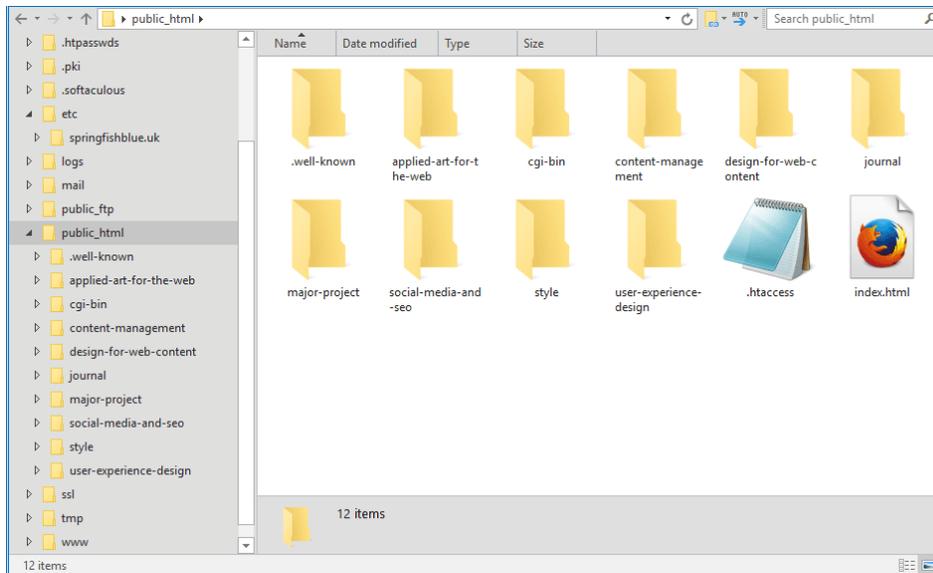


File and folder organisation

Web design requires very careful organisation of your files and folders because most sites are made up from hundreds or thousands of individual files. Developing such sites can become very difficult if you don't have a file and folder strategy. The information below describes how you *might* want to organise the files and folders on your coursework website.

public_html



Above is a typical view of the [public_html](#) folder (sometimes referred to as the *web root folder*) for your coursework website viewed in a FTP client. All files in the web root are publicly available over the Web, that's why it's called [public_html](#). Files and folders *above* the web root are not publicly available. There are several files and folders in this view. The [index.html](#) file is the homepage for this domain. Typically, this will show a list of all your coursework when displayed in a browser. The [style](#) folder contains the CSS files for the homepage.

Each course has its own folder, and this is the first level of organisation. So, for example, all the coursework you have done for the Design for Web Content course will be placed in the [design-for-web-content](#) folder. Note that all file and folder names are in lower-case and the hyphen character is used in place of spaces between words. This is a general rule you should follow.

In addition to the 6 course folders, you may see some others. In the view above, there is a folder called [journal](#). That's the folder where the WordPress blog is installed (yours may be called something different). You may look at the files in that folder but don't change anything or your blog may not function correctly.

