go to your browser and navigate to <http://localhost/> You should see the following



Now we have a clean server to start creating with.

## Creating your php Website

From here, right click and create a new file called index.php and put in the following content:

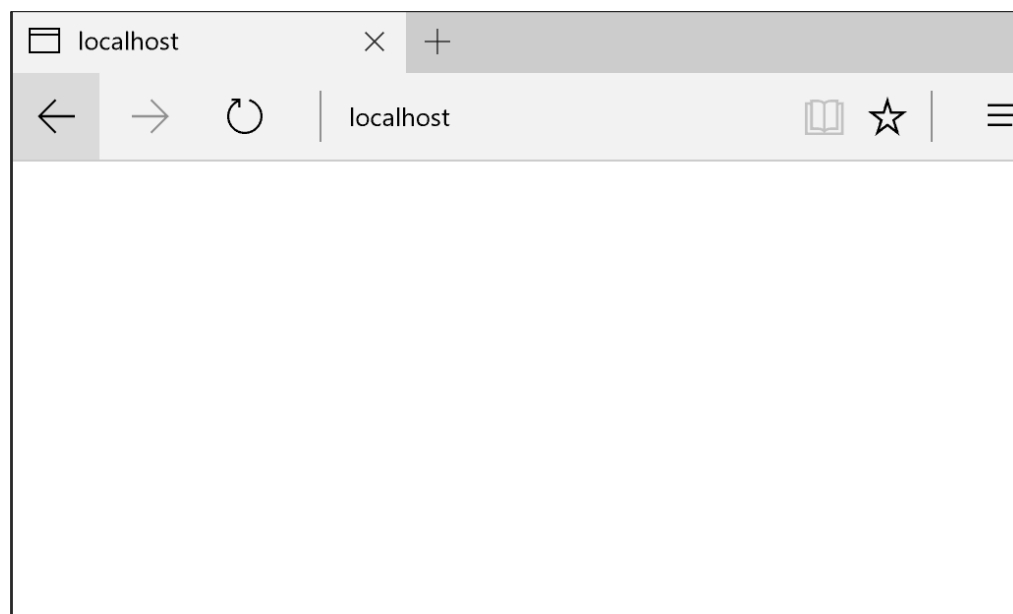
A screenshot of a code editor interface. The top menu bar includes 'File', 'Edit', 'Find', 'View', 'Navigate', 'Debug', and 'Help'. On the left, a sidebar shows 'Working Files' with 'index.php' selected, and a file explorer below it showing 'htdocs' containing 'index.php'. The main editor area displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style type="text/css">
5     </style>
6   </head>
7   <body>
8   </body>
9 </html>|
```

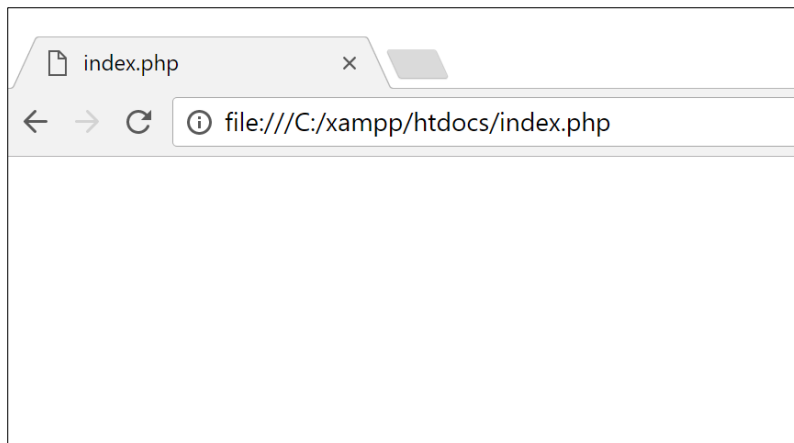
As you can see, our php file looks exactly like a html file. Basically, a php file is a html file with server capabilities built in. The extension tells the web server that the php service needs to examine and do something extra before supplying the end user with the page.

Now let's test the page, we should see nothing on the page.

Test using the server; go to localhost (Using Edge)



Test by using a browser (Chrome with the file being dragged onto the browser).

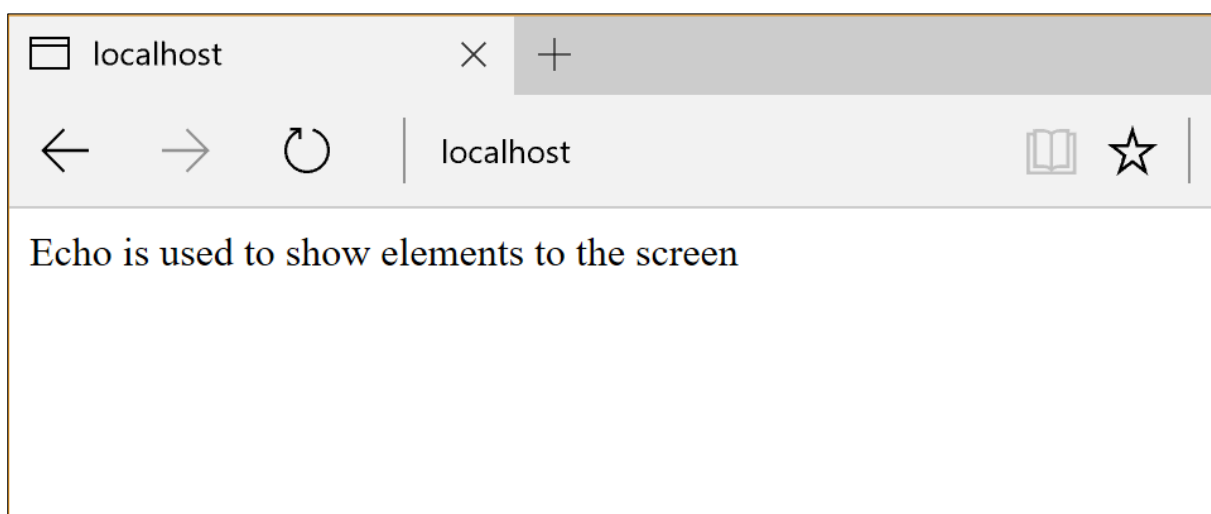


Now we will add some content to the page so you can see php in action.

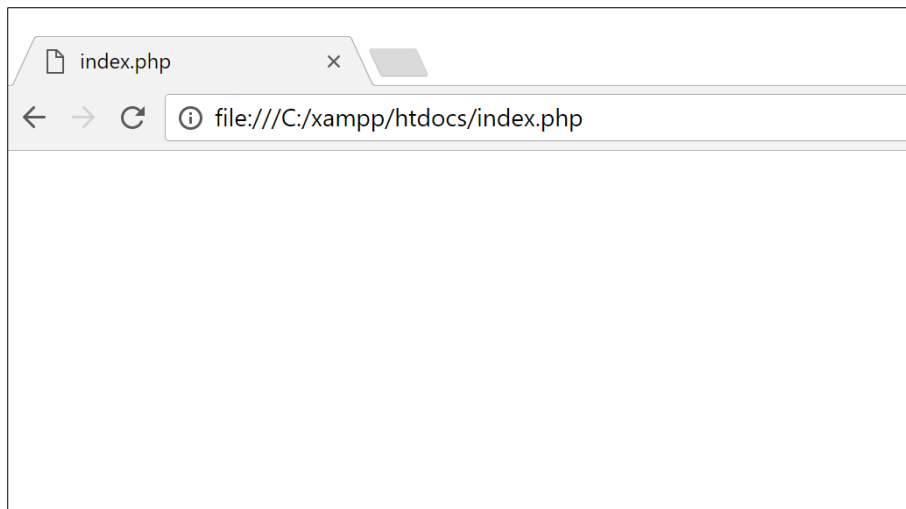
Brackets

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <style type="text/css">
5      </style>
6  </head>
7  <body>
8      <?php
9          echo 'Echo is used to show elements to the screen';
10     ?>
11 </body>|
12 </html>
```

Test in Edge (localhost)



Test in Chrome (dragged as a local file)



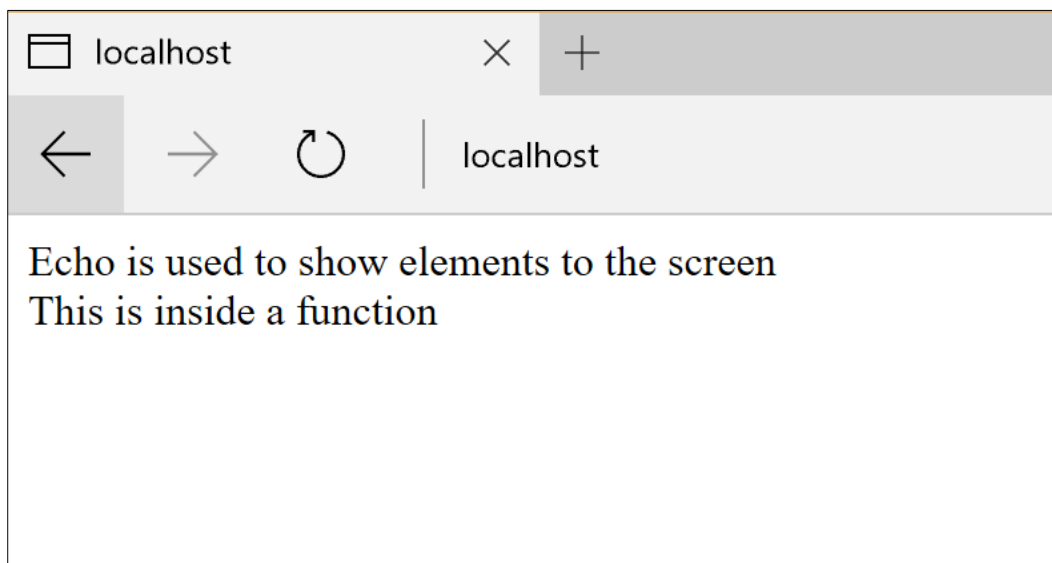
As can be seen, chrome will not display anything, whereas using localhost activates the php so the sentence can be shown in the browser.

