

# PHP for Site Modularity

## Content Management



JANUARY 9, 2014 • 38 COMMENTS

## The Challenges And Rewards Of Teaching Web Design

# Inspiration <sup>353</sup> # Opinion Column <sup>91</sup> # Community <sup>53</sup> # Global Web Design <sup>96</sup>



### ABOUT THE AUTHOR

Jeremy Girard was born with six toes on each foot. The extra toes were removed before he was a year old, robbing him of any super-powers and ending his ... [More about Jeremy...](#)



**Just over four years ago, I decided to take a part-time position** teaching website design and front-end development at a local university, [the University of Rhode Island](#). My time in the classroom has been one of the most challenging as well as one of the most rewarding experiences in my career, and I believe that other Web professionals would greatly benefit from spending some time in an educational setting teaching others their craft.

In this article, we will look at some of the challenges to prepare for if you are considering taking on a teaching position. I will also present some of my personal experiences and insights, including the benefits and rewards I've enjoyed as a teacher, to help you consider such a move for your own career.

### Further Reading on SmashingMag:

- [Mistakes Developers Make When Learning Design](#)
- [Stop Shouting, Start Teaching](#)
- [Lessons Learned From Leading New Web Professionals](#)
- [Rekindling Your Passion For Web Design](#)

### Challenges

Let's start by looking at some of those aforementioned challenges as well as the solutions I've used to meet them over the years.

# Big sites

The management of big websites presents a whole range of considerations that don't necessarily apply to small websites. Maybe you have lots of content that needs to be searchable by category, maybe you want to enable user-generated content or perhaps the page structure is complex, and you need an easy way to manage and update those pages.

When faced with such issues, most people will think about using a CMS (Content Management System) such as WordPress.

The power behind most CMSs is the PHP scripting language and to understand how a CMS works, we need to gain at least a basic understanding of PHP.

# Front-End/Back-End

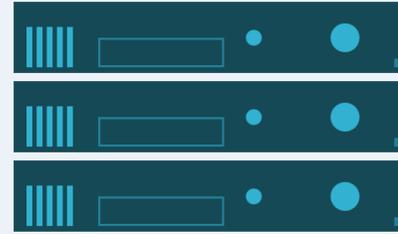


## Client-side Technologies

HTML

CSS

JavaScript



## Server-side Technologies

PHP

MySQL

Apache

The term “full-stack developer” refers to someone who can develop across the client/server boundary. Someone who is comfortable working with both client-side and server-side technologies.

# LAMP P

Linux  
Apache  
MySQL  
PHP

operating system  
web server  
database  
scripting

# What is PHP?

- PHP is a “server-side” scripting language specifically designed for producing dynamic web pages.
- PHP is the most widely used server-side web scripting language in the world.
- Many significant websites are developed using PHP (Facebook, Wikipedia, Microsoft, and anything based on WordPress).
- PHP is an open source project, which means that PHP is free to use, and this has been a major factor in its popularity.
- Is PHP still relevant?

# PHP in statistics

- **77% of all live websites** use PHP.
- PHP powers **22% of the top 10,000** sites.
- The **PHP used by Facebook in 2004** is still used today by around **18,000 websites**.
- **81% of all PHP uses** are still for **web development**.
- PHP developers earn an average **annual revenue of over \$94,000**.
- **25.8% of web developers** prefer PHP.
- **4 countries** have more than **1 million PHP-powered websites**.
- Only **6.2% of developers** want to learn PHP.
- There are **700,000 searches per month** on Google for the term PHP.

# What is the history of PHP?

- PHP (version 1.0) was announced in 1995 by Rasmus Lerdorf.
- His original name for it was *Personal Home Page* Tools (PHP Tools) and it was written in the C programming language.
- Usage of PHP really took off after the release of version 3.0 in 1998.
- PHP version 7.4 (2020) is now common on most web servers (the PHP 6 development project has been abandoned).
- some web servers may still run version 5.6 or earlier for backwards compatibility.
- PHP 8.x is the most recent version (2021) and most web hosts are now supporting it.

# What does “PHP” stand for?

The original name (personal home page) has been changed and the scripting language is now called:

*PHP Hypertext Pre-processor*

a recursive acronym!

cPanel - MultiPHP Manager

https://springfishblue.uk:2083/cpsess6529974611/frontend/paper\_lantern/multipl...

CLOOK Search (/) springfi LOGOUT

## MultiPHP Manager

### System PHP Version

The system default PHP version is set by the system administrator. Any domain that is set to the *inherit* value indicates that it does not have its own PHP version explicitly set. Read more about [inherit](#).

**PHP 7.4 (ea-php74)**

### Set PHP Version per Domain

Select the domains that you wish to change from the table, then apply a PHP version from the list.

Your hosting provider recommends PHP 7.3 or PHP 7.4.

PHP Version PHP 5.3 (ea-php53) Apply

Search

Selected 0

<input type="checkbox"/>	Domain	PHP Version
<input type="checkbox"/>	springfishblue.uk	PHP 7.4 (ea-ph...

- PHP 5.3 (ea-php53)
- PHP 5.4 (ea-php54)
- PHP 5.5 (ea-php55)
- PHP 5.6 (ea-php56)
- PHP 7.0 (ea-php70)
- PHP 7.1 (ea-php71)
- PHP 7.2 (ea-php72)
- PHP 7.3 (ea-php73) - Recommended
- PHP 7.4 (ea-php74) - Recommended
- PHP 8.0 (ea-php80)
- inherit

cPanel 94.0.22

Home Trademarks Privacy Policy Documentation

The hosting control panel (cPanel) at Clook allows you to select which version of PHP you would like to use. Some web apps (e.g. WordPress) require specific versions to work correctly.

# How does PHP work?

- PHP scripts are embedded within HTML markup.
- The file must be saved with a “.php” extension so that the server knows how to treat it.
- When a user requests a PHP page, the server passes the page to the PHP *interpreter*, which runs the scripts and passes the compiled page back to the web server for delivery to the user.

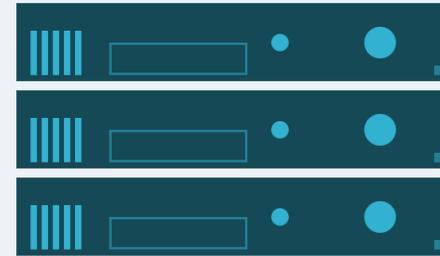
`index.php`

# The HTML server request



Server request for  
index.html

1



index.html sent  
to browser

2



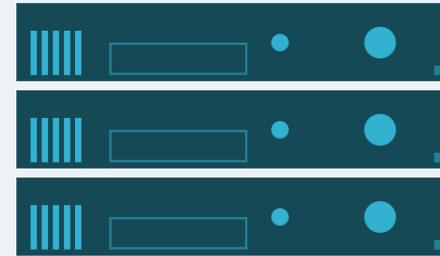
The HTML server request is a simple transaction. The browser asks for a file and the server sends that file to the browser.

# The PHP server request



Server request for  
index.php

1



Parsed version of  
index.php sent  
to browser

4



index.php sent  
to the PHP  
interpreter

3



index.php returned to  
the web server after it  
has been parsed

2



The PHP server request is more complicated. The original PHP file is **never** sent to the browser. Instead, it is sent to the PHP interpreter to be *parsed*. The browser is sent the parsed version.

# How easy is PHP to learn?

- PHP is pretty easy to learn, it is logical and sequential.
- If you have done any programming at all, you'll pick it up very easily.
- If you've never done any programming, PHP is a good place to start.
- If JavaScript scares you, you'll find PHP makes you only slightly nervous.
- Here are a few examples...