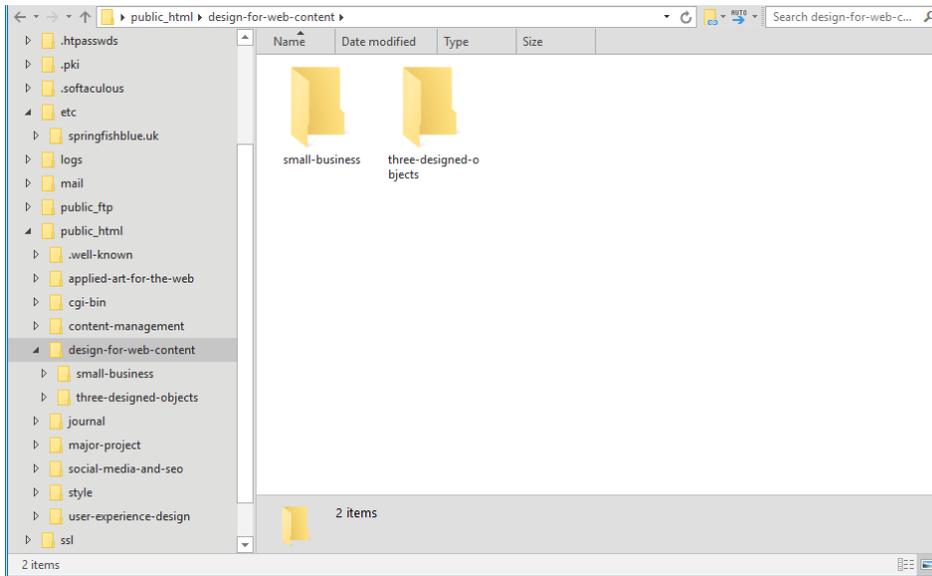
The [.well-known](#) folder is required for the SSL certificate, which is used for https encryption. The folder is empty but must remain in your web root folder.

The [cgi-bin](#) folder is also empty and is not usually required for modern websites. This folder may be deleted if you wish.

Finally, the `.htaccess` file performs various functions and should not be deleted. We will learn more about this file in due course. Don't worry if your root folder doesn't have a `.htaccess` file.

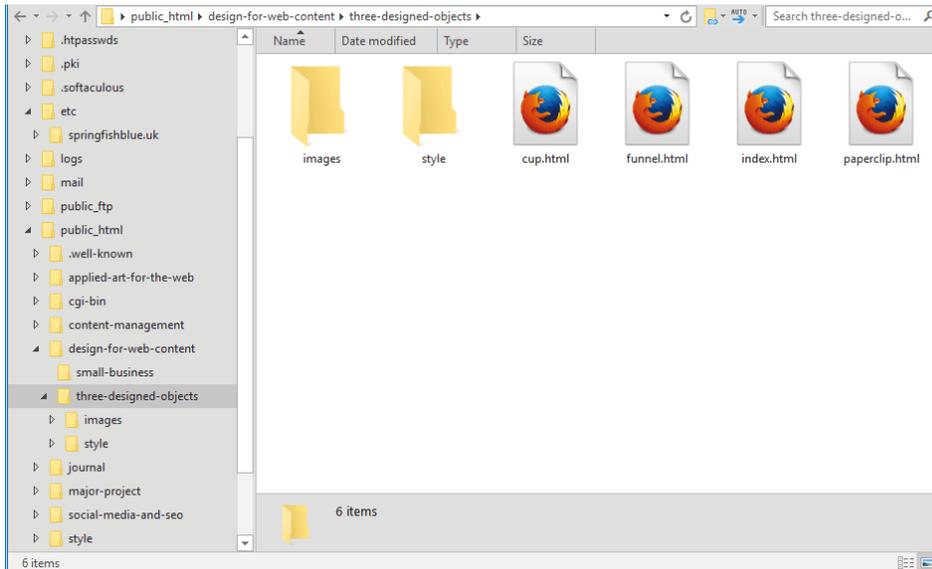
Although we are looking at the file and folder structure on the web server, using FTP, the arrangement of files and folders should be identical on your laptop. In effect, what you see on your web server should be a copy of what you see locally (see page 6 for more details).

public_html / design-for-web-content



This view shows the contents of the `design-for-web-content` folder. It contains just two folders, one for each of the two projects in this course. There is no need for an `index.html` file here because the projects will be linked to from the domain homepage we saw in `public_html`.

