

Setting up a Web Site within the Department

These actions must be done on the student Linux Farm systems,

There are two main types of web sites; static or dynamic.

- Static web sites, usually HTML documents, but also includes other formats e.g. PDF documents. It also includes web pages that utilise client side scripting (program code that is processed by the client system, e.g. embedded javascript in an HTML document).
- Dynamic web sites, usually referred to as CGI pages, are server side scripts or programs that are processed by the web server to generate the web page content.
- On the Computer Science servers the URLs used to access static and dynamic pages are formatted differently and accessing (linking to) dynamic content from a static web page or vice-versa, must use the appropriate full URL rather than relative folder paths.
e.g. don't use ``

Our departmental web servers are case sensitive, i.e. MyFile.eg is different to myfile.eg

Static Content

From within a Linux session create the web page folder and set its permissions so that the Web server can read the files.

```
mkdir public_html  
chmod a+rx public_html
```

Create web pages within the public_html folder then set their permissions so that the Web server can read them.

```
cd public_html  
chmod a+r <filename>
```

where <filename> is the name of the web page file.

By default the web server will open “index.html”, “index.htm”, or “index.php”, so usually best if you name your main page “index.html” (for static web content) or “index.php” (for dynamic content generated using PHP scripts)

The Web site can be accessed by using the URL

<https://student.csc.liv.ac.uk/~<username>/<filename>>

or (if you want to open index.html)

<https://student.csc.liv.ac.uk/~username/>

where <username> is your Computer Science computer username.

E.g. if the username is sgabcde, use

<https://student.csc.liv.ac.uk/~sgabcde/>

Dynamic Content

If you want to create CGI web content then you will need to create a cgi-bin folder and set the permissions so that the Web server can read them.

```
mkdir public_html/cgi-bin  
chmod a+rx public_html/cgi-bin
```

Create the pages to be served within this folder and set the permissions.

```
chmod a+rx public_html/cgi-bin/<filename>
```

The Web site can be accessed by using the URL

<https://student.csc.liv.ac.uk/cgi-bin/cgiwrap/<username>/<filename>>

where <username> is your Computer Science computer username.

E.g. if the username is sgabcde and the script is MyScript.pl, use

<https://student.csc.liv.ac.uk/cgi-bin/cgiwrap/sgabcde/MyScript.pl>

Note:

PHP scripts are handled differently by the web server when compared to other dynamically generated content (e.g. Python, PERL, or javascript) and the PHP files do not need to be located in the cgi-bin folder. Just locate them in the public_html folder.

Permissions

- Linux permissions (as used by the Web server) have three components: Owner, Group, World.
- Each of these can have three levels of access: Write, Read, Execute
- The permissions are represented as a 9 character string and stored as a 9 digit binary number with each digit representing a component permission as follows:

Owner (u)			Group (g)			World (o)		
Read (r)	Write (w)	Execute (x)	Read (r)	Write (w)	Execute (x)	Read (r)	Write (w)	Execute (x)

So if the owner has full permissions, but no-one else has access this would be represented as:

rwX----- or 111000000

If the owner has full permissions and the owner's security group has Read:

rwXr----- or 111100000

If owner has Read & Write and everyone has Read:

rw-r--r-- or 110100100

The components are usually represented by a three part decimal equivalent of the component binary numbers, e.g. rwXr--r-- would be 744 (111100100 in binary notation)

- To assign permissions use 'chmod'.
- With chmod you can use "u", "g", "o" to specify the component, or "a" to specify ALL.
You can add permission using "+", e.g. chmod a+r <filename>
You can remove permissions using "-", e.g. chmod a-x <filename>
You can set permissions absolutely, e.g. for rwXr-xr-x use;
chmod 755 <filename>
- **The Web server will access the Web folder as a World user.**
- All web site folders should have World Execute permissions, e.g. rwX-----x
- **The Web server will access the Web page file as the Owner.**
- All Static files should have World Read, e.g. rwX---r-- (111000100)
- All application (dynamic) files should have World Read and Execute, e.g. rwX---r-x

Note:
It is a very bad idea to set Write access for anybody other than the owner