# What does PHP look like?

```
<html>
  <head>
    <title>PHP Test</title>
  </head>
  <body>
    <?php echo "<p>Hello World</p>"; ?>
  </body>
</html>
```

Before parsing

PHP scripts are always enclosed within special tags `<?php script ?>`

Each *statement* is terminated with a semi-colon `;`

Notice that, unlike JavaScript, it's OK to write PHP inline with HTML.

# What does the browser see?

```
<html>  
  <head>  
    <title>PHP Test</title>  
  </head>  
  <body>  
  
    <p>Hello World</p>  
  
  </body>  
</html>
```

After parsing

After *parsing*, the PHP script is replaced with the output. In this case, just an HTML paragraph, so the browser never gets to see any PHP.

# No need to declare variables

```
<?php  
$page_name = "some text" ;  
$page_number = 5 ;  
?>
```

Variables can be created simply by stating a name and assigning a value. Notice that text *strings* are quoted but numbers are not (just as in JavaScript).

Variables in PHP are represented by a dollar sign followed by the name of the variable. The name must begin with a letter or an underscore but **not** a number.

It's common practice in PHP to use the underscore character as a separator for variable names. CamelCase is not used.

# Using variables for consistency

```
<?php $page_name = "PHP Test"; ?>
<html>
  <head>
    <title><?php echo $page_name; ?></title>
  </head>
  <body>
    <h1><?php echo $page_name; ?></h1>
  </body>
</html>
```

What the designer writes

Variables are just containers that hold bits of information.

Variable names in PHP always begin with a dollar sign \$

# The browser only sees the result

```
<html>  
  <head>  
    <title>PHP Test</title>  
  </head>  
  <body>  
    <h1>PHP Test</h1>  
  </body>  
</html>
```

What the user's browser sees

The `echo` function tells the parser to print (or echo) a string or the contents of a variable. In this case, the `$page_name` variable is replaced with its contents.

# Making pages intelligent

```
<html>
  <head>
    <title>Today</title>
  </head>
  <body>
    <?php
    $today = date("l");
    echo "    <p>Today is $today</p>";
    if ($today == "Saturday" OR $today == "Sunday"){
      echo "    <p>Hooray! It's the weekend.</p>";
    }else{
      echo "    <p>Oh, it's just another work day.</p>";
    }
    ?>
  </body>
</html>
```



*That's a lower-case "L"*

What the designer writes

This script finds out what day it is using the `date` function and then decides if it's a weekend. It then prints one message if it is a weekend and another if it's a weekday. Notice how much easier it is to do this in PHP than in JavaScript.

# If today is Wednesday...

```
<html>
  <head>
    <title>Today</title>
  </head>
  <body>
    <p>Today is Wednesday</p>
    <p>Oh, it's just another work day.</p>
  </body>
</html>
```

What the user's browser sees

Once the script has been parsed by the web server and passed to the browser, the html looks like this. The user is completely unaware of the process, all they see is the end result (HTML).

This is much easier than JavaScript because we don't have to manipulate the DOM, we can just write whatever markup we want as part of our script.

PHP can do many things but its ability to build pages “on the fly” from component parts is the killer feature for web designers like us.

# Site modularity

If you look at any well-designed website, you should notice that each page contains several common elements.

The page on the left has 3 elements that are common to all pages in the same site, the masthead or branding, the sidebar and the footer.

Both the sidebar and the footer contain content that may need to be updated frequently. For example, site navigation will need to be updated if new pages are added to the site.

It can be difficult to keep all navigation links co-ordinated on a complex site because any one change needs to be made to all pages in the site, and there may be hundreds of pages.

**MA Web Design & Content Planning**

Study web design in London at the University of Greenwich

**T**HE MA WEB DESIGN & CONTENT PLANNING PROGRAMME takes graduates from many disciplines and provides them with the hands-on skills necessary to plan, design, build and promote an effective website in relation to the needs of a business, profession, government office or organisation. The programme is also suitable for those who are considering a future in freelance web design. Applicants should have an interest in design and a working familiarity with the software employed in a modern office environment. See [Our students](#) for more information.

There are many web design courses in London but we are the only institution currently offering a post-graduate MA in this fast-evolving subject.

**Learning web design**

The design of excellent websites presents a notable intellectual challenge and on graduating from the programme, students will have the skills to design, publish and manage a corporate intranet or a public website. In addition, students will learn how to promote and generate revenue from public websites. The objective is to provide a balanced education in the three aspects of web design: technical, functional and aesthetic. See [Our philosophy](#) for more information.

The technical aspects of the programme provide an understanding of the wide range of tools and technologies that are available for the design and development of websites. This includes the creation of semantically correct XHTML and CSS based design using web standards.

The functional aspects focus on the creation of websites using good information architecture and that demonstrate best practice in accessibility, usability and findability. Students develop an understanding of how people use the web, how they like to navigate and search for information and the importance of SEO. See [Design principles](#) for more information.

The aesthetic aspects of web design are considered with regard to both graphic/artistic and business objectives, such as corporate identity and branding. The architectural principle that form should follow function is relevant, not as a rule but because aesthetic considerations must be judged with regard to the technical and functional aspects of a website.

**Flexible teaching mode**

The timetable is designed to make it easy for those in full-time employment to attend the programme and students have the option of a full-time (one day per week) or part-time (one half day per week) mode of study. Although contact time is relatively short, students should be aware that there will be a substantial amount of course work, project work and reading to be completed in their own time. See [Programme details](#) for more information.

We believe our programme is unique in offering a balanced blend of the technical, functional and aesthetic aspects of web design and we are the only university in London offering such a focused programme at masters level.

**Need to know more?**

If after reading through this website, you still have some questions about the MA Web Design & Content Planning programme, please get in touch using our [contact form](#) and we'll do our best to answer your questions.

**Programme details**  
**Core courses**  
**Our philosophy**  
**Design principles**  
**Our students**  
**Frequent questions**  
**Web design bookshelf**  
**Contact**

MA Web Design & Content Planning (full-time and part-time modes) now recruiting for September 2010. Find out more...

**Redesign this site**  
Select one of the colour options below to change the way this site looks. How does this work?

**Frequently asked questions**

- Will we learn how to use Dreamweaver?
- At what time of year does the programme begin?
- What do students do after graduation?
- How much study time will I need each week?
- Where are lectures held?
- Where can I get financial advice?

More FAQs...

**Student opinion**  
Most students tell us that the programme has exceeded their expectations. See [Our students](#).

**Latest forum topics**

- Abal/ISP Redesign
- New 5 for 1 offer at Sit...
- SEO for Universal Search
- Smashing Magazine on the...
- WordPress
- WordPress & Itaccess
- Careers
- 53 CSS-based techniques
- Could this campaign see...
- Should the BBC dump half...

More forum topics...

