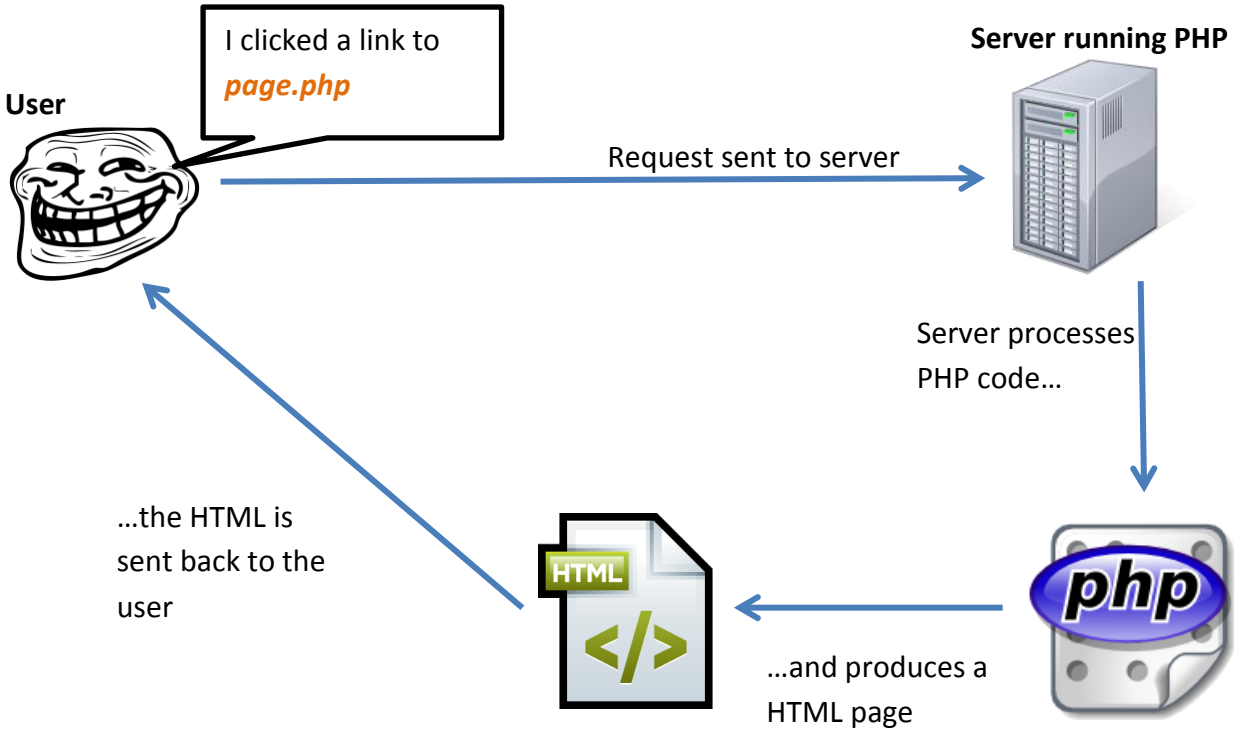
What is PHP?

PHP is a scripting language primarily used to make interactive web pages. It is a **server side** language which means it does its processing on a server, a bit like this:



So…the user only ever sees HTML pages. They **cannot** see the PHP code!

What is PHP good at?	What is PHP bad at?
It is easy to connect your website to a database in order to store data	In order for anything to update on the page a request for a new page must be sent to the server (e.g. by clicking a link or submitting a form)
It has lots of useful built in functions you can use to help you, and good docs on php.net	You have to run a server (or pay for hosting with PHP) or your code will not run
You can use HTML to make your interface – easy!	Not really PHP's fault, but you still have to contend with designing for multiple browsers.
It can be used procedurally or in Object Oriented style	The Object Oriented support in PHP is relatively new and some people don't like it.

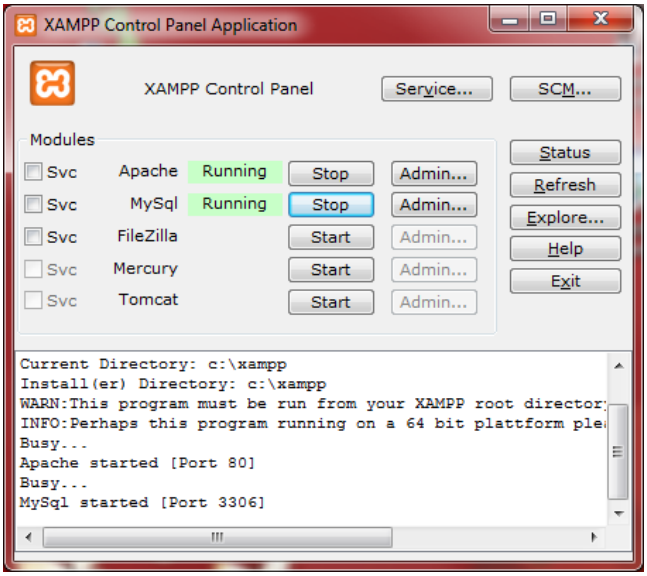
How do I set up a server?