Now we will add a function to php

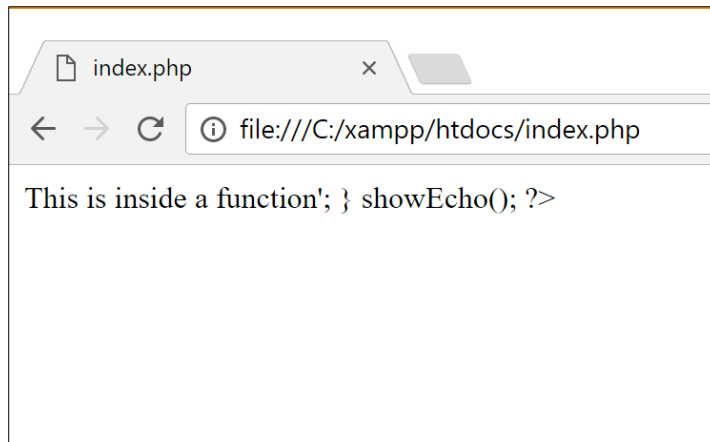
```
<body>
  <?php
    function showEcho()
    {
      echo 'Echo is used to show elements to the screen';
      echo '<br>This is inside a function';
    }

    showEcho();
  ?>
</body>
```

Test in Edge (localhost)



Test in Chrome (File dragged into browser)



As you can see, the file being used locally is no longer able to understand what is occurring, and henceforth just starts writing the php code to the screen. Therefore, a server must be used when dealing with php.

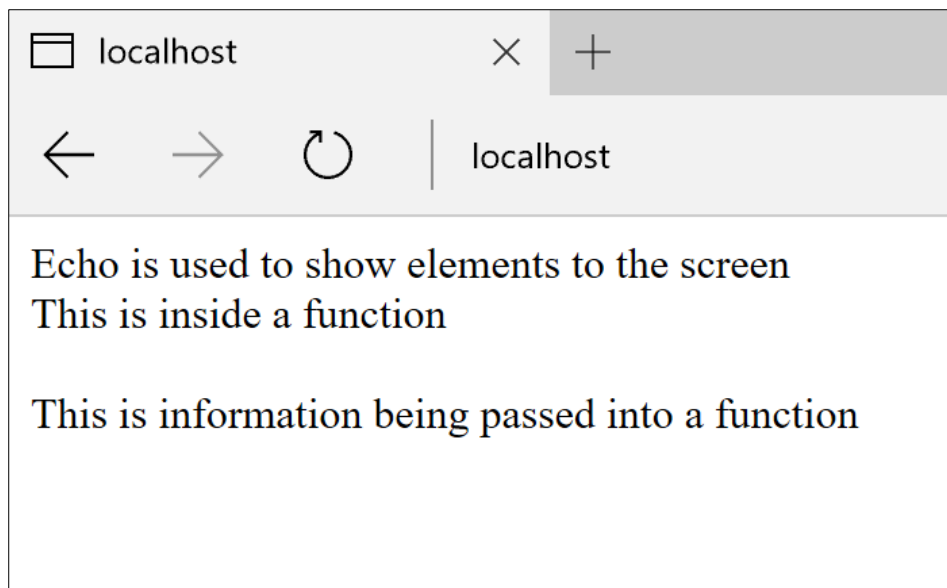
From here on out, we will deal with the localhost and server, so testing is to be done in a browser accessing localhost.

## PHP Variables

Now we can add some more code with variables

```
<body>
  <?php
    function showEcho($thisVariable)
    {
      echo 'Echo is used to show elements to the screen';
      echo '<br>This is inside a function';
      echo $thisVariable;
    }
    $thisVariable = "<br><br>This is information being passed into a function";
    showEcho($thisVariable);
  ?>
</body>
```

Now we test, and it should produce the following



Next, we will run some information through variables and start to look at how information can be transferred between pages.

Collecting information from the end user is normally done through the use of forms, so we will create a form on the page and follow the data.

Create the following pages

Index.php

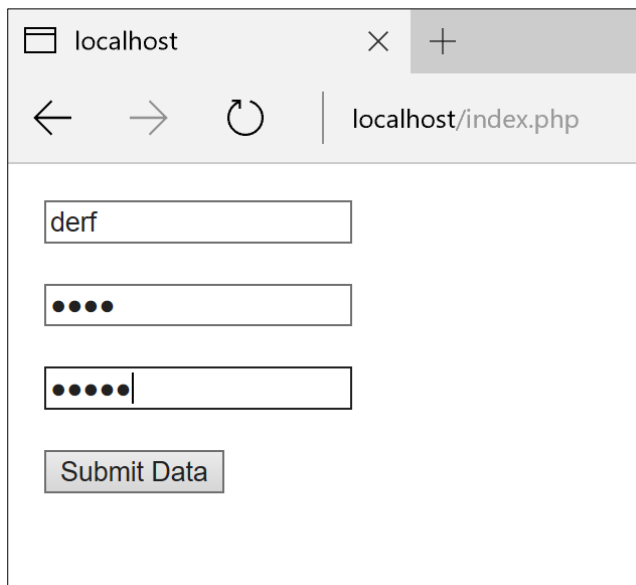
```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <style type="text/css">
5          input{margin: 10px 10px 10px 10px;}
6      </style>
7  </head>
8  <body>
9      <form action="collection.php" method="post">
10         <input type="text" name="username" placeholder="Enter Username"><br>
11         <input type="password" name="pwd" placeholder="Enter Password"><br>
12         <input type="password" name="pwd2" placeholder="Confirm Password"><br>
13         <input type="submit" value="Submit Data">
14     </form>
15 </body>
16 </html>
```

## Collection.php

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <style type="text/css">
5      </style>
6  </head>
7  <body>
8      <p><a href="index.php">Return Home</a></p>
9      <?php
10         $username = $_POST['username'];
11         $pwd = $_POST['pwd'];
12         $pwd2 = $_POST['pwd2'];
13
14         echo $username.'<br>'.$pwd.'<br>'.$pwd2;
15     ?>
16 </body>
17 </html>
```

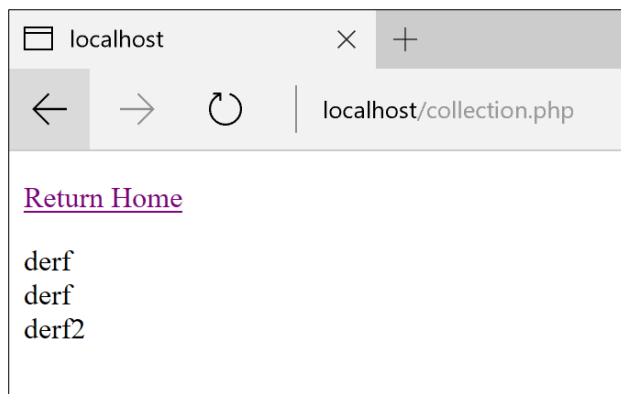
Test the page, using localhost

Live Index page



The screenshot shows a web browser window with the title 'localhost'. The address bar displays 'localhost/index.php'. The page content includes a text input field containing 'derf', a password input field with four dots, another password input field with five dots, and a 'Submit Data' button.

## Live Collection page



As you can see we have passed the information over from one page to another, there are multiple ways of transferring this data, in this case we used POST. We will quickly re-write the index page to use GET and you can see the result. Note, we are only changing index.php not collection.php, this is to show an issue that can occur if you don't make modifications all the way through a site.

## Index page

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <style type="text/css">
5     input{margin: 10px 10px 10px 10px;}
6 </style>
7 </head>
8 <body>
9 <form action="collection.php" method="get">
10 <input type="text" name="username" placeholder="Enter Username"><br>
11 <input type="password" name="pwd" placeholder="Enter Password"><br>
12 <input type="password" name="pwd2" placeholder="Confirm Password"><br>
13 <input type="submit" value="Submit Data">
14 </form>
15 </body>
16 </html>
```

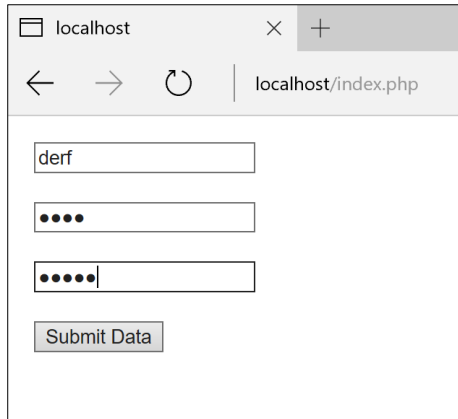
NB: Only change is method from post to get.