**Site Navigation**  
[Site homepage](#)  
[Programme details](#)  
[Core courses](#)  
[Our philosophy](#)  
[Design principles](#)  
[Our students](#)  
[Frequent questions](#)  
[Web design bookshelf](#)  
[Contact](#)

**Our Favourite Websites**  
[A List Apart](#)  
[Digital Web Magazine](#)  
[Smashing Magazine](#)  
[456 Berea Street](#)  
[Think Vitamin](#)  
[Added Bytes](#)  
[i love typography](#)  
[Sitepoint](#)  
[zeldman.com](#)

**University of Greenwich**  
[UoG homepage](#)  
[School of Architecture & Construction](#)  
[Avery Hill Campus](#)  
[Postgraduate prospectus](#)  
[A guide to postgraduate study](#)  
[How to apply](#)  
[Tuition fees](#)  
[Contact details](#)  
[UoG Students' Union](#)

Copyright © 2003 – 2010 David Watson & Tom Turner | XHTML | CSS | Accessibility | Site Map

# Common elements

**MA Web Design & Content Planning**

## Study web design in London at the University of Greenwich

**T**HE MA WEB DESIGN & CONTENT PLANNING PROGRAMME takes graduates from many disciplines and provides them with the hands-on skills necessary to plan, design, build and promote an effective website in relation to the needs of a business, profession, government office or organisation. The programme is also suitable for those who are considering a future in freelance web design. Applicants should have an interest in design and a working familiarity with the software employed in a modern office environment. See [Our students](#) for more information.

There are many web design courses in London but we are the only institution currently offering a post-graduate MA in this fast-evolving subject.

### Learning web design

The design of excellent websites presents a notable intellectual challenge and on graduating from the programme, students will have the skills to design, publish and manage a corporate intranet or a public website. In addition, students will learn how to promote and generate revenue from public websites. The objective is to provide a balanced education in the three aspects of web design: technical, functional and aesthetic. See [Our philosophy](#) for more information.

The technical aspects of the programme provide an understanding of the wide range of tools and technologies that are available for the design and development of websites. This includes the creation of semantically correct XHTML and CSS based design using web standards.

The functional aspects focus on the creation of websites using good information architecture and that demonstrate best practice in accessibility, usability and findability. Students develop an understanding of how people use the web, how they like to navigate and search for information and the importance of SEO. See [Design principles](#) for more information.

The aesthetic aspects of web design are considered with regard to both graphic/artistic and business objectives, such as corporate identity and branding. The architectural principle that form should follow function is relevant, not as a rule but because aesthetic considerations must be judged with regard to the technical and functional aspects of a website.

### Flexible teaching mode

The timetable is designed to make it easy for those in full-time employment to attend the programme and students have the option of a full-time (one day per week) or part-time (one half day per week) mode of study. Although contact time is relatively short, students should be aware that there will be a substantial amount of course work, project work and reading to be completed in their own time. See [Programme details](#) for more information.

We believe our programme is unique in offering a balanced blend of the technical, functional and aesthetic aspects of web design and we are the only university in London offering such a focused programme at masters level.

### Need to know more?

If after reading through this website, you still have some questions about the MA Web Design & Content Planning programme, please get in touch using our [contact form](#) and we'll do our best to answer your questions.

**Site Navigation**

- Site homepage
- Programme details
- Core courses
- Our philosophy
- Design principles
- Our students
- Frequent questions
- Web design bookshelf
- Contact

**Our Favourite Websites**

- A List Apart
- School of Architecture & Construction
- Smashing Magazine
- 456 Berea Street
- Think Vitamin
- Added Bytes
- i love typography
- Sitepoint
- zeldman.com

**University of Greenwich**

- UCG homepage
- School of Architecture & Construction
- Avery Hill Campus
- Postgraduate prospectus
- A guide to postgraduate study
- How to apply
- Tuition fees
- Contact details
- UCG Students' Union

Copyright © 2003 – 2010 David Watson & Tom Turner | [XHTML](#) | [CSS](#) | [Accessibility](#) | [Site Map](#)

**MA Web Design & Content Planning**

## Study web design in London at the University of Greenwich

**T**HE MA WEB DESIGN & CONTENT PLANNING PROGRAMME takes graduates from many disciplines and provides them with the hands-on skills necessary to plan, design, build and promote an effective website in relation to the needs of a business, profession, government office or organisation. The programme is also suitable for those who are considering a future in freelance web design. Applicants should have an interest in design and a working familiarity with the software employed in a modern office environment. See [Our students](#) for more information.

There are many web design courses in London but we are the only institution currently offering a post-graduate MA in this fast-evolving subject.

### Learning web design

The design of excellent websites presents a notable intellectual challenge and on graduating from the programme, students will have the skills to design, publish and manage a corporate intranet or a public website. In addition, students will learn how to promote and generate revenue from public websites. The objective is to provide a balanced education in the three aspects of web design: technical, functional and aesthetic. See [Our philosophy](#) for more information.

The technical aspects of the programme provide an understanding of the wide range of tools and technologies that are available for the design and development of websites. This includes the creation of semantically correct XHTML and CSS based design using web standards.

The functional aspects focus on the creation of websites using good information architecture and that demonstrate best practice in accessibility, usability and findability. Students develop an understanding of how people use the web, how they like to navigate and search for information and the importance of SEO. See [Design principles](#) for more information.

The aesthetic aspects of web design are considered with regard to both graphic/artistic and business objectives, such as corporate identity and branding. The architectural principle that form should follow function is relevant, not as a rule but because aesthetic considerations must be judged with regard to the technical and functional aspects of a website.

### Flexible teaching mode

The timetable is designed to make it easy for those in full-time employment to attend the programme and students have the option of a full-time (one day per week) or part-time (one half day per week) mode of study. Although contact time is relatively short, students should be aware that there will be a substantial amount of course work, project work and reading to be completed in their own time. See [Programme details](#) for more information.

We believe our programme is unique in offering a balanced blend of the technical, functional and aesthetic aspects of web design and we are the only university in London offering such a focused programme at masters level.

### Need to know more?

If after reading through this website, you still have some questions about the MA Web Design & Content Planning programme, please get in touch using our [contact form](#) and we'll do our best to answer your questions.

**Site Navigation**

- Site homepage
- Programme details
- Core courses
- Our philosophy
- Design principles
- Our students
- Frequent questions
- Web design bookshelf
- Contact

**Our Favourite Websites**

- A List Apart
- Digital Web Magazine
- Smashing Magazine
- 456 Berea Street
- Think Vitamin
- Added Bytes
- i love typography
- Sitepoint
- zeldman.com

**University of Greenwich**

- UCG homepage
- School of Architecture & Construction
- Avery Hill Campus
- Postgraduate prospectus
- A guide to postgraduate study
- How to apply
- Tuition fees
- Contact details
- UCG Students' Union

Copyright © 2003 – 2010 David Watson & Tom Turner | [XHTML](#) | [CSS](#) | [Accessibility](#) | [Site Map](#)

Header

Sidebar

Footer

# Site Modularity

Wouldn't it be great if we could update the navigation links on our site by editing a single file – just like we can change the style of our site by editing a single CSS file?

With PHP we can do just that. All we need to do is to identify any common or repeating elements in our site design and put the markup for that element in a separate file.

We then use PHP to bring all the different parts together to make up the whole page.

The different component files are referred to as *server-side includes*.

The screenshot shows the MA Web Design & Content Planning website. Red arrows point to three key areas:

- Header:** The top navigation bar containing the site title and main menu items.
- Sidebar:** The right-hand column containing a list of links, a color selection tool, frequently asked questions, student opinion, and forum topics.
- Footer:** The bottom section containing site navigation, favorite websites, university information, and a logo.

The main content area includes an introductory paragraph about the program, a 'Learning web design' section, and several other informational paragraphs.

# Page schematic

For example, a code schematic for the page on the right might look something like this:

```
<html>
[HEAD SECTION]
<body>
[INCLUDE HEADER HERE]
<main>
[CONTENT GOES HERE]
</main>
[INCLUDE SIDEBAR HERE]
[INCLUDE FOOTER HERE]
</body>
</html>
```

The screenshot shows a website page for the MA Web Design & Content Planning programme at the University of Greenwich. The page is annotated with red arrows pointing to the Header, Sidebar, and Footer.

