COMP519 Practical 1
HTML (1)

Introduction

- This practical is dedicated to creating your first HTML page. While you work through the exercises below compare your results with those of your fellow students and ask for help and comments if required.
- You might proceed more quickly if you cut-and-paste code from this PDF file. Note that a cut-and-paste operation may introduce extra spaces into your code. It is important that those are removed and that your code exactly matches that shown in this worksheet.
- The exercises and instructions in this worksheet assume that you use the Department's Linux systems. This is to ensure that by the time we turn to server-side programming of web applications you are sufficiently familar with the Linux systems that you will need to use for that.

  For HTML documents and client-side scripting, it is possible to create the necessary files on the Windows filestore, but then they must be opened directly instead of via the Departmental student web server as shown in the exercises.
- Use MobaXterm to connect to one of the Linux systems, log in using your MWS username and password. To edit files use the MobaXterm editor.
- If you do not manage to get through all the exercises during this practical session, please complete them in your own time before the next practical takes place.

Exercises

1. Let us start with the introductory example you have seen in the lectures.

   a. Create a directory

   ~/public_html/

   using the command

   ```
   mkdir $HOME/public_html
   ```

   in the MobaXterm terminal. Then create an empty file with the name `page01.html` in that directory with the command

   ```
   touch $HOME/public_html/page01.html
   ```

   b. Open the file `page01.html` in the MobaXterm editor. Enter the following HTML markup in the editor:

   ```
   <html>
   <!-- File: page01.html
       Creation: 2020-10-12
       Description: introductory page
    -->
    <head>
     <title>My first HTML document</title>
    </head>
   ```

```
 <body>
  <p>Hello world!</p>
 </body>
</html>
```

You can also cut and paste the markup from the PDF file, but if you do so, make sure that in this process no extra spaces are introduced into the markup.

c. Save the HTML markup to `page01.html` in your `public_html` directory. A simple 'Save' should suffice for this purpose.

d. Now we have to make sure that the permissions for various directories and files is set correctly, so that our web server can access them but nobody else can. In particular, your home directory and the `public_html` directory must be world-readable and world-executable, while the HTML file must only be readable by yourself. You can set the permissions correctly using the commands

```
chmod a+rx ~/
chmod a+rx ~/public_html/
chmod og-rwx ~/