## Collection page

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <style type="text/css">
5 </style>
6 </head>
7 <body>
8 <p><a href="index.php">Return Home</a></p>
9 <?php
10     $username = $_POST['username'];
11     $pwd = $_POST['pwd'];
12     $pwd2 = $_POST['pwd2'];
13
14     echo $username. '<br>' . $pwd. '<br>' . $pwd2;
15 >
16 </body>
17 </html>
```



### Live index page



localhost

localhost/index.php

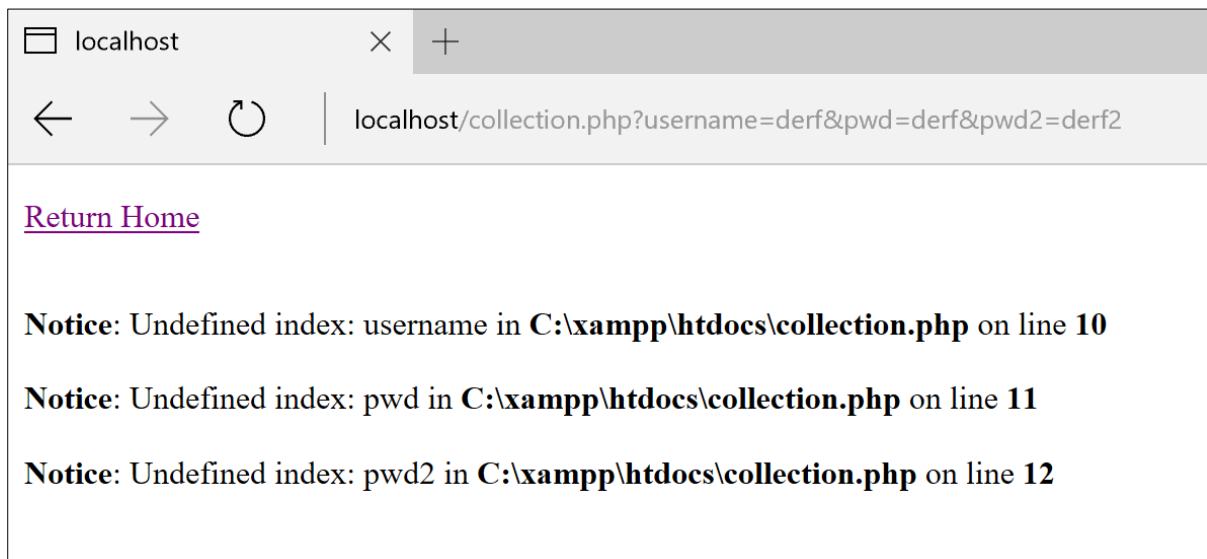
derf

....

.....

Submit Data

### Live collection page



localhost

localhost/collection.php?username=derf&pwd=derf&pwd2=derf2

[Return Home](#)

**Notice:** Undefined index: username in **C:\xampp\htdocs\collection.php** on line **10**

**Notice:** Undefined index: pwd in **C:\xampp\htdocs\collection.php** on line **11**

**Notice:** Undefined index: pwd2 in **C:\xampp\htdocs\collection.php** on line **12**

The error message that is appearing is informing you that the variables (username, pwd and pwd2) do not exist, this is because the variables are being created and having information put into them at the same time. Because this page doesn't have any POST data, the variables cannot be created. If we used \$\_GET instead of \$\_POST, the page would work fine.

Also, note that the get method has written the information into the address bar.

Post and Get are useful for transferring information from one page to another, but not quite so good for carrying that same information to a third or additional pages. To achieve this, we can use what is called Session variables.

We'll create a function that shows the memory that's being used, this function will be on the collection page. So we'll change the method back to post on the index page. See below to see what needs to be done.

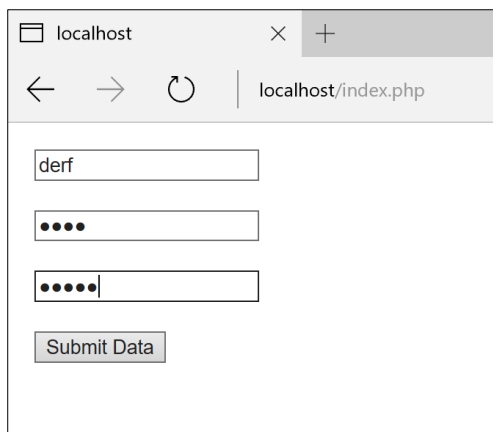
## Index.php (Change method from GET to POST)

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <style type="text/css">
5     input{margin: 10px 10px 10px 10px;}
6 </style>
7 </head>
8 <body>
9 <form action="collection.php" method="post">
10 <input type="text" name="username" placeholder="Enter Username"><br>
11 <input type="password" name="pwd" placeholder="Enter Password"><br>
12 <input type="password" name="pwd2" placeholder="Confirm Password"><br>
13 <input type="submit" value="Submit Data">
14 </form>
15 </body>
16 </html>
```

## Collection.php

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <style type="text/css">
5 </style>
6 <?php
7     function showMem()
8     {
9         echo '<pre>';
10        echo '<h3>Post Memory</h3>';
11        print_r($_POST);
12        echo '<h3>Get Memory</h3>';
13        print_r($_GET);
14        echo '<h3>Session Memory</h3>';
15        print_r($_SESSION);
16        echo '</pre>';
17    }
18 <?>
19 </head>
20 <body>
21 <p><a href="index.php">Return Home</a></p>
22 <?php
23     $username = $_POST['username'];
24     $pwd = $_POST['pwd'];
25     $pwd2 = $_POST['pwd2'];
26
27     echo $username.'<br>'.$pwd.'<br>'.$pwd2;
28 <?>
29 <?php showMem(); ?>
30 </body>
31 </html>
```

## Live index.php



localhost

localhost/index.php

derf

....

.....

Submit Data

## Live collection.php

[Return Home](#)

derf  
derf  
derf2

### Post Memory

```
Array
(
    [username] => derf
    [pwd] => derf
    [pwd2] => derf2
)
```

### Get Memory

```
Array
(
)
```

### Session Memory

**Notice:** Undefined variable: `_SESSION` in `C:\xampp\htdocs\collection.php` on line 15

As you can see, the information has been passed through. The new function is showing the current details in memory.

Now we will add sessions and an additional page.

## Index page

```
1  <?php
2      session_start();|
3  ?>
4  <!DOCTYPE html>
5  <html>
6  <head>
7      <style type="text/css">
8          input{margin: 10px 10px 10px 10px;}
9      </style>
10 </head>
11 <body>
12 <form action="collection.php" method="post">
13     <input type="text" name="username" placeholder="Enter Username"><br>
14     <input type="password" name="pwd" placeholder="Enter Password"><br>
15     <input type="password" name="pwd2" placeholder="Confirm Password"><br>
16     <input type="submit" value="Submit Data">
17 </form>
18 </body>
19 </html>
```

## Collection page

```
1  <?php
2      session_start();
3  ?>
4  <!DOCTYPE html>
5  <html>
6  <head>
7      <style type="text/css">
8      </style>
9      <?php
10         function showMem()
11         {
12             echo '<pre>';
13             echo '<h3>Post Memory</h3>';
14             print_r($_POST);
15             echo '<h3>Get Memory</h3>';
16             print_r($_GET);
17             echo '<h3>Session Memory</h3>';
18             print_r($_SESSION);
19             echo '</pre>';
20         }
21     ?>
22 </head>
23 <body>
24     <p><a href="index.php">Return Home</a></p>
25     <?php
26         $username = $_POST['username'];
27         $pwd = $_POST['pwd'];
28         $pwd2 = $_POST['pwd2'];|
29         echo $username.'<br>'.$pwd.'<br>'.$pwd2;
30     ?>
31     <?php showMem(); ?>
32     <p> <a href="check.php">Continue to next page</a></p>
33 </body>
34 </html>
```

## Check page

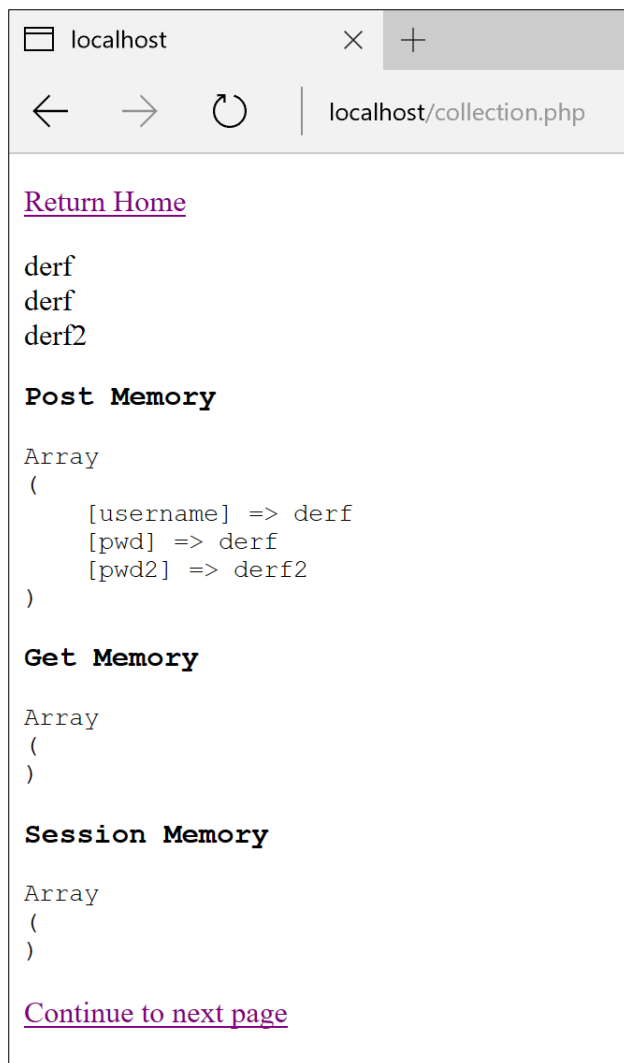
```
1  <?php
2      session_start();
3  ?>
4  <!DOCTYPE html>
5  <html>
6  <head>
7      <style type="text/css">
8      </style>
9      <?php
10         function showMem()
11         {
12             echo '<pre>';
13             echo '<h3>Post Memory</h3>';
14             print_r($_POST);
15             echo '<h3>Get Memory</h3>';
16             print_r($_GET);
17             echo '<h3>Session Memory</h3>';
18             print_r($_SESSION);
19             echo '</pre>';
20         }
21     ?>
22 </head>
23 <body>
24     <?php showMem(); ?>
25     <p><a href="index.php">Return to index page</a></p>
26 </body>
27 </html>
```

## Live index.php

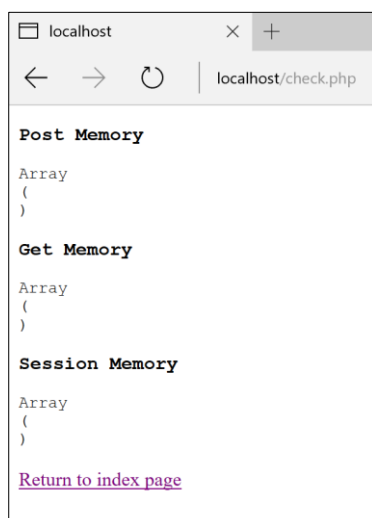
localhost × +

← → ↺ | localhost/index.php

## Live collection.php



## Live check.php



Notice how the third page, check.php, contains nothing in memory. What we will do is generate a session id on the index page and modify collection.php to store the username and password in sessions as well. This will showcase how persistent session memory is.

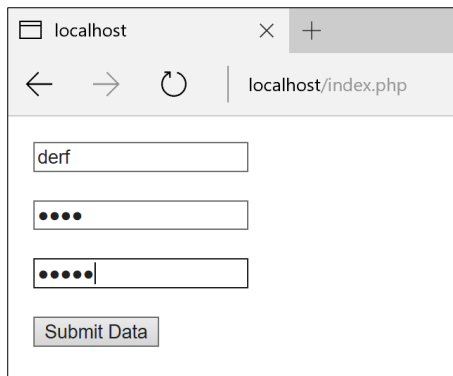
### Index.php

```
1  <?php
2      session_start();
3      $_SESSION['id'] = session_id();
4  ?>
5  <!DOCTYPE html>
6  <html>
7  <head>
8      <style type="text/css">
9          input{margin: 10px 10px 10px 10px;}
10     </style>
11 </head>
12 <body>
13     <form action="collection.php" method="post">
14         <input type="text" name="username" placeholder="Enter Username"><br>
15         <input type="password" name="pwd" placeholder="Enter Password"><br>
16         <input type="password" name="pwd2" placeholder="Confirm Password"><br>
17         <input type="submit" value="Submit Data">
18     </form>
19 </body>
20 </html>
```

### Collection.php