**Header:** MA Web Design & Content Planning

**Main Content:** Study web design in London at the University of Greenwich. The MA WEB DESIGN & CONTENT PLANNING PROGRAMME takes graduates from many disciplines and provides them with the hands-on skills necessary to plan, design, build and promote an effective website in relation to the needs of a business, profession, government office or organisation. The programme is also suitable for those who are considering a future in freelance web design. Applicants should have an interest in design and a working familiarity with the software employed in a modern office environment. See [Our students](#) for more information. There are many web design courses in London but we are the only institution currently offering a post-graduate MA in this fast-evolving subject.

**Learning web design**  
The design of excellent websites presents a notable intellectual challenge and on graduating from the programme, students will have the skills to design, publish and manage a corporate intranet or a public website. In addition, students will learn how to promote and generate revenue from public websites. The objective is to provide a balanced education in the three aspects of web design: technical, functional and aesthetic. See [Our philosophy](#) for more information.

The technical aspects of the programme provide an understanding of the wide range of tools and technologies that are available for the design and development of websites. This includes the creation of semantically correct XHTML and CSS based design using web standards.

The functional aspects focus on the creation of websites using good information architecture and that demonstrate best practice in accessibility, usability and findability. Students develop an understanding of how people use the web, how they like to navigate and search for information and the importance of SEO. See [Design principles](#) for more information.

The aesthetic aspects of web design are considered with regard to both graphic/artistic and business objectives, such as corporate identity and branding. The architectural principle that form should follow function is relevant, not as a rule but because aesthetic considerations must be judged with regard to the technical and functional aspects of a website.

**Flexible teaching mode**  
The timetable is designed to make it easy for those in full-time employment to attend the programme and students have the option of a full-time (one day per week) or part-time (one half day per week) mode of study. Although contact time is relatively short, students should be aware that there will be a substantial amount of course work, project work and reading to be completed in their own time. See [Programme details](#) for more information.

We believe our programme is unique in offering a balanced blend of the technical, functional and aesthetic aspects of web design and we are the only university in London offering such a focused programme at masters level.

**Need to know more?**  
If after reading through this website, you still have some questions about the MA Web Design & Content Planning programme, please get in touch using our [contact form](#) and we'll do our best to answer your questions.

**Programme details**  
[Core courses](#)  
[Our philosophy](#)  
[Design principles](#)  
[Our students](#)  
[Frequent questions](#)  
[Web design bookshelf](#)  
[Contact](#)

MA Web Design & Content Planning (full-time and part-time modes) now recruiting for September 2010. Find out more...

**Redesign this site**  
Select one of the colour options below to change the way this site looks. How does this work?

**Frequently asked questions**  
 Will we learn how to use Dreamweaver?  
 At what time of year does the programme begin?  
 What do students do after graduation?  
 How much study time will I need each week?  
 Where are lectures held?  
 Where can I get financial advice?  
More FAQs...

**Student opinion**  
Most students tell us that the programme has exceeded their expectations. See [Our students](#).

**Latest forum topics**  
 Aha!ISP Redesign  
 New 5 for 1 offer at Sit...  
 SEO for Universal Search  
 Smashing Magazine on the...  
 WordPress  
 WordPress & Itaccess  
 Careers  
 53 CSS-based techniques  
 Could this campaign see...  
 Should the BBC dump half...  
More forum topics...

**Site Navigation**  
[Site homepage](#)  
[Programme details](#)  
[Core courses](#)  
[Our philosophy](#)  
[Design principles](#)  
[Our students](#)  
[Frequent questions](#)  
[Web design bookshelf](#)  
[Contact](#)

**Our Favourite Websites**  
[A List Apart](#)  
[Digital Web Magazine](#)  
[Smashing Magazine](#)  
[456 Beraa Street](#)  
[Think Vitamin](#)  
[Added Bytes](#)  
[i love typography](#)  
[Sitepoint](#)  
[zeldman.com](#)

**University of Greenwich**  
[UoG homepage](#)  
[School of Architecture & Construction](#)  
[Avery Hill Campus](#)  
[Postgraduate prospectus](#)  
[A guide to postgraduate study](#)  
[How to apply](#)  
[Tuition fees](#)  
[Contact details](#)  
[UoG Students' Union](#)

Copyright © 2003 – 2010 David Watson & Tom Turner | [XHTML](#) | [CSS](#) | [Accessibility](#) | [Site Map](#)

# Includes

In reality, those include statements might look something like this:

```
<html>
[HEAD SECTION]
<body>
<?php include 'header.php'; ?>
<main>
[CONTENT GOES HERE]
</main>
<?php include 'sidebar.php'; ?>
<?php include 'footer.php'; ?>
</body>
</html>
```

In each case, the server replaces the include statement with the contents of the file indicated.

[The include function](#)

The screenshot shows a website layout for the MA Web Design & Content Planning programme. The layout is divided into three main sections indicated by red arrows:

- Header:** The top section containing the programme title "MA Web Design & Content Planning" and a brief introductory text about the programme.
- Sidebar:** A vertical column on the right side containing navigation links (e.g., "Programme details", "Core courses"), a "Redesign this site" section with a color selection tool, "Frequently asked questions", "Student opinion", and "Latest forum topics".
- Footer:** The bottom section containing "Site Navigation", "Our Favourite Websites", and "University of Greenwich" information, including contact details and a logo.

Additional content on the main page includes sections for "Study web design in London at the University of Greenwich", "Learning web design", "Flexible teaching mode", and "Need to know more?".

# Component files

The three files...

header.php

sidebar.php

footer.php

...are known as *server-side includes* because they are added to the main page by the server before passing the completed page to the user's browser for display.

In reality, the page on the right is built from 4 files, a parent file (index.php) and the 3 includes.

The file extension for includes can be anything you like. You will often see...

header.inc (**not** recommended)

...or...

header.inc.php (recommended)

MA Web Design & Content Planning

Study web design in London at the University of Greenwich

THE MA WEB DESIGN & CONTENT PLANNING PROGRAMME takes graduates from many disciplines and provides them with the hands-on skills necessary to plan, design, build and promote an effective website in relation to the needs of a business, profession, government office or organisation. The programme is also suitable for those who are considering a future in freelance web design. Applicants should have an interest in design and a working familiarity with the software employed in a modern office environment. See [Our students](#) for more information.

There are many web design courses in London but we are the only institution currently offering a post-graduate MA in this fast-evolving subject.

**Learning web design**

The design of excellent websites presents a notable intellectual challenge and on graduating from the programme, students will have the skills to design, publish and manage a corporate intranet or a public website. In addition, students will learn how to promote and generate revenue from public websites. The objective is to provide a balanced education in the three aspects of web design: technical, functional and aesthetic. See [Our philosophy](#) for more information.

The technical aspects of the programme provide an understanding of the wide range of tools and technologies that are available for the design and development of websites. This includes the creation of semantically correct XHTML and CSS based design using web standards.

The functional aspects focus on the creation of websites using good information architecture and that demonstrate best practice in accessibility, usability and findability. Students develop an understanding of how people use the web, how they like to navigate and search for information and the importance of SEO. See [Design principles](#) for more information.

The aesthetic aspects of web design are considered with regard to both graphic/artistic and business objectives, such as corporate identity and branding. The architectural principle that form should follow function is relevant, not as a rule but because aesthetic considerations must be judged with regard to the technical and functional aspects of a website.

**Flexible teaching mode**

The timetable is designed to make it easy for those in full-time employment to attend the programme and students have the option of a full-time (one day per week) or part-time (one half day per week) mode of study. Although contact time is relatively short, students should be aware that there will be a substantial amount of course work, project work and reading to be completed in their own time. See [Programme details](#) for more information.