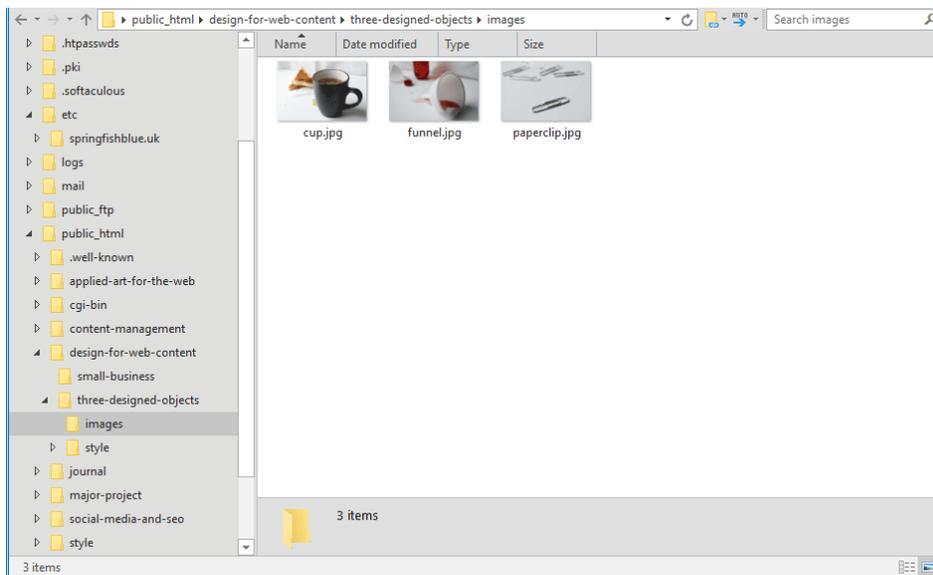
public_html / design-for-web-content / three-designed-objects



This view shows the contents of the *three-designed-objects* folder (we could call this the *project* folder). The brief for this project specifies that you should create a small, 4-page site. The four *.html* files are those four pages, with *index.html* being the homepage. The other *.html* files will display the information about each of the three designed objects. The *index.html* file will contain links to the other three files so that they form a website of linked pages. Think of each project as being a different website. Each website needs its own folder and sub-folders.

In addition to the *.html* files in this project folder, there are two sub-folders, *images* and *style...*

[public_html / design-for-web-content / three-designed-objects / images](#)



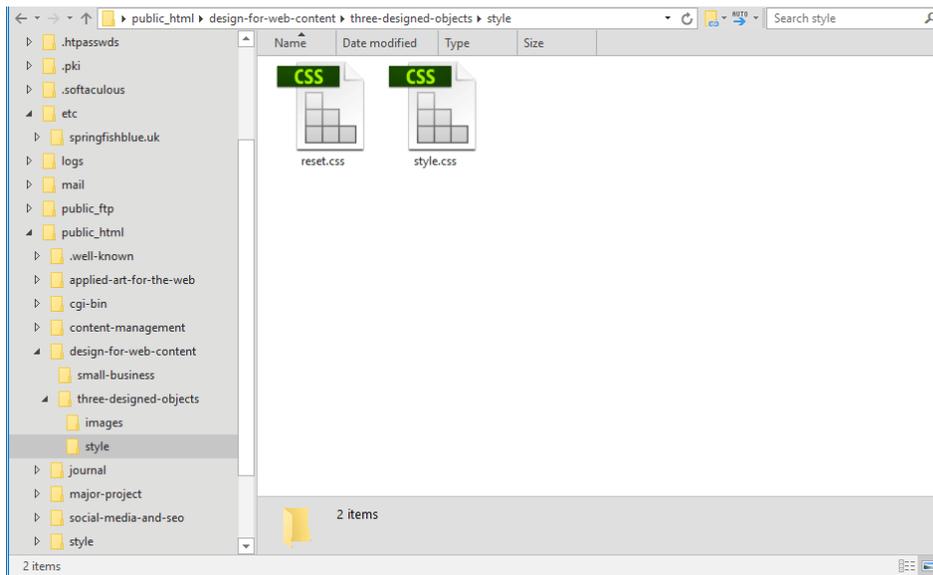
This is the view of the *images* folder, it contains three *.jpg* files. These are the images that will be shown on each of the corresponding *.html* files when viewed in a browser. Typically, images are stored in their own folder, usually called *images*, but you could call it something else if you wish.

```