```
1  <?php
2      session_start();
3  ?>
4  <!DOCTYPE html>
5  <html>
6  <head>
7      <style type="text/css">
8      </style>
9      <?php
10         function showMem()
11         {
12             echo '<pre>';
13             echo '<h3>Post Memory</h3>';
14             print_r($_POST);
15             echo '<h3>Get Memory</h3>';
16             print_r($_GET);
17             echo '<h3>Session Memory</h3>';
18             print_r($_SESSION);
19             echo '</pre>';
20         }
21     ?>
22 </head>
23 <body>
24     <p><a href="index.php">Return Home</a></p>
25     <?php
26         $username = $_POST['username'];
27         $pwd = $_POST['pwd'];
28         $pwd2 = $_POST['pwd2'];
29         echo $username.'<br>'.$pwd.'<br>'.$pwd2;
30         $_SESSION['username'] = $username;
31         $_SESSION['password'] = $pwd;
32     ?>
33     <?php showMem(); ?>
34     <p><a href="check.php">Continue to next page</a></p>
35 </body>
36 </html>
```

## Live index.php



localhost × +

← → ↻ | localhost/index.php

derf

....

....|

Submit Data

## Live collection.php



localhost × +

← → ↻ | localhost/collection.php

[Return Home](#)

derf  
derf  
derf2

**Post Memory**

```
Array
(
    [username] => derf
    [pwd] => derf
    [pwd2] => derf2
)
```

**Get Memory**

```
Array
(
)
```

**Session Memory**

```
Array
(
    [id] => q0bp480rt4rgj2vg94ervrfuv4
    [username] => derf
    [password] => derf
)
```

[Continue to next page](#)



Live check.php



As you can see, the POST data doesn't survive to the third page, whereas the session information is still accessible.

Notice we have duplicated the function `showMem()`, this is bad coding form as we are wasting memory. To tidy this up, we will create a new page called `functions.php` in which we will store the function `showMem()` and link to the two pages. Think of this like creating an external stylesheet, one page referenced by multiple pages to locate a working piece of code.

So, let's create the following structure and pages.

Index.php

- Same as before, no need to change

## Collection.php

```
1  <?php
2      session_start();
3      include_once('functions.php');
4  ?>
5  <!DOCTYPE html>
6  <html>
7      <head>
8          <style type="text/css">
9          </style>
10     </head>
11     <body>
12         <p><a href="index.php">Return Home</a></p>
13         <?php
14             $username = $_POST['username'];
15             $pwd = $_POST['pwd'];
16             $pwd2 = $_POST['pwd2'];
17             echo $username.'<br>'.$pwd.'<br>'.$pwd2;
18             $_SESSION['username'] = $username;
19             $_SESSION['password'] = $pwd;
20         ?>
21         <?php showMem(); ?>
22         <p> <a href="check.php">Continue to next page</a></p>
23     </body>
24 </html>
```

## Check.php

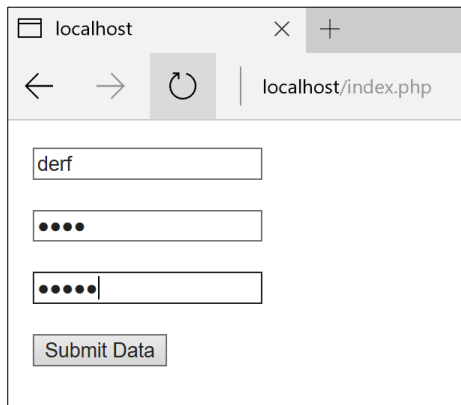
```
1  <?php
2      session_start();
3      include_once('functions.php');
4  ?>
5  <!DOCTYPE html>
6  <html>
7      <head>
8          <style type="text/css">
9          </style>
10     </head>
11     <body>
12         <?php showMem(); ?>
13         <p><a href="index.php">Return to index page</a></p>
14     </body>
15 </html>
```

## Functions.php

```
1  <?php
2
3  function showMem()
4  ▼ {
5      echo '<pre>';
6      echo '<h3>Post Memory</h3>';
7      print_r($_POST);
8      echo '<h3>Get Memory</h3>';
9      print_r($_GET);
10     echo '<h3>Session Memory</h3>';
11     print_r($_SESSION);
12     echo '</pre>';
13 }
14 |
15 ?>
```

## Test the pages

### Live Index.php



localhost × +

← → ↻ | localhost/index.php

derf

....

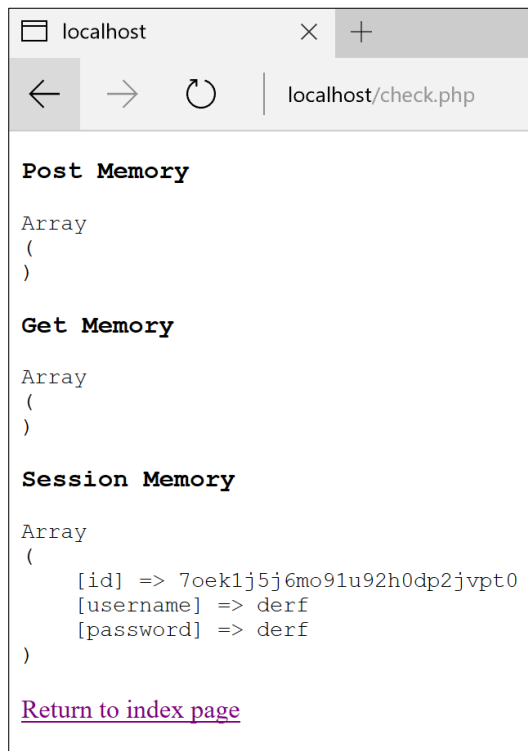
.....

Submit Data

## Live Collection.php



## Live Check.php



If you notice, the session id is different, this is because during writing of this, I closed the browser, each time the browser is opened and goes to localhost, a new session id is generated.

Notice how cleaner the code is when the functions are stored separately.

So, now that we can transfer information from one page to another and have data stay persistent throughout the site, how do we use it? Normally we use this information to allow the end user to make choices as to what they see. As you have tested the page you will have seen that the data being supplied for password has been different, what we will do now, is to make information appear to the end user based on that information. So if the passwords match then we will congratulate them, if they do not match, we will tell them to try again.

We are not using check.php so that can stay the same.

Make the following changes

Index.php

- Same as before, no need to change

Collection.php