Happily, there are some nice people on the internet at http://apachefriends.org who have made things very easy! They offer a package called **XAMPP** which contains everything we need in one easily installable package. It even works on OSX and Linux (although my instructions were written using Windows).

XAMPP contains…

- Apache – a web server to send out the pages
- PHP – interpreter for the PHP language
- MySQL – database software
- Some other stuff we don't really need to care about ☺

1. Download either **XAMPP** (the full Monty) or **XAMPP USB Lite** if you want the small version so that you can install it on a USB stick or other portable hard drive.
2. Open the installer and follow the instructions. It is best to install XAMPP in a permanent location rather than on the desktop.
3. Once the installer has finished, find the **xampp-control.exe** (you might want to make a shortcut to this) and run it. Click on "Start" next to the Apache and MySQL options and make sure they go green and say "Running". You can now close the window.



Uh oh…why isn't it working?

Obviously network environments vary and school networks are often frustratingly locked down. Apache is a web server which operates on port 80, and MySQL uses port 3306. If you have **Skype** or some other communication program running, this often uses port 80 meaning Apache won't run – so close it and try again. For other problems consult http://apachefriends.org or your friendly network technician (bring tea?)

How do I write PHP programs?

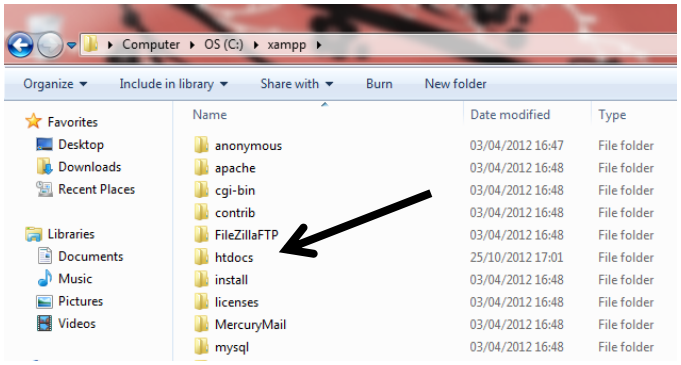
First, let's test whether your server is actually working. Assuming you managed to start Apache and MySQL on the previous page, open up your browser and type in

http://localhost/ (or if you're feeling techy, http://127.0.0.1 which means the same)

You should see the XAMPP test page if it's working correctly, if not see the bottom of the previous page for troubleshooting advice.

Where do I put my programs?

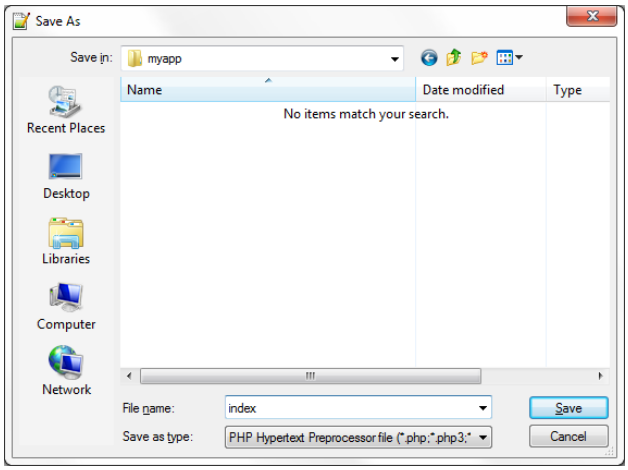
When we write PHP programs we need to save them in a special folder on the server called **htdocs**. This is the web root – it's where the server looks for pages. Find it by navigating to the place where your XAMPP was installed and looking for the htdocs folder, here is mine:



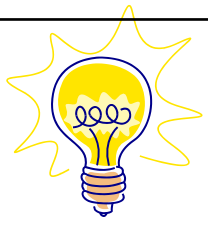
1. Make a subfolder inside the htdocs folder called **myapp**

2. Open up a blank file in a text editor of your choice (I recommend Notepad++) and save your file as index.php inside the **myapp** folder

Be careful that you don't accidentally save the file as **index.php.txt** which can easily happen in Notepad if you don't change the 'Save as type' dropdown to "*.* All files". In Notepad++, just change the 'Save as type' to PHP and that should do the job nicely.