We believe our programme is unique in offering a balanced blend of the technical, functional and aesthetic aspects of web design and we are the only university in London offering such a focused programme at masters level.

**Need to know more?**

If after reading through this website, you still have some questions about the MA Web Design & Content Planning programme, please get in touch using our [contact form](#) and we'll do our best to answer your questions.

**Programme details**  
[Core courses](#)  
[Our philosophy](#)  
[Design principles](#)  
[Our students](#)  
[Frequent questions](#)  
[Web design bookshelf](#)  
[Contact](#)

MA Web Design & Content Planning (full-time and part-time modes) now recruiting for September 2010  
[Find out more...](#)

**Redesign this site**

Select one of the colour options below to change the way this site looks. How does this work?

**Frequently asked questions**

- Will we learn how to use Dreamweaver?
- At what time of year does the programme begin?
- What do students do after graduation?
- How much study time will I need each week?
- Where are lectures held?
- Where can I get financial advice?

[More FAQs...](#)

**Student opinion**

Most students tell us that the programme has exceeded their expectations. See [Our students](#).

**Latest forum topics**

- Aha!ISP Redesign
- New 5 for 1 offer at Sit...
- SEO for Universal Search
- Smashing Magazine on the...
- WordPress
- WordPress & Itaccess
- Careers
- 53 CSS-based techniques
- Could this campaign see...
- Should the BBC dump half...

[More forum topics...](#)

**Site Navigation**  
[Site homepage](#)  
[Programme details](#)  
[Core courses](#)  
[Our philosophy](#)  
[Design principles](#)  
[Our students](#)  
[Frequent questions](#)  
[Web design bookshelf](#)  
[Contact](#)

**Our Favourite Websites**  
[A List Apart](#)  
[Digital Web Magazine](#)  
[Smashing Magazine](#)  
[456 Berea Street](#)  
[Think Vitamin](#)  
[Added Bytes](#)  
[i love typography](#)  
[Sitepoint](#)  
[zeldman.com](#)

**University of Greenwich**  
[UoG homepage](#)  
[School of Architecture & Construction](#)  
[Avery Hill Campus](#)  
[Postgraduate prospectus](#)  
[A guide to postgraduate study](#)  
[How to apply](#)  
[Tuition fees](#)  
[Contact details](#)  
[UoG Students' Union](#)

Copyright © 2003 – 2010 David Watson & Tom Turner | [XHTML](#) | [CSS](#) | [Accessibility](#) | [Site Map](#)

# Include file naming

Parent file = index.php

Most commonly, includes are given a “.inc.php” extension (to differentiate from the parent .php files) and then stored in a folder called “includes”. As with other files that make up your site, it’s important to give includes a logical name, usually this will describe what they are or what they do. In most cases, SEO is not a consideration because include files alone are not navigable. In fact, most designers will disallow indexing of the includes folder using the robots.txt file:

User-agent: \*

Disallow: /includes/

Keep your includes away from prying eyes.

The screenshot shows a website layout with several red arrows pointing to specific areas, each labeled with an include file name:

- header.inc.php**: Points to the top navigation bar containing the site title "MA Web Design & Content Planning".
- sidebar.inc.php**: Points to the right-hand sidebar area containing a list of links (Programme details, Core courses, etc.), a "Redesign this site" section, and a "Frequently asked questions" section.
- footer.inc.php**: Points to the bottom of the page, which contains a "Site Navigation" menu, "Our Favourite Websites", "University of Greenwich" information, and a copyright notice.

The website content includes a main heading "Study web design in London at the University of Greenwich", a paragraph about the MA Web Design & Content Planning Programme, and sections for "Learning web design", "Flexible teaching mode", and "Need to know more?".

# Include files are just HTML fragments!

```
<header>
  <p class="site_logo">Modular web design</p>
</header>
```

header.inc.php



```
<nav>
  <ul>
    <li><a href="#">First navigation link</a></li>
    <li><a href="#">Second navigation link</a></li>
    <li><a href="#">Third navigation link</a></li>
  </ul>
</nav>
```

navigation.inc.php



```
<footer>
  <p>&copy; David Watson <?php echo date("Y"); ?></p>
</footer>
```

footer.inc.php



# Not fully-formed HTML files

```
<header>
  <p class="site_logo">Modular web design</p>
</header>
```

header.inc.php

```
<nav>
  <ul>
    <li><a href="#">First navigation link</a></li>
    <li><a href="#">Second navigation link</a></li>
    <li><a href="#">Third navigation link</a></li>
  </ul>
</nav>
```

navigation.inc.php

```
<footer>
  <p>&copy; David Watson <?php echo date("Y"); ?></p>
</footer>
```

footer.inc.php

Note that include files contain only fragments of HTML, they are not fully-formed HTML documents. There is no doctype declaration, no head element or body.

Whatever is in the include file will be added to the parent file as is. Don't forget about indentation!

Note: the reason we give include files a **.php** extension is to ensure that they are sent to the php interpreter so that any scripts they contain will also be parsed. See **footer.inc.php** for example.

# The fragments are added to the parent file

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>A Modular Webpage | PHP Example</title>
    <link href="style/main.css" rel="stylesheet">
  </head>

  <body>
    <?php
    include($_SERVER['DOCUMENT_ROOT'].'/includes/header.inc.php');
    include($_SERVER['DOCUMENT_ROOT'].'/includes/navigation.inc.php');
    ?>

    <main>
      <!-- Content Begin -->

      <h1>PHP example using server-side includes</h1>
      <p>Some body text... </p>

      <!-- Content End -->
    </main>

    <?php
    include($_SERVER['DOCUMENT_ROOT'].'/includes/footer.inc.php');
    ?>
  </body>
</html>
```

index.php



The included fragments of HTML simply replace the include statement when the file is parsed on the server. The result is a complete HTML file, built from four separate parts.

# When the file reaches the browser...

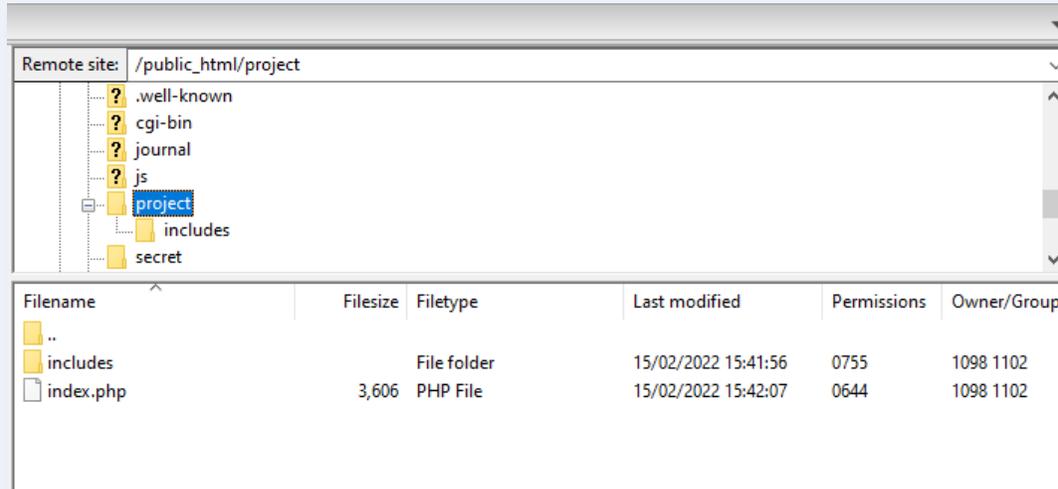
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>A Modular Webpage | PHP Example</title>
    <link href="style/main.css" rel="stylesheet">
  </head>
  <body>
    <header>
      <p class="site_logo">Modular web design</p>
    </header>
    <nav>
      <ul>
        <li><a href="#">First navigation link</a></li>
        <li><a href="#">Second navigation link</a></li>
        <li><a href="#">Third navigation link</a></li>
      </ul>
    </nav>
    <main>
      <!-- Content Begin -->
      <h1>PHP example using server-side includes</h1>
      <p>Some body text... </p>
      <!-- Content End -->
    </main>
    <footer>
      <p>&copy; David Watson 2023</p>
    </footer>
  </body>
</html>
```