```

The images in the *images* folder can be displayed in the *.html* files using the ** element. Notice that the value for the *src* attribute is the path from the *.html* file to the image file, so it includes the name of the folder, “*images*” as well as the name of the file.

public_html / design-for-web-content / three-designed-objects / style



This is the view of the [style](#) folder. Some people prefer to keep all their style-related files in a separate folder like this, but it isn't always necessary. For example, you may have only a single [.css](#) file for a small project, consisting of just a few webpages. In that case, you may choose to store the [style.css](#) file in the main project folder along with [index.html](#).

In the view above, there are two [.css](#) files. The [reset.css](#) file is our reset file and [style.css](#) contains our own CSS rules that describe how our 4-page website will look. Both files are linked to from the `<head>` section of the four [.html](#) files in the project folder. Your project may only have a single [.css](#) file that contains both the reset and your style rules (usually this is called [style.css](#) but it could be called anything, providing it has a [.css](#) extension).

```
<head>
  <meta charset="UTF-8">
  <title>Paper clip | Three Designed Objects</title>
  <link rel="stylesheet" href="style/reset.css">
  <link rel="stylesheet" href="style/style.css">
</head>
```

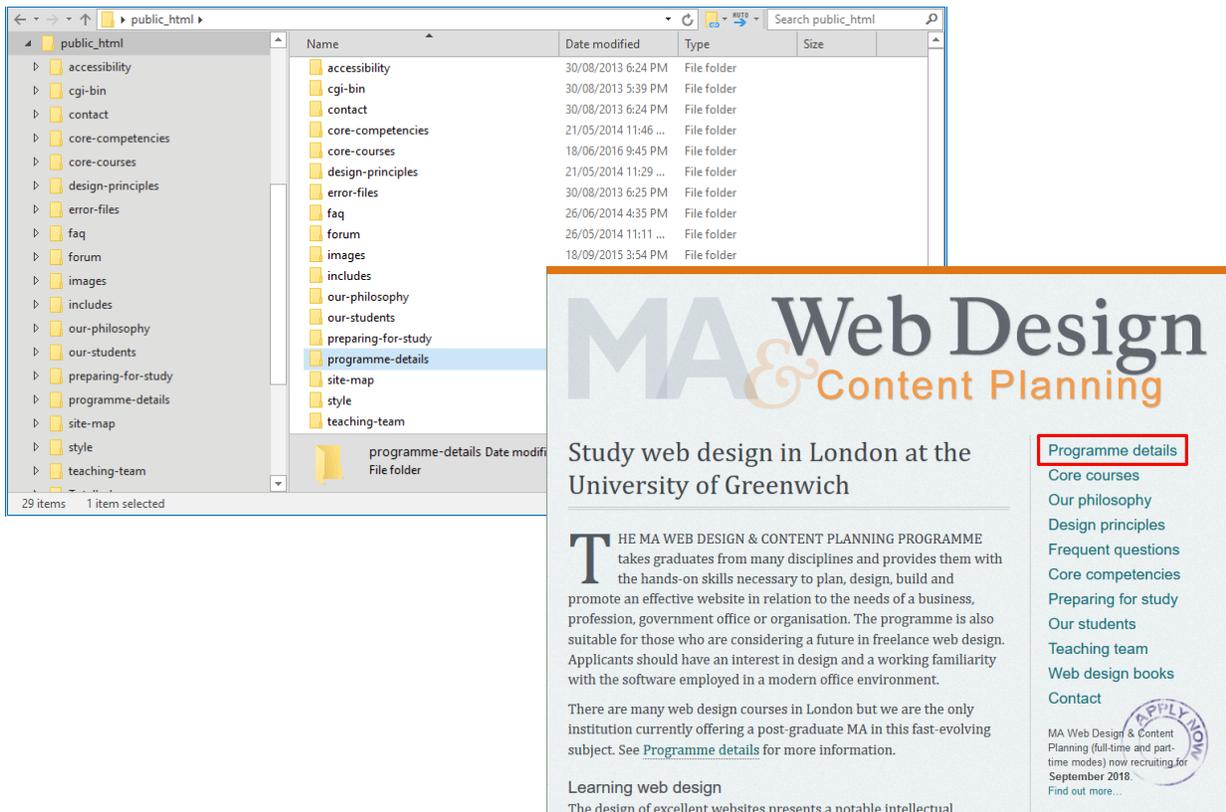
The `<head>` section of each of the [.html](#) files will look something like the code snippet shown above. Note that the value of each `href` attribute is the path from the [.html](#) file to the [.css](#) file, so it includes the name of the folder "style" as well as the name of the files.