```
1  <?php
2      session_start();
3      include_once('functions.php');
4  ?>
5  <!DOCTYPE html>
6  <html>
7  <head>
8      <style type="text/css">
9      </style>
10 </head>
11 <body>
12     <p><a href="index.php">Return Home</a></p>
13     <?php
14         $username = $_POST['username'];
15         $pwd = $_POST['pwd'];
16         $pwd2 = $_POST['pwd2'];
17         $result = checkPasswords($pwd,$pwd2);
18         if ($result)
19         {
20             echo 'Entered details: '.$username.'  
'.$pwd.'  
'.$pwd2;
21             $_SESSION['username'] = $username;
22             $_SESSION['password'] = $pwd;
23             echo '<br><br>This is where you could put a profile page<br>';
24         } else
25         {
26             echo 'Your passwords didn\'t match, please return to the previous page and try again';
27             echo '<br><a href="index.php">Re-enter Details</a>';
28         }
29     ?>
30     <?php showMem(); ?>
31     <p> <a href="check.php">Continue to next page</a></p>
32 </body>
33 </html>
```

## Functions.php

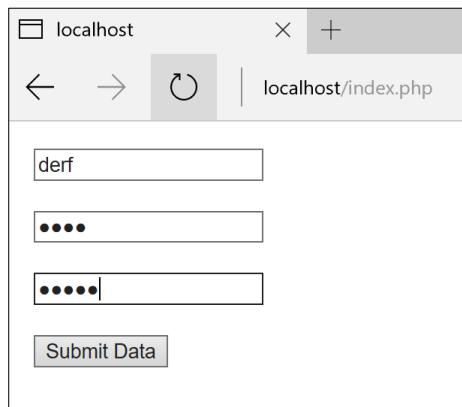
```
1  <?php
2
3  function showMem()
4  ▼ {
5      echo '<pre>';
6      echo '<h3>Post Memory</h3>';
7      print_r($_POST);
8      echo '<h3>Get Memory</h3>';
9      print_r($_GET);
10     echo '<h3>Session Memory</h3>';
11     print_r($_SESSION);
12     echo '</pre>';
13 }
14
15 function checkPasswords($p1,$p2)
16 ▼ {
17     if($p1 == $p2)
18         return true;
19     else
20         return false;|
21
22 }
23 ?>
```

## Check.php

- Same as before, no need to change

## Live index.php

Enter Username:derf; password:derf; password2: derf or derf2



localhost

localhost/index.php

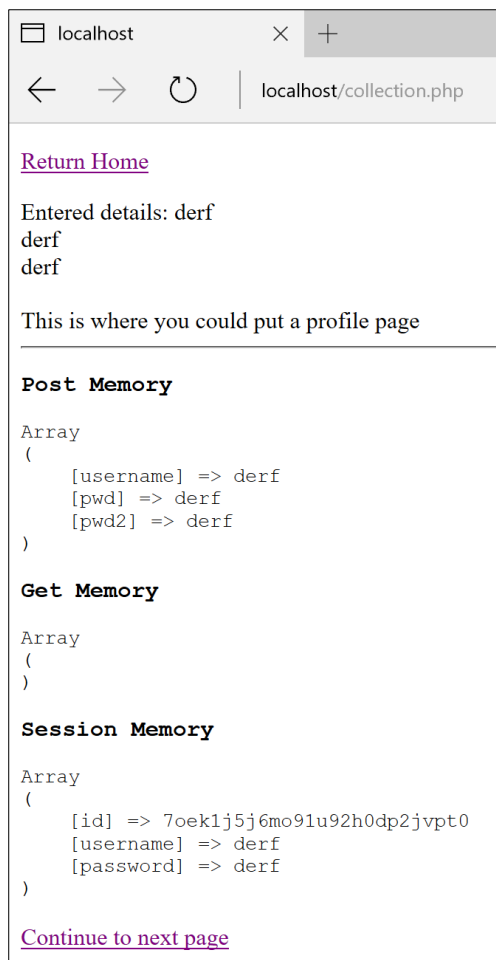
derf

.....

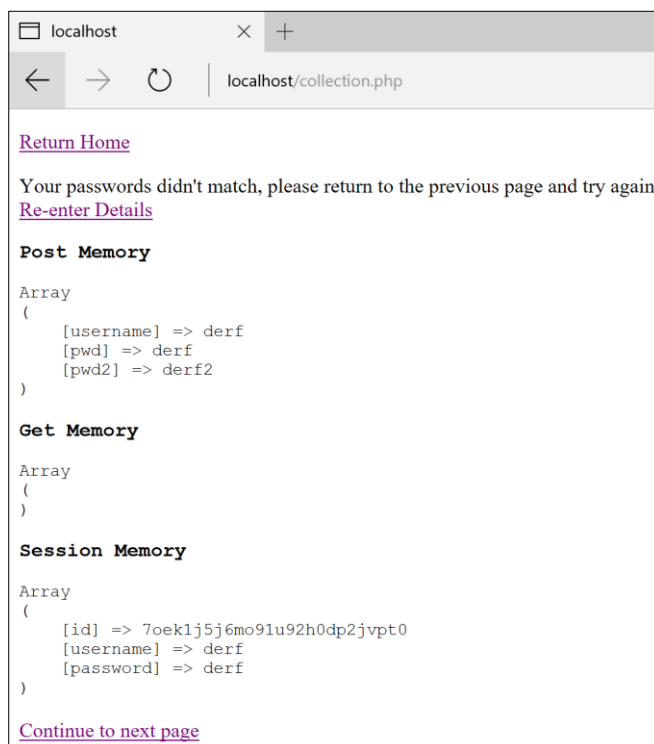
.....|

Submit Data

## Live collection.php (both passwords match)



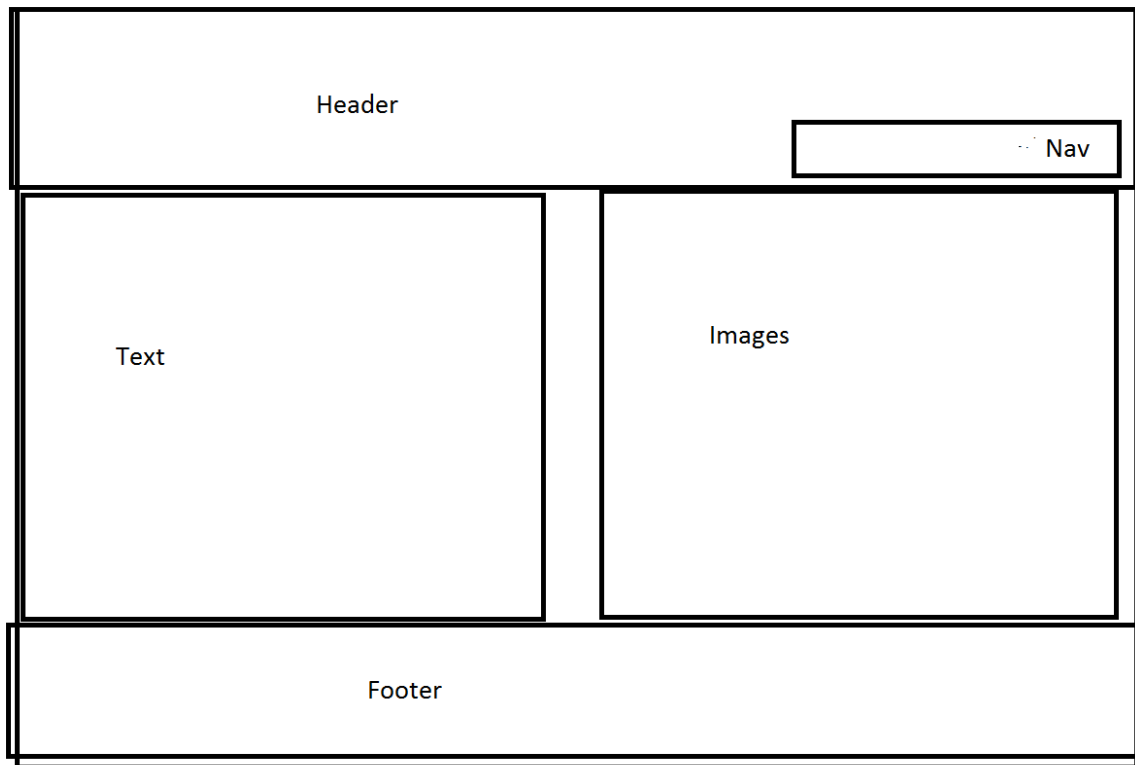
## Live collection.php (passwords do not match)



## PHP as a templating Language

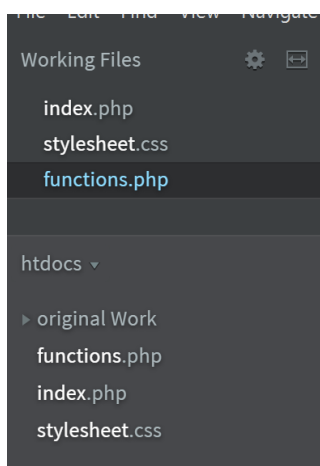
PHP has the ability to allow you to create templates for a website, we will now create and template some new pages.

The design will be



To start with let's move all of the current files into a folder so we can keep them for later. Create the following files:

- Index.php
- Stylesheet.css
- Functions.php





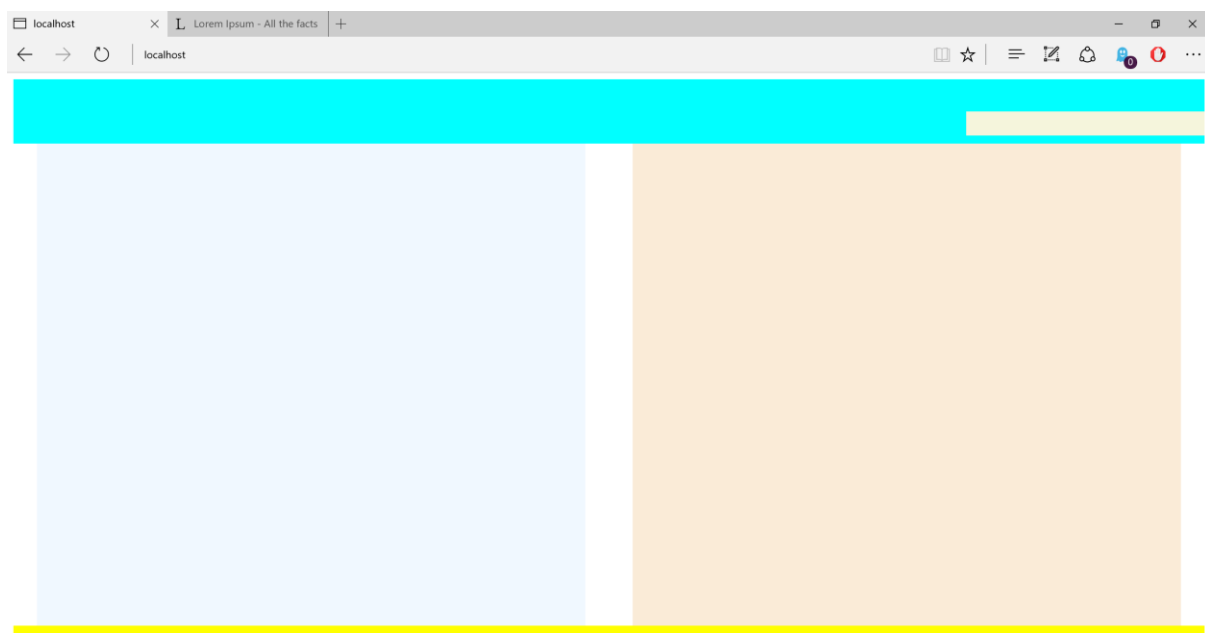
Now that we have the basic file structure, go online, grab a couple of pictures (your choice, at least 2 and prep some lorim ipsum for the text side of the page ) let's code up the design.

Stage 1 – Everything is built on the one page

Index.php

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <style type="text/css">
5          #header{width: 100%; height: 80px; background-color: aqua;}
6          #main{width: 100%; min-height: 600px; background-color: white;}
7          #footer{width: 100%; height: 20px; background-color: yellow;font-size: .8em; text-align:
8              right;}
9          #left{width: 46%; min-height: inherit; margin-left: 2%; margin-right: 2%; float: left;
10             background-color: aliceblue;}
11         #right{width: 46%; min-height:inherit; margin-left: 2%; margin-right: 2%; float: right;
12             background-color: antiquewhite;}
13         #nav{min-width: 20%; width: 20%; height: 30px; float: right; text-align: left; margin-top:
14             40px; background-color: beige;}
15     </style>
16 </head>
17 <body>
18     <div id="header">
19         <div id="nav"></div><!-- eo nav -->
20     </div><!-- eo header -->
21     <div id="main">
22         <div id="left"></div><!-- eo left -->
23         <div id="right"></div><!-- eo right -->
24     </div><!-- eo main -->
25     <div id="footer"></div><!-- eo footer -->
26 </body>
27 </html>
```

This will give you:



Now that we have the general layout and design of the page completed, it's time to move elements to their separate sections, ie styles move into the stylesheet.css and are linked up that way; background colours are removed and content gets put into place.

So, let's move the styles and remove the background colours, 2 stages.