index.php



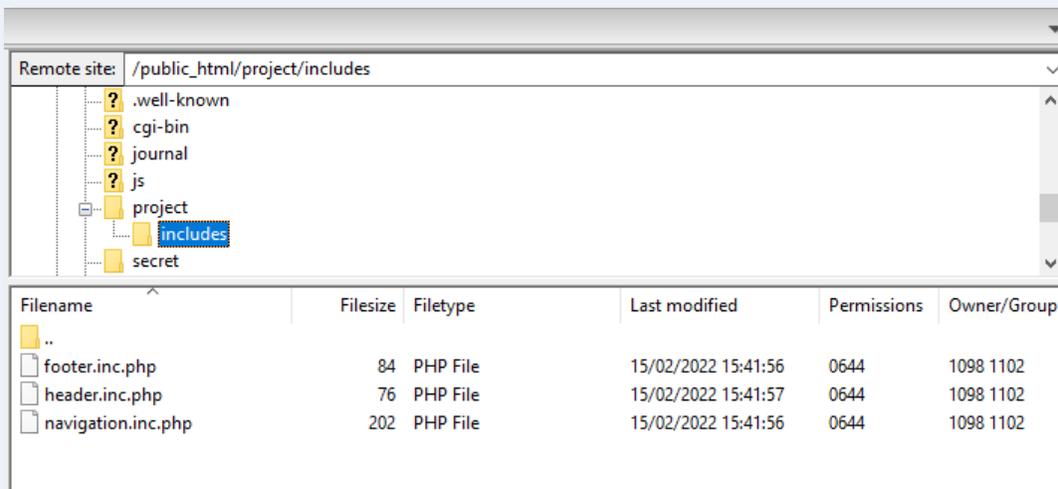
The index.php file is transformed by the PHP interpreter from a file containing a mixture of HTML and PHP to a file containing only HTML, and the transformed file is sent to the browser.

# Folder organisation



Remote site: /public\_html/project

Filename	Filesize	Filetype	Last modified	Permissions	Owner/Group
..					
includes		File folder	15/02/2022 15:41:56	0755	1098 1102
index.php	3,606	PHP File	15/02/2022 15:42:07	0644	1098 1102



Remote site: /public\_html/project/includes

Filename	Filesize	Filetype	Last modified	Permissions	Owner/Group
..					
footer.inc.php	84	PHP File	15/02/2022 15:41:56	0644	1098 1102
header.inc.php	76	PHP File	15/02/2022 15:41:57	0644	1098 1102
navigation.inc.php	202	PHP File	15/02/2022 15:41:56	0644	1098 1102

On the server, you might choose to have an include folder for each project where all your PHP include files are stored. This keeps things nice and tidy.

You could then add any of the include files in any other document in your project (irrespective of their location) using a statement in the following form:

```
<?php  
include $_SERVER['DOCUMENT_ROOT'] .  
'/project/includes/footer.inc.php';  
?>
```

You may sometimes see the include path in parenthesis but include will work either way, so it is usually written without parenthesis.

# Server-side include paths

## A relative path:

```
<?php include 'includes/footer.inc.php'; ?>
```

The problem with using relative paths for includes is that they change depending on where in the site the parent file is located.

## An absolute path:

```
<?php include '/home/mydomain/public_html/includes/footer.inc.php'; ?>
```

PHP can see above the document root so, unlike HTML, PHP absolute paths begin at the topmost server level and not at the document root. Notice that the name of the domain (or the domain id) is included in an absolute path.

## A “portable” absolute path:

```
<?php include $_SERVER['DOCUMENT_ROOT'] . '/includes/footer.inc.php'; ?>
```

In order to make absolute paths “portable” (they work on different servers or if the domain changes), use the `$_SERVER` *superglobal* array as in the example above.

# Server-side include paths

The `$_SERVER['DOCUMENT_ROOT']` superglobal array element contains the path from the server root to the document root (amongst many other things). This will be something like `"/home/mydomain/public_html"`. The period `"."` means append (add) the text that follows, so the result is the full path from the server root to the specified file.

```
<?php include $_SERVER['DOCUMENT_ROOT'] . '/includes/footer.inc.php'; ?>
```

You may sometimes see the `require` function being used in place of `include`:

```
<?php require $_SERVER['DOCUMENT_ROOT'] . '/includes/footer.inc.php'; ?>
```

The `include` and `require` functions are very similar. The difference is that `require` will stop processing the document if the file is not found and `include` will continue – or attempt to continue.

# Minimising repetition

The `$_SERVER['DOCUMENT_ROOT']` superglobal array element is a useful way to ensure our paths are robust, but is it necessary to use it with every include statement?

No, if there are several include statements in the same file, it may be more efficient to use the `set_include_path` function, before your include statements, like this:

```
<?php set_include_path ( $_SERVER[ 'DOCUMENT_ROOT' ] ); ?>
```

Then you can write include statements like this:

```
<?php include 'includes/header.inc.php'; ?>
```

...some HTML here.

```
<?php include 'includes/footer.inc.php'; ?>
```

[set\\_include\\_path](#)

Notice that there is no leading slash in the path because the `set_include_path` function has already made sure that we will be starting at web root.

If you only ever learn to do one thing with PHP, learn to use server-side includes; it will revolutionise the way you build websites, and you'll never regret it (honest).

# Managing repeated content

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>PHP Example</title>
    <link href="/style/style.css" rel="stylesheet">
  </head>

  <body>
    <?php
include $_SERVER['DOCUMENT_ROOT'].'/includes/header.inc.php';
include $_SERVER['DOCUMENT_ROOT'].'/includes/navigation.inc.php';
?>

    <main>

      <!-- Content Begin -->

      <h1>PHP Example</h1>
      <p>Some body text... </p>

      <!-- Content End -->

    </main>
    <?php include $_SERVER['DOCUMENT_ROOT'].'/includes/footer.inc.php'; ?>
  </body>
</html>
```

In this file, the text "PHP Example" is used for the <title> and <h1> elements and there could be other instances of the same text on this page. Managing content like this when consistency is important can be difficult.

# Variables in PHP

Assigning a value to a variable

```
<?php $this_page = "PHP Example"; ?>
```

Printing a variable

```
<?php echo $this_page; ?>
```

The output

PHP Example

[PHP variables](#)

Notice that, unlike JavaScript, we don't need to use a keyword to declare a variable (var, let, const). We simply state the name of the variable (with the \$ prefix) and PHP understands what we want to do.

# Using variables for consistency

```
<?php $this_page = "PHP Example"; ?>
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title> <?php echo $this_page; ?> </title>
    <link href="/style/style.css" rel="stylesheet">
  </head>

  <body>
    <?php
    include $_SERVER['DOCUMENT_ROOT'].'/includes/header.inc.php';
    include $_SERVER['DOCUMENT_ROOT'].'/includes/navigation.inc.php';
    ?>