Each project will have its own [.css](#) file(s) and its own [images](#) folder.

Structure and navigation (information architecture)

Website structures vary a lot, so you always need to work out how best to organise your files and folders for any given project. The example above will work well for your coursework site, but you may need to use a different approach depending on the nature and complexity of the website you are building. In general, you should always attempt to devise a system for organising your folders that is consistent with the primary navigation of your site. This is often referred to as *information architecture*.

In the example below, you can see that there is one folder in the web root corresponding to each link in the primary navigation. For example, all files used for the “Programme details” section of the site are stored in the *programme-details* folder.



The image shows a file explorer window on the left and a website screenshot on the right. The file explorer shows a directory structure for a website, with the 'programme-details' folder highlighted. The website screenshot shows the 'MA Web Design & Content Planning' page, with the 'Programme details' link highlighted in the navigation menu.

Name	Date modified	Type	Size
accessibility	30/08/2013 6:24 PM	File folder	
cgi-bin	30/08/2013 5:39 PM	File folder	
contact	30/08/2013 6:24 PM	File folder	
core-competencies	21/05/2014 11:46 ...	File folder	
core-courses	18/06/2016 9:45 PM	File folder	
design-principles	21/05/2014 11:29 ...	File folder	
error-files	30/08/2013 6:25 PM	File folder	
faq	26/06/2014 4:35 PM	File folder	
forum	26/05/2014 11:11 ...	File folder	
images	18/09/2015 3:54 PM	File folder	
includes			
our-philosophy			
our-students			
preparing-for-study			
programme-details			
site-map			
style			
teaching-team			

MA Web Design & Content Planning

Study web design in London at the University of Greenwich

[Programme details](#)

- Core courses
- Our philosophy
- Design principles
- Frequent questions
- Core competencies
- Preparing for study
- Our students
- Teaching team
- Web design books
- Contact

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.

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. See [Programme details](#) for more information.

Learning web design
The design of excellent websites presents a notable intellectual

MA Web Design & Content Planning (full-time and part-time modes) now recruiting for September 2018. Find out more...

Maintaining a local "mirror" and backing up

Whatever scheme you devise for your files and folders, it's very important to maintain a local mirror of your online files. This will allow you to edit your files locally and upload them only when you are confident everything is working correctly. Always keep your local files synchronised with your online files (hence "mirror").

In addition to the local mirror of your work, you should also keep a separate backup of your files and folders. Ideally, this is on an external drive or a cloud drive (such as Dropbox) or both – you can't have too many backups.

In the event of your laptop being damaged or stolen or of your local and online files being accidentally deleted or overwritten, you always have a saved version of your work elsewhere. Ideally, take a backup of your work at least once per week or at the end of each day if you are making lots of changes. I promise you will never regret doing this.

Your web host will also take backups of your online files, but you can't always rely on the fact that they will have the version of your files that you need, so take responsibility for your own backups.

October 2020