Stage 1

## Index.php

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link href="stylesheet.css" type="text/css" rel="stylesheet">
5   </head>
6   <body>
7     <div id="header">
8       <div id="nav"></div><!-- eo nav -->
9     </div><!-- eo header -->
10    <div id="main">
11      <div id="left"></div><!-- eo left -->
12      <div id="right"></div><!-- eo right -->
13    </div><!-- eo main -->
14    <div id="footer"></div><!-- eo footer -->
15  </body>
16 </html>
```

## Stylesheet.css

```
1 #header{width: 100%; height: 80px; background-color: aqua;}
2 #main{width: 100%; min-height: 600px; background-color: white;}
3 #footer{width: 100%; height: 20px; background-color: yellow;font-size: .8em; text-align: right;}
4 #left{width: 46%; min-height: inherit; margin-left: 2%; margin-right: 2%; float: left; background-
  color: aliceblue;}
5 #right{width: 46%; min-height:inherit; margin-left: 2%; margin-right: 2%; float: right; background-
  color: antiquewhite;}
6 #nav{min-width: 20%; width: 20%; height: 30px; float: right; text-align: left; margin-top: 40px;
  background-color: beige;}
```

## Result



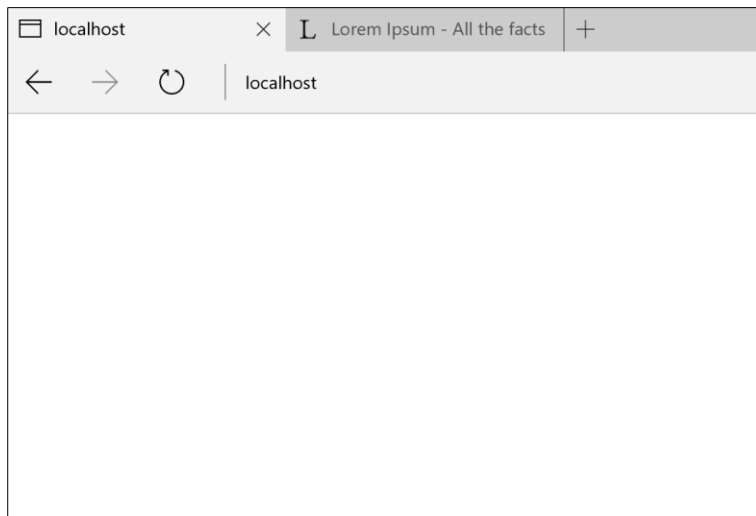
## Stage 2 – removing of background colours

### Stylesheet.css

```
1 #header{width: 100%; height: 80px;}
2 #main{width: 100%; min-height: 600px;}
3 #footer{width: 100%; height: 20px;font-size: .8em; text-align: right;}
4 #left{width: 46%; min-height: inherit; margin-left: 2%; margin-right: 2%; float: left;}
5 #right{width: 46%; min-height:inherit; margin-left: 2%; margin-right: 2%; float: right;}
6 #nav{min-width: 40%; height: 30px; float: right; text-align: left; margin-top: 40px;}
7
```

NB: nav has increased width and removed the width command, yet kept the min-width command in addition to removing the colours.

### Result



Now, we add content to the various locations of the page, with additional styles as needed.

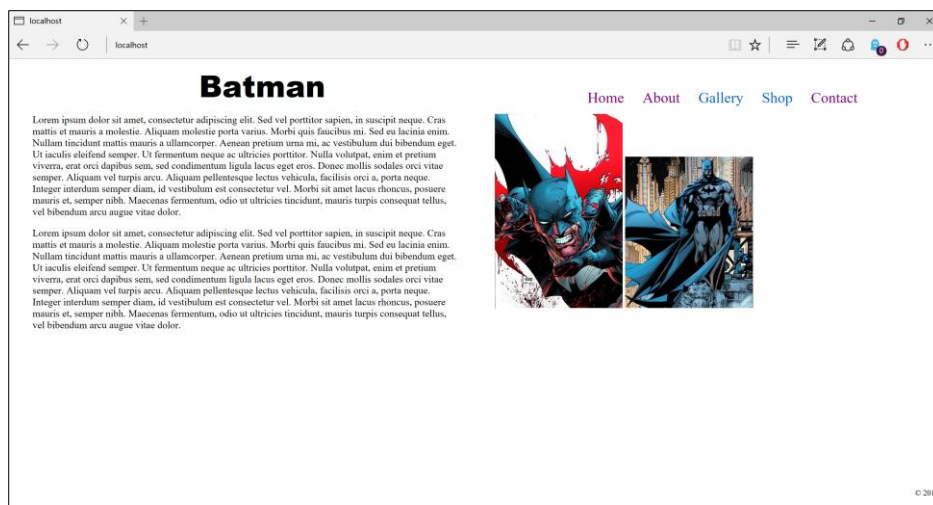
### Index.php

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <link href="stylesheet.css" type="text/css" rel="stylesheet">
5      <style type="text/css">
6          .headerText{font-size:3em; font-weight: 800; font-family:sans-serif; margin-left: 20%;}
7          .imgSize{width: 30%;}
8          #nav a{float: left; margin-left: 5%; font-size: 1.5em; text-decoration: none;}
9          #nav a:hover{text-decoration: underline;}
10     </style>
11 </head>
12 <body>
13     <div id="header"><span class="headerText">Batman</span>
14     <div id="nav">
15         <a href="index.php">Home</a>
16         <a href="pages/about.php">About</a>
17         <a href="pages/gallery.php">Gallery</a>
18         <a href="pages/shop.php">Shop</a>
19         <a href="pages/contact.php">Contact</a>
20     </div><!-- eo nav -->
21 </div><!-- eo header -->
22 <div id="main">
23     <div id="left">
24         Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vel porttitor sapien, in
        suscipit neque. Cras mattis et mauris a molestie. Aliquam molestie porta varius. Morbi
        quis faucibus mi. Sed eu lacinia enim. Nullam tincidunt mattis mauris a ullamcorper.
        Aenean pretium urna mi, ac vestibulum dui bibendum eget. Ut iaculis eleifend semper. Ut
        fermentum neque ac ultricies porttitor. Nulla volutpat, enim et pretium viverra, erat orci
        dapibus sem, sed condimentum ligula lacus eget eros. Donec mollis sodales orci vitae
        semper. Aliquam vel turpis arcu. Aliquam pellentesque lectus vehicula, facilisis orci a,
        porta neque. Integer interdum semper diam, id vestibulum est consectetur vel. Morbi sit
        amet lacus rhoncus, posuere mauris et, semper nibh. Maecenas fermentum, odio ut ultricies
        tincidunt, mauris turpis consequat tellus, vel bibendum arcu augue vitae dolor.
25
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vel porttitor sapien, in
        suscipit neque. Cras mattis et mauris a molestie. Aliquam molestie porta varius. Morbi
        quis faucibus mi. Sed eu lacinia enim. Nullam tincidunt mattis mauris a ullamcorper.
        Aenean pretium urna mi, ac vestibulum dui bibendum eget. Ut iaculis eleifend semper. Ut
        fermentum neque ac ultricies porttitor. Nulla volutpat, enim et pretium viverra, erat orci
        dapibus sem, sed condimentum ligula lacus eget eros. Donec mollis sodales orci vitae
        semper. Aliquam vel turpis arcu. Aliquam pellentesque lectus vehicula, facilisis orci a,
        porta neque. Integer interdum semper diam, id vestibulum est consectetur vel. Morbi sit
        amet lacus rhoncus, posuere mauris et, semper nibh. Maecenas fermentum, odio ut ultricies
        tincidunt, mauris turpis consequat tellus, vel bibendum arcu augue vitae dolor. </p>
26     </div><!-- eo left -->
27     <div id="right">
28         
29         
30     </div><!-- eo right -->
31 </div><!-- eo main -->
32 <div id="footer">&copy; 2017</div><!-- eo footer -->
33 </body>
34 </html>

```

## Result



Notice that the navigation links point to pages that don't exist and are inside a folder. The images are linked to images that were stored inside an images folder.

So far, there is nothing that has required us to use php. Now we will start to turn aspects of this into a simple template that we can use on the additional pages.

The first change will be pulling out the header. And storing it as php functions; this is a dual function as we will be able to get the pages to dynamically write the navigation. We will also move the additional styles into the stylesheet.css.

## Index.php

```
1  <?php
2      include_once('functions.php');
3  ?>
4  <!DOCTYPE html>
5  <html>
6      <head>
7          <link href="stylesheet.css" type="text/css" rel="stylesheet">
8          <style type="text/css">|
9          </style>
10     </head>
11     <body>
12         <?php
13             headerLayout('index');
14         ?>
15         <div id="main">
16             <div id="left">
17                 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vel porttitor sapien, in
                suscipit neque. Cras mattis et mauris a molestie. Aliquam molestie porta varius. Morbi
                quis faucibus mi. Sed eu lacinia enim. Nullam tincidunt mattis mauris a ullamcorper.
                Aenean pretium urna mi, ac vestibulum dui bibendum eget. Ut iaculis eleifend semper. Ut
                fermentum neque ac ultricies porttitor. Nulla volutpat, enim et pretium viverra, erat orci
                dapibus sem, sed condimentum ligula lacus eget eros. Donec mollis sodales orci vitae
                semper. Aliquam vel turpis arcu. Aliquam pellentesque lectus vehicula, facilisis orci a,
                porta neque. Integer interdum semper diam, id vestibulum est consectetur vel. Morbi sit
                amet lacus rhoncus, posuere mauris et, semper nibh. Maecenas fermentum, odio ut ultricies
                tincidunt, mauris turpis consequat tellus, vel bibendum arcu augue vitae dolor.
18             <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vel porttitor sapien, in
                suscipit neque. Cras mattis et mauris a molestie. Aliquam molestie porta varius. Morbi
                quis faucibus mi. Sed eu lacinia enim. Nullam tincidunt mattis mauris a ullamcorper.
                Aenean pretium urna mi, ac vestibulum dui bibendum eget. Ut iaculis eleifend semper. Ut
                fermentum neque ac ultricies porttitor. Nulla volutpat, enim et pretium viverra, erat orci
                dapibus sem, sed condimentum ligula lacus eget eros. Donec mollis sodales orci vitae
                semper. Aliquam vel turpis arcu. Aliquam pellentesque lectus vehicula, facilisis orci a,
                porta neque. Integer interdum semper diam, id vestibulum est consectetur vel. Morbi sit
                amet lacus rhoncus, posuere mauris et, semper nibh. Maecenas fermentum, odio ut ultricies
                tincidunt, mauris turpis consequat tellus, vel bibendum arcu augue vitae dolor. </p>
19             </div><!-- eo left -->
20             <div id="right">
21                 
22                 
23             </div><!-- eo right -->
24         </div><!-- eo main -->
25         <div id="footer">&copy; 2017</div><!-- eo footer -->
26     </body>
27 </html>
```