    <main>

      <!-- Content Begin -->

      <h1> <?php echo $this_page; ?> </h1>
      <p>Some body text... </p>

      <!-- Content End -->

    </main>
    <?php include $_SERVER['DOCUMENT_ROOT'].'/includes/footer.inc.php'; ?>
  </body>
</html>
```

In this file, the repeated text elements are all managed using a PHP variable. The text is assigned to a variable at the top of the file and the echo function is used to print the value of the variable, whenever that text is required.

In this example, we've created a variable called `$this_page`. All variable names in PHP begin with the dollar sign “\$”.

The string value “PHP Example” is assigned to the variable.

Note that PHP statements can be placed anywhere in the file – they don't have to be inside the `<html>` element.

# The echo statement shorthand

The echo statement

```
<?php echo $this_page; ?>
```

The echo shorthand

```
<?= $this_page; ?>
```

The output

## PHP Example

[The echo language construct](#)

In PHP, as in other languages, brevity is a good thing. The echo shorthand provides a neater way to print a variable to a webpage. As always, there are different opinions on whether using the echo shorthand is a good idea or not. I show it here so you know it exists and will understand it when you see it.

# Keeping things up to date

There's no need to interrupt your New Year's Eve party in order to update the copyright notice on your website; just let PHP do it for you.

```
<footer>  
    <p>Copyright David Watson <?php echo date("Y"); ?></p>  
</footer>
```

This footer will always display a copyright notice with the current year.

Do bear in mind that the PHP `date` function uses server time and not local time. If local time is important, consider using JavaScript instead.

[The date function](#)

# Mixing PHP with HTML?

```
<h1> <?php echo $this_page; ?> </h1>  
<p>Some body text... </p>
```

Notice that in the examples we've seen so far, all our PHP is mixed up with the HTML. Surely that's wrong, and we should keep them separate like we do with JavaScript and CSS?

Actually, it doesn't matter because the browser never sees any of this, it only sees the compiled HTML. This may initially feel wrong if you follow the Web Standards Model, but you'll get used to it.

A little bit of PHP goes a very long way – it makes the management of larger sites very much easier than HTML alone; and there's plenty more it can do, like smart navigation...

# A navigation include file

```
<nav>
  <ul>
    <li><a href="php-example.php">PHP Example</a></li>
    <li><a href="another-link.php">Another Link</a></li>
    <li>...
      ...</li>
  </ul>
</nav>
```

The markup fragment (above) is saved in a file called navigation.inc.php. Whenever this file is included, using the statement below, the navigation markup will appear in its place once the PHP is parsed and passed to the browser.

```
<?php include $_SERVER['DOCUMENT_ROOT'].'/includes/navigation.inc.php'; ?>
```

Include files don't need a doctype declaration, a head section, they are just fragments of HTML. They only work as includes, not stand-alone web pages.

# The included navigation

```
<nav>
  <ul>
    <li><a href="php-example.php">PHP Example</a></li>
    <li><a href="another-link.php">Another Link</a></li>
    <li>...
      ...</li>
  </ul>
</nav>
```

This included markup will display the same navigation links on every page – great for ease of site management because if we want to add another link, we just add it to navigation.inc.php and it will display across the entire site.

However, what if we want the navigation to indicate which page we're on? We don't want a live link to the PHP Example page if we're already on that page. PHP to the rescue...

This include is just plain HTML and will display the same on all pages.

# Adding some intelligence

...We can get PHP to decide which page we're on and then add a link or not.

In our example page, we assigned the page title to a variable like this:

```
$this_page = "PHP Example";
```

We can use this information to add some intelligence to our navigation include:

```
<ul>
<?php
if ($this_page == "PHP Example") {
    echo "<li>PHP Example</li>";
}else{
    echo "<li><a href=\"php-example.php\">PHP Example</a></li>";
}
?>
    <li><a href="another-link.php">Another Link</a></li>
    <li>...
        ...</li>
</ul>
```

We're using an if else conditional statement to make a decision based on the value of a variable.

[The if control structure](#)

# Site-wide intelligent navigation

```
<ul id="navigation">
<?php
if ($this_page == "PHP Example"){
    echo "<li>PHP Example</li>";
}else{
    echo "<li><a href=\"php-example.php\">PHP Example</a></li>";
}
if ($this_page == "Another Page"){
    echo "<li>Another Link</li>";
}else{
    echo "<li><a href=\"another-page.php\">Another Link</a></li>";
}
if ($this_page == "A Third Page"){
    echo "<li>A Third Link</li>";
}else{
    echo "<li><a href=\"a-third-page.php\">A Third Link</a></li>";
}
?>
</ul>
```

We can add the same conditional logic to all our links – assuming `$this_page` is defined.

In fact, we could do this without using `$this_page` because the name of the current page is held in the `$_SERVER` superglobal array...  
...but let's not get ahead of ourselves.

You may have noticed that PHP can be used to do many of the same things that JavaScript can do. However, using PHP is a more robust strategy because it doesn't rely on the client.

Here are just a couple of  
really useful things you  
should know before you  
start using PHP...

# Using quotes

We can print text in PHP using the `echo` function like this:

```
echo "David likes PHP";
```

We can also print HTML using the same function:

```
echo "<p>David likes PHP</p>";
```

But a problem arises when the text or HTML we are printing includes quotes:

```
echo "<p class=\"content\">David likes PHP</p>";
```

This will cause an error because `echo` thinks the string ends after the second double quote. There are 2 ways around the problem:

*Escape* the double quotes with the backslash character:

```
echo "<p class=\"content\">David likes PHP</p>";
```

Or use single quotes:

```
echo '<p class="content">David likes PHP</p>';
```

Both the above options give the same result:

```
<p class="content">David likes PHP</p>
```

This is *really important* and is the cause of much anguish for the PHP beginner.

# ...and another thing about quotes

*Variables in double quoted strings are evaluated but variables in single quoted strings are not.*

So the following script:

```
$my_name = "David";  
echo "$my_name likes PHP"; # double quotes
```

Returns the following text:

David likes PHP

While this script:

```
$my_name = "David";  
echo '$my_name likes PHP'; # single quotes
```

Returns this text:

\$my\_name likes PHP

Escaping double quotes is usually a better strategy because you usually want variables to be evaluated.

[Single quotes vs Double quotes in PHP](#)

# Comments in PHP scripts

```
<?php
# assign the name of the day to a variable
$today = date(1);
# check the value of the variable against a number of options
switch ($today) {
    # if today is Monday, assign the image and alt variables
    case "Monday":
        $image = "image-01.jpg";
        $alt = "A Blackbird";
        break;
    # if today is Tuesday, assign the image and alt variables
    case "Tuesday":
        $image = "image-02.jpg";
        $alt = "A Blue Tit";
        break;
}
```

It's always a good idea to add comments to a script so that you can see what it's doing – or what it's supposed to do. Single line comments begin with the hash symbol (#).

# More comments