You might be wondering why we have called our page **index.php** ? Apache (the web server) always looks for a page called index first – this is the home page of any site or app we may make. Other files can just be saved as ***filename.php*** and will work just as well.

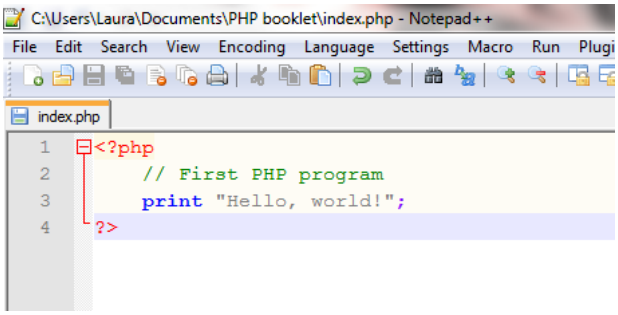


How do I write a program?

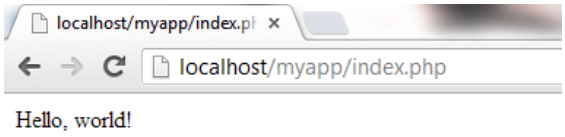
Now we need to write some PHP code! PHP is always written within PHP tags which look like this:

```
<?php        ?>
```

Here is my program – I'm sure you can tell that the green part is a *comment*.



Now run the program by visiting this address in your browser:



http://localhost/myapp

Congratulations, you are now a PHP programmer!

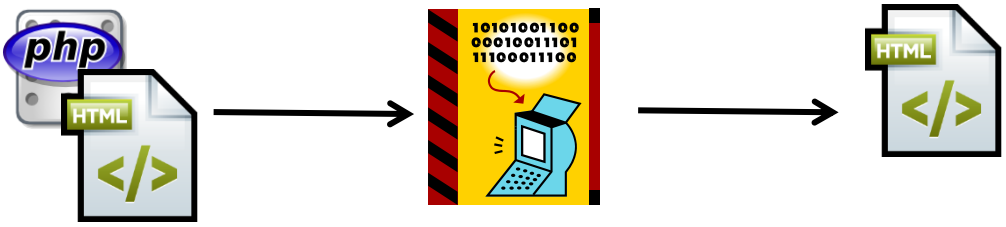
If you did not call your folder **myapp**, your address will be http://localhost/yourfoldername - replace yourfoldername with the name you called your folder.

What you should have understood so far

- When you want to work on PHP code you need to start Apache/MySQL on the XAMPP control panel
- Save all of your PHP documents into the **htdocs** folder with the file extension **.php**
- PHP files can be inside subfolders of htdocs
- Run your PHP file by browsing to http://localhost/foldername
- The default page (home page) of each folder is **index.php**
- If you want to view a page with a different name, open http://localhost/foldername/pagename.php

PHP mixed with HTML

As I mentioned at the start, to create the interface for PHP programs you can use HTML. This means we can combine PHP and HTML code in our program:

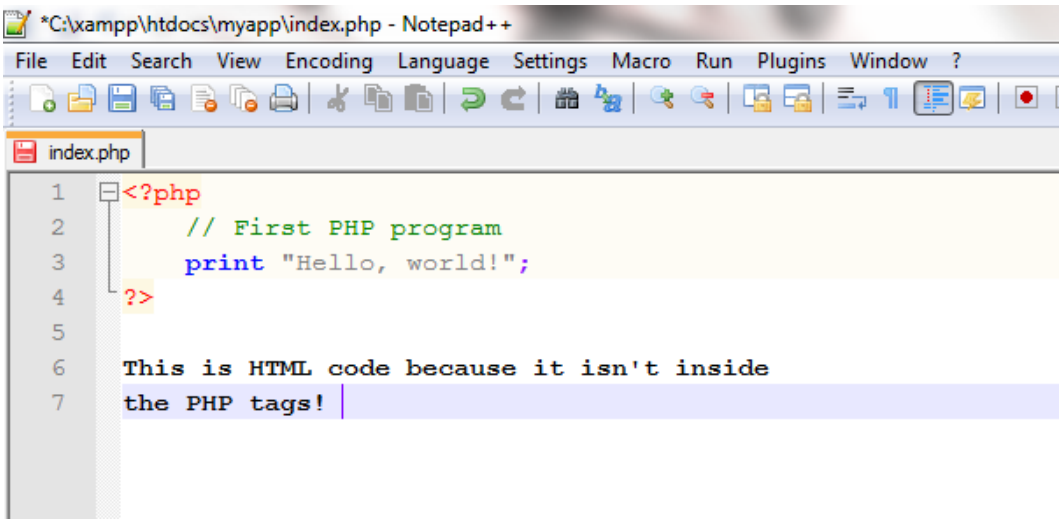


Document contains both PHP and HTML code

PHP statements *evaluated* to produce HTML code

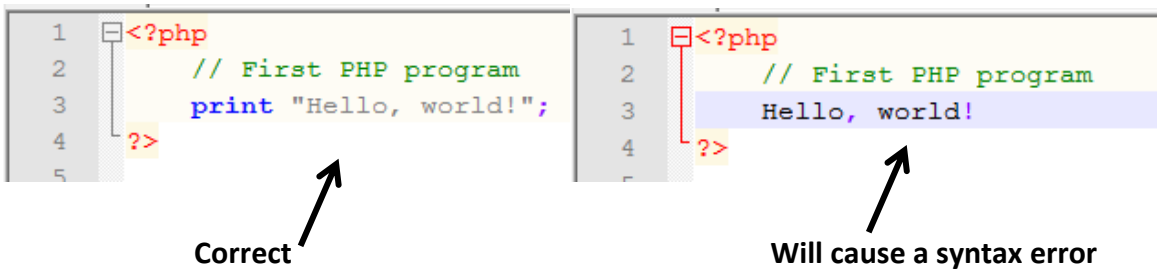
Final output is **only** HTML code

You can easily tell which parts are HTML code and which are PHP code because PHP code always has to be inside the PHP tags **<?php** and **?>**



Things to watch out for

1. If you want to output something from a PHP section, you have to print it as a string



Correct

Will cause a syntax error

2. Sometimes you may want to put some HTML code inside a PHP section. This is fine, but you must treat it as if it were just another string to be printed:

```php
<?php
    // First PHP program
    <strong>
    print "Hello, world!";
    </strong>
?>
```

This will cause a syntax error because the HTML code is inside a PHP section

```php
<?php
    // First PHP program
    print "<strong>";
    print "Hello, world!";
    print "</strong>";
?>
```

This will work as you have treated the HTML tags as strings

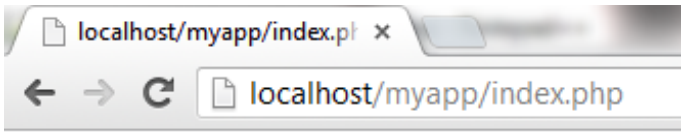
You can also do this

```php
<?php
    // First PHP program
    print "<strong>Hello, world!</strong>";
?>
```

Concatenating variables

You can concatenate variables into your output using the concatenation operator which is a dot in PHP:

```php
<?php
    $name = "Laura";
    print "Hello ".$name;
?>
```



Hello Laura

PHP and HTML exercise

Test your understanding of the differences between HTML and PHP

HTML	PHP
Is a _____ language.	Is a _____ language.
Tells the _____ how to _____ the items on the page.	Processes the code on the _____ and then sends the result back to the browser.
Each tag begins with a ___ and ends with a ___	Most lines end with a ___
The bit within a tag such as *"color=blue"* is known as an _____	A line of code is known as a _____
	Can temporarily store data in a _____
	Can be used to connect to a _____

Mark-up	Statement	<
RAM	Scripting	Display
Variable	Server	>
Internet	;	Database
Browser	Code	Attribute

Score: / 12

Basic PHP

You will already be familiar with another programming language from your AS work. Here is a quick reference of the basic statement syntax for PHP

Assignment	Variable names in PHP have a dollar sign ($) in front of them, e.g.
	
	`$name = "Bob";`
	`$age = 10;`
	
	The assignment operator is a single equals =

Selection	`if ( condition ){`
	`  // statements here`
	`}`
	`elseif ( condition ) {`
	`  // statements here`
	`}`
	`else {`
	`  // statements here`
	`}`

While loop	`while( condition ){`
	`  // statements here`
	`}`

For loop	`for( $i = 0; $i < 5; $i++ ){`
	`  // statements here`
	`}`
	
	`$i = 0` → Create a variable `$i` and set its starting value to 0
	`$i < 5` → Condition to test. While this condition remains true, the loop will execute
	`$i++` → Increment (add one) to `$i`

Do-while loop	`do {`
	`  // statements`
	`while ( condition );`

Operators	The comparison operator is a double equals == Logical 'and' operator is `&&` Logical 'or' operator is `||` (pipe is to the left of Z on a Windows keyboard) The concatenation operator is a dot .