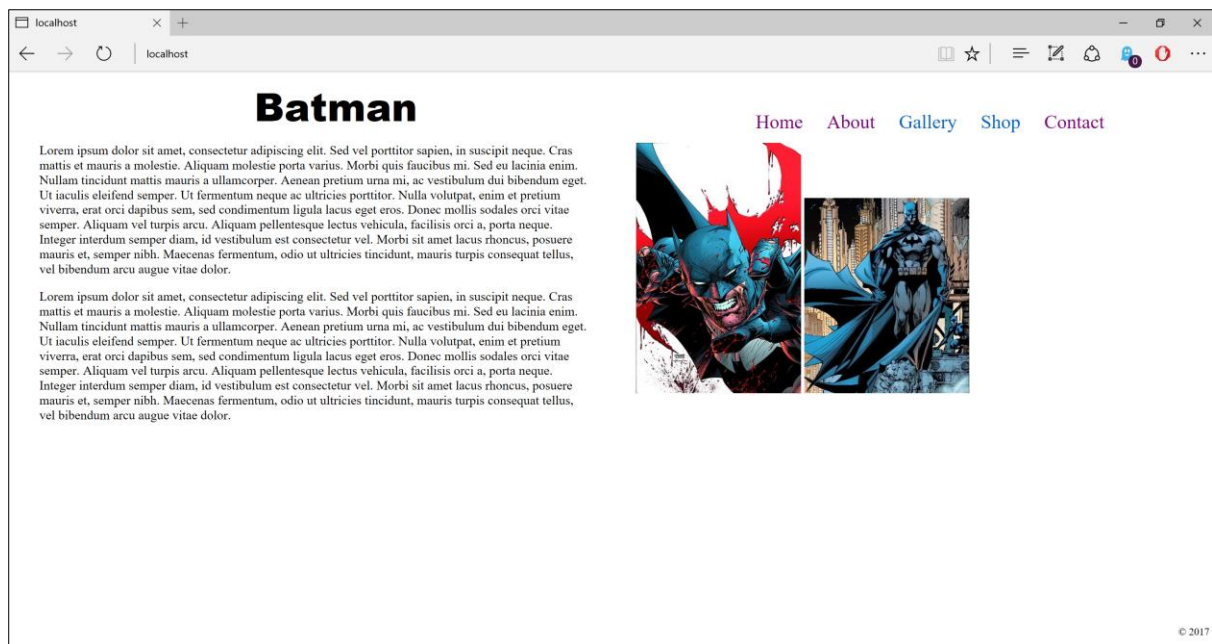
## Stylesheet.css

```
1  /* Overall layout*/
2  #header{width: 100%; height: 80px;}
3  #main{width: 100%; min-height: 600px;}
4  #footer{width: 100%; height: 20px;font-size: .8em; text-align: right;}
5  #left{width: 46%; min-height: inherit; margin-left: 2%; margin-right: 2%; float: left;}
6  #right{width: 46%; min-height:inherit; margin-left: 2%; margin-right: 2%; float: right;}
7  #nav{min-width: 40%; height: 30px; float: right; text-align: left; margin-top: 40px;}
8
9  /* Styles for specific aspects of the page */
10 .headerText{font-size:3em; font-weight: 800; font-family:sans-serif; margin-left: 20%;}
11 .imgSize{width: 30%;}
12 #nav a{float: left; margin-left: 5%; font-size: 1.5em; text-decoration: none;}
13 #nav a:hover{text-decoration: underline;}
14 |
```

## Functions.php

```
1  <?php
2
3  function headerLayout($page)
4  ▼ {
5      echo ' <div id="header"><span class="headerText">Batman</span>';
6      navigation($page);
7      echo '</div><!-- eo header -->';
8  }
9
10 function navigation($page)
11 ▼ {
12     if($page == 'index')
13     ▼ {
14         echo'<div id="nav">
15             <a href="index.php">Home</a>
16             <a href="pages/about.php">About</a>
17             <a href="pages/gallery.php">Gallery</a>
18             <a href="pages/shop.php">Shop</a>
19             <a href="pages/contact.php">Contact</a>
20         </div><!-- eo nav -->';
21     }
22     else
23     ▼ {
24         echo'<div id="nav">
25             <a href="../index.php">Home</a>
26             <a href="about.php">About</a>
27             <a href="gallery.php">Gallery</a>
28             <a href="shop.php">Shop</a>
29             <a href="contact.php">Contact</a>
30         </div><!-- eo nav -->';
31     }
32 }
33 |
34 ?>
```

## Result



Always remember to test the pages after each modification, this will allow for debugging to occur easier and quicker.

Next we modify the index page footer code, we extract that to the functions page and replace the code.

Functions.php

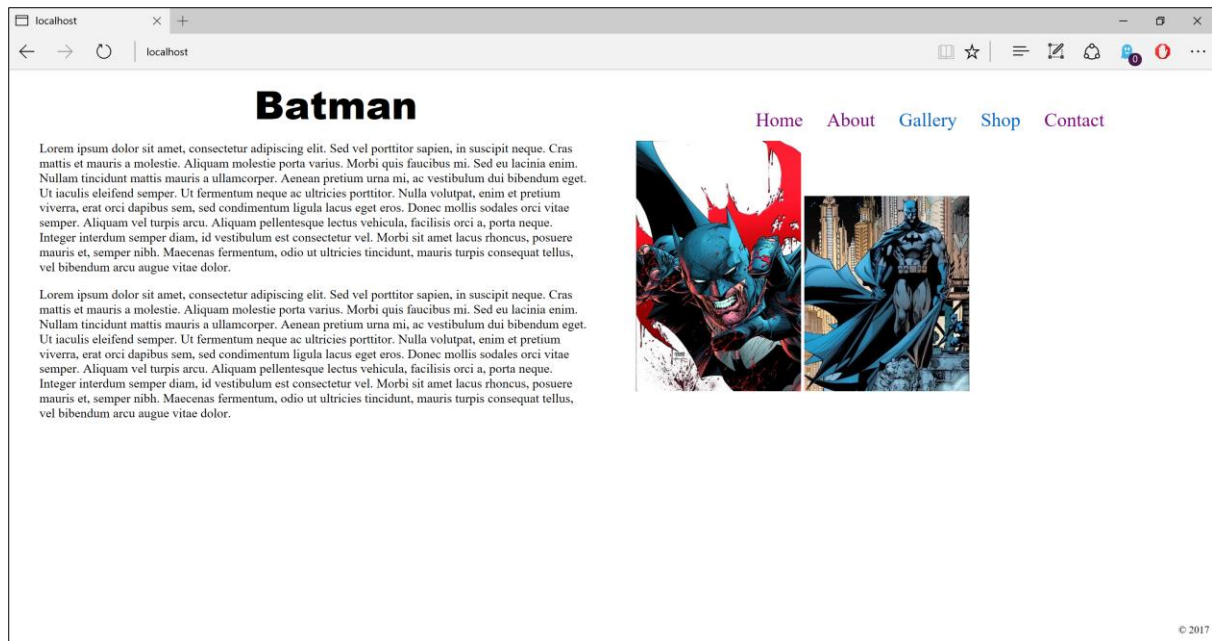
```
34 function footerLayout()  
35 {  
36     echo '<div id="footer">&copy; 2017</div><!-- eo footer -->';  
37 }  
38 ?>
```

Index.php

```
23     </div><!-- eo right -->  
24 </div><!-- eo main -->  
25 <?php  
26     footerLayout();  
27 ?>  
28 </body>  
29 </html>
```

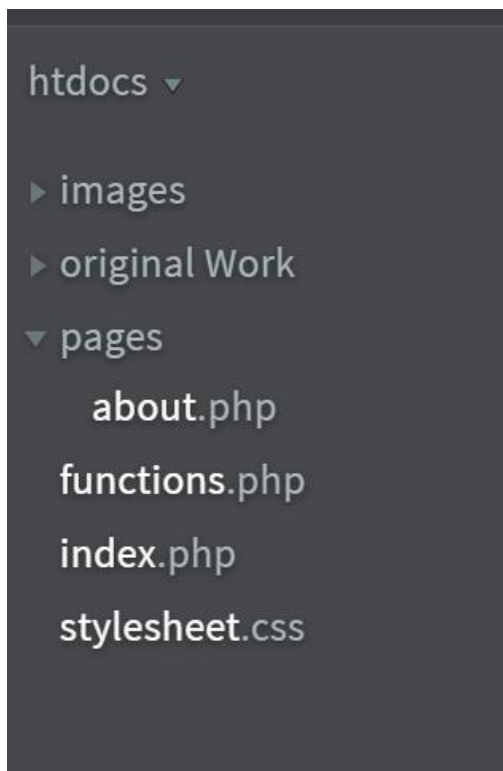
## Result





Now that this has all been done, we've got the basis for our site template. So, create the folder pages and then create about.php in that folder.