```
<?php
// assign the name of the day to a variable
$today = date(1);
/* use a switch to check the value of the variable against
   a number of options, one for each day if the week */
switch ($today) {
    // if today is Monday
    case "Monday":
        $image = "image-01.jpg";
        $alt = "A Blackbird";
        break;
    // if today is Tuesday
    case "Tuesday":
        $image = "image-02.jpg";
        $alt = "A Blue Tit";
        break;
}
```

PHP is extraordinarily well endowed when it comes to commenting. In addition to the hash symbol (#), single line comments can be made with a double-slash (//), like JS. Multi-line comments use the same syntax as CSS and JS; start with /\* and end with \*/.

# Learning PHP

The screenshot shows the Codecademy interface for learning PHP. At the top, there is a navigation bar with 'Codecademy', 'Learn', and 'Teach' tabs. On the right side of the navigation bar, there is a notification bell icon, a progress indicator showing '1 point today' and '1 day streak', and a user profile icon labeled 'Me'. The main content area is divided into two columns. The left column features a large 'P' in a circle, the title 'PHP', a brief description: 'PHP is the most popular server-side language and it requires minimal set-up.', and a blue 'Continue (1%)' button. Below this are two buttons for 'HTML Glossary' and 'CSS Glossary'. The right column is titled 'Welcome to PHP!' and contains three tutorial sections: 1. 'Introduction to PHP' (with a blue arrow icon), 2. 'Control Flow: If/Else' (with a radio button icon), and 3. 'Control Flow: Switch' (with a radio button icon). Each section includes a short introductory paragraph.

**Codecademy** Learn Teach

1 point today  
1 day streak

Me

## Welcome to PHP!

**1. Introduction to PHP**

This tutorial will introduce you to PHP, a server-side scripting language you can use to make dynamic websites and web applications.

### Control Flow: If/Else

**2. Conditionals and Control Flow**

Control flow is how a program decides to perform one action instead of another. In PHP, we can use 'if' and 'else' to do this!

### Control Flow: Switch

**3. Control Flow: Switch**

The switch statement lets you perform more complex control flow operations without using an endless chain of 'if's and 'else's'.

**PHP**

PHP is the most popular server-side language and it requires minimal set-up.

[Continue \(1%\)](#)

[HTML Glossary](#)

[CSS Glossary](#)

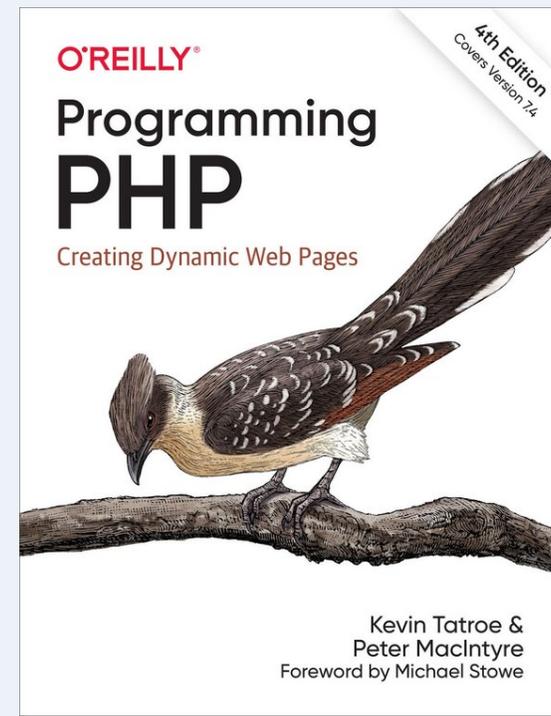
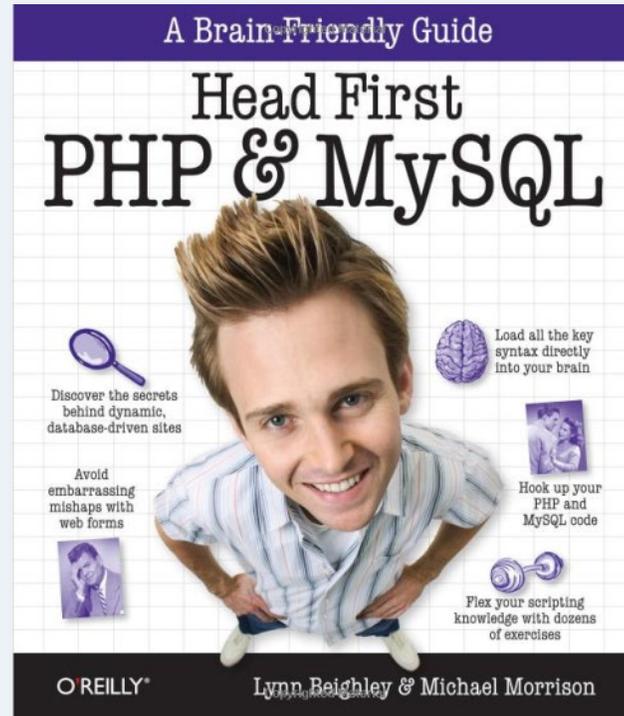
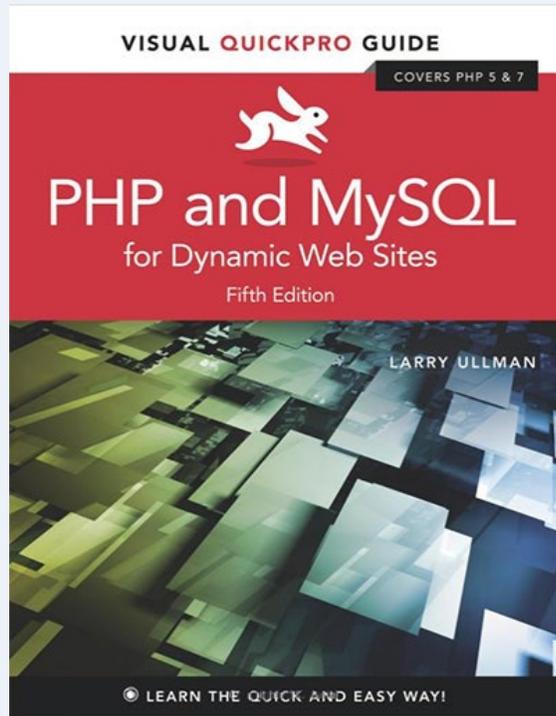
There are lots of online tutorials available for PHP, including some useful YouTube videos but the [Codecademy Introduction to PHP](#) is a particularly good place to begin..

# The long awaited...

Published 14<sup>th</sup> February 2022



# Recommended books



The Larry Ullman book is designed for web designers who want to build dynamic websites and as such is a good match for us. Some will find the Head First book a little easier to get into if you're new to programming and the O'Reilly book gives a more structured approach for those who want to become more expert with PHP. All are available from the library.

# The PHP Manual

The screenshot displays the PHP Manual website interface. At the top, there is a navigation bar with the PHP logo and links for Downloads, Documentation, Get Involved, and Help. A search box is located on the right side of the navigation bar. Below the navigation bar, the page title is 'PHP Manual', with navigation links for « Preface and Introduction ».

The main content area is divided into two columns. The left column contains a table of contents with links to various sections: PHP Manual, Copyright, PHP Manual, » Getting Started, Installation and Configuration, Language Reference, Security, Features, Function Reference, PHP at the Core: A Hacker's Guide, FAQ, and Appendices. The right column is titled 'Getting Started' and contains a list of links: Introduction, What is PHP?, What can PHP do?, A simple tutorial, What do I need?, Your first PHP-enabled page, Something Useful, Dealing with Forms, Using old code with new versions of PHP, and What's next?. There is also a 'User Contributed Notes' section with an 'add a note' button. A language dropdown menu is set to 'English' with 'Edit' and 'Report a Bug' links nearby.

At the bottom of the page, there is a footer with copyright information: Copyright © 2001-2014 The PHP Group, and links to My PHP.net, Contact, Other PHP.net sites, Mirror sites, and Privacy policy.

There are many online references for PHP but the PHP Manual is the one definitive reference, written by the makers of PHP; it includes a useful [Getting Started](#) section.

```
<?php echo $end ?>
```