Folder layout



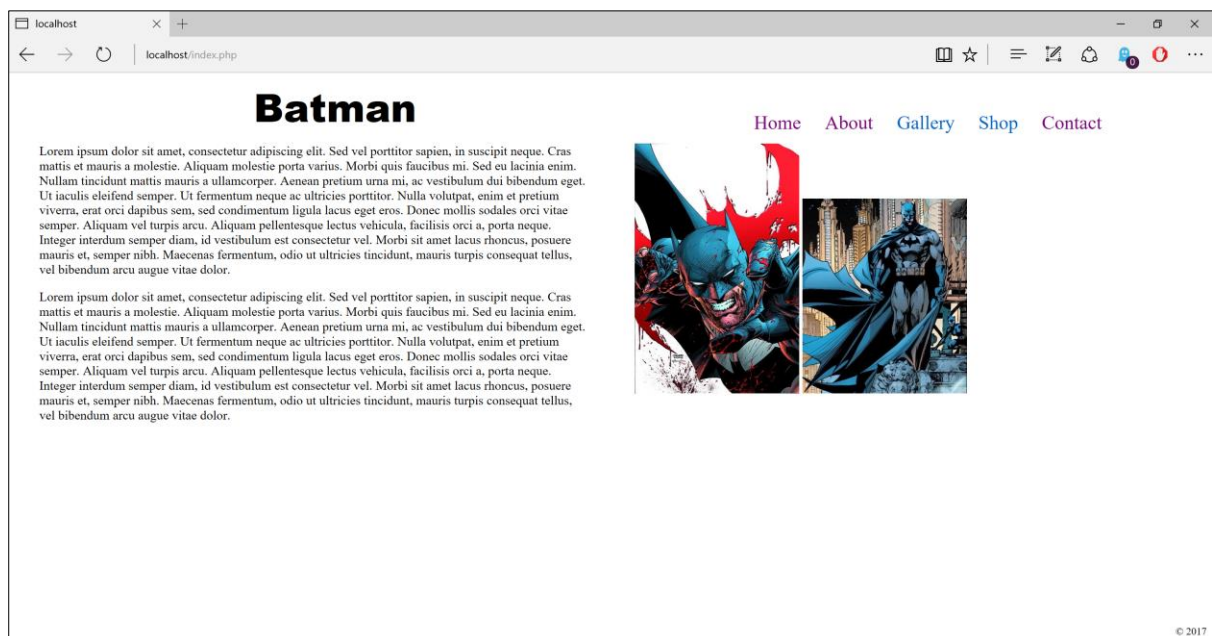


## About.php

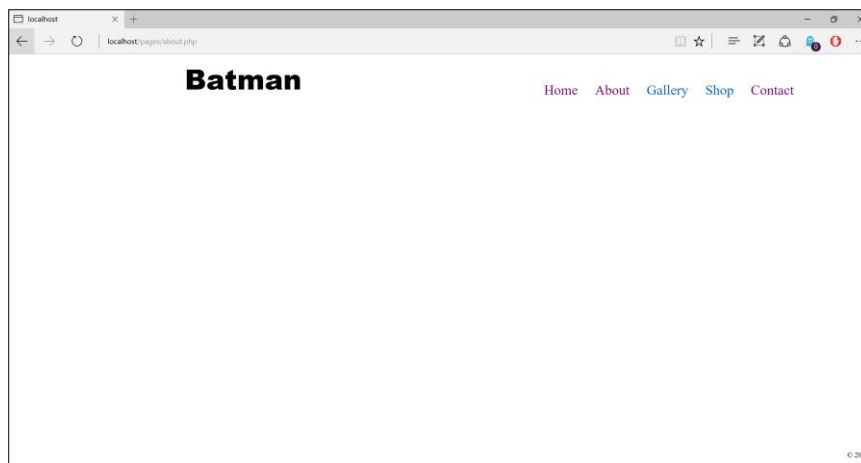
```
1  <?php
2      include_once('../functions.php');
3  ?>
4  <!DOCTYPE html>
5  <html>
6  <head>
7      <link href="../stylesheet.css" type="text/css" rel="stylesheet">
8      <style type="text/css">
9      </style>
10 </head>
11 <body>
12     <?php
13         headerLayout('about');
14     ?>
15     <div id="main">
16         <div id="left"></div><!-- eo left -->
17         <div id="right"></div><!-- eo right -->
18     </div><!-- eo main -->
19     <?php
20         footerLayout();
21     ?>
22 </body>
23 </html>
```

Now save and test, you should be able to navigate from the index page to the about page with no problems at all

## Live index.php



## Live about.php



The only real issue is the header details, so, let's fix that up with some php code.

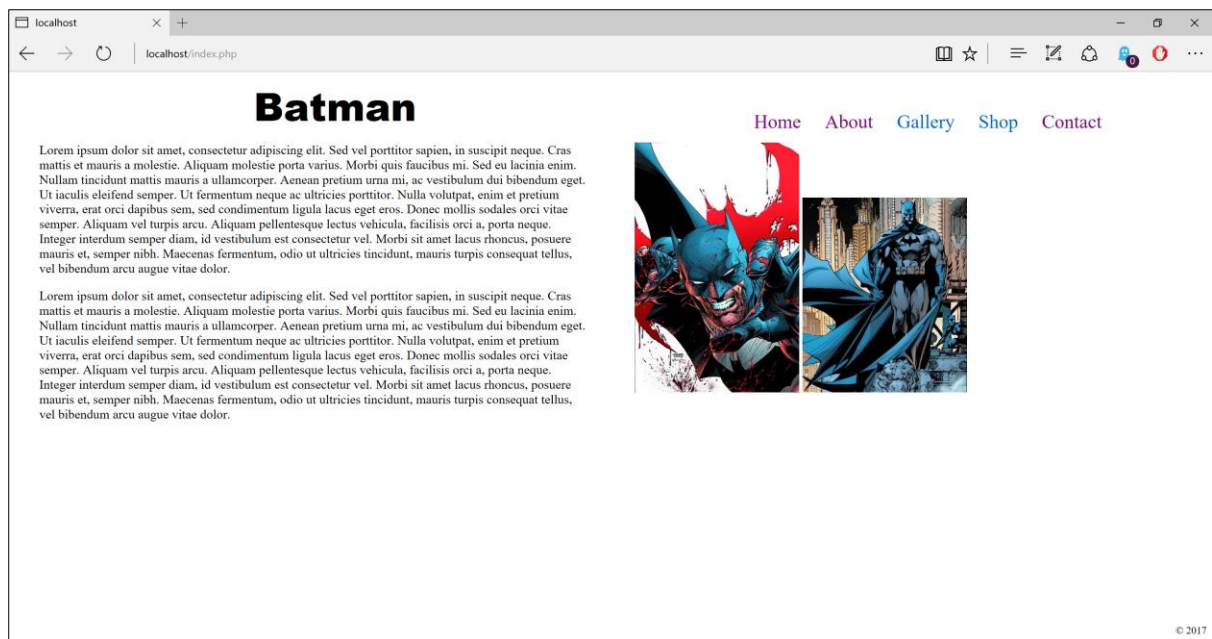
## Functions.php

```
1  <?php
2
3  function headerLayout($page)
4  {
5      if ($page == 'index')
6      {
7          echo ' <div id="header"><span class="headerText">Batman</span>';
8      }
9      else
10     {
11         echo ' <div id="header"><span class="headerText">'. $page. '</span>';
12     }
13     navigation($page);
14     echo '</div><!-- eo header -->';
15 }
```

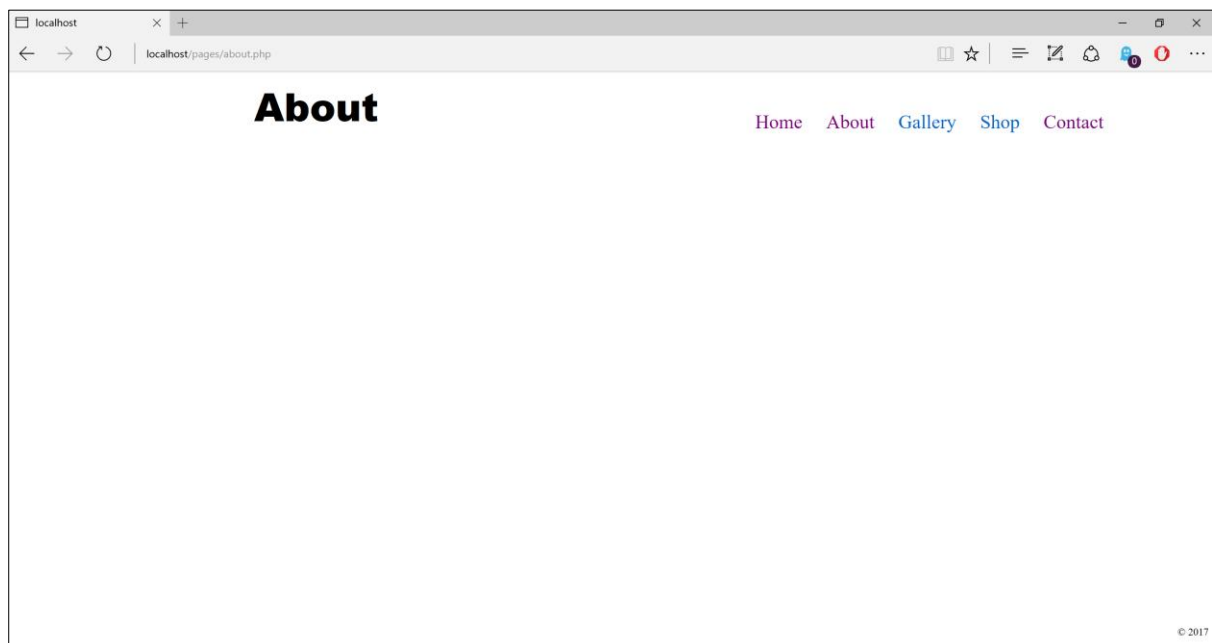
## About.php

```
1  <?php
2      include_once('../functions.php');
3      $page = 'About';
4  ?>
5  <!DOCTYPE html>
6  <html>
7  <head>
8      <link href="../stylesheet.css" type="text/css" rel="stylesheet">
9      <style type="text/css">
10
11     </style>
12 </head>
13 <body>
14     <?php
15         headerLayout($page);
16     ?>
17     <div id="main">
18         <div id="left"></div><!-- eo left -->
19         <div id="right"></div><!-- eo right -->
20     </div><!-- eo main -->
21     <?php
22         footerLayout();
23     ?>
24 </body>
25 </html>
```

## Live index.php



## Live about.php



Now that we have the header being written up by code, we can quickly make the other pages. By doing a file→save as on the about.php and then changing the content of \$page on every page.

## Gallery.php

```
1  <?php
2      include_once('../functions.php');
3      $page = 'Gallery|';
4  ?>
5  <!DOCTYPE html>
6  <html>
7      <head>
8          <link href="../stylesheet.css" type="text/css" rel="stylesheet">
9          <style type="text/css">
10             </style>
11      </head>
12      <body>
13          <?php
14              headerLayout($page);
15          ?>
16          <div id="main">
17              <div id="left"></div><!-- eo left -->
18              <div id="right"></div><!-- eo right -->
19          </div><!-- eo main -->
20          <?php
21              footerLayout();
22          ?>
23      </body>
24  </html>
```

## Shop.php

```
1  <?php
2      include_once('../functions.php');
3      $page = 'Shop|';
4  ?>
5  <!DOCTYPE html>
6  <html>
7      <head>
8          <link href="../stylesheet.css" type="text/css" rel="stylesheet">
9          <style type="text/css">
10             </style>
11      </head>
12      <body>
13          <?php
14              headerLayout($page);
15          ?>
16          <div id="main">
17              <div id="left"></div><!-- eo left -->
18              <div id="right"></div><!-- eo right -->
19          </div><!-- eo main -->
20          <?php
21              footerLayout();
22          ?>
23      </body>
24  </html>
```

## Contact.php

```
1  <?php
2      include_once('../functions.php');
3      $page = 'Contact';
4  ?>
5  <!DOCTYPE html>
6  <html>
7  <head>
8      <link href="../stylesheet.css" type="text/css" rel="stylesheet">
9      <style type="text/css">
10         </style>
11     </head>
12     <body>
13         <?php
14             headerLayout($page);
15         ?>
16         <div id="main">
17             <div id="left"></div><!-- eo left -->
18             <div id="right"></div><!-- eo right -->
19         </div><!-- eo main -->
20         <?php
21             footerLayout();
22         ?>
23     </body>
24 </html>
```

Once all of the pages have been saved, you can test them in a browser via localhost. The navigation and headers should all change and work as expected.

Like all webpages, the structure has now been completed and each page is primarily different due to the content on that page. Based off the structure we have just completed, all of the differences can be done via the modification of elements that are situated within the main div tag.

## Backup

This concludes tutorial. Save your work (c:\xampp\htdocs) to USB, student drive, Onedrive, Dropbox or zip and email the folder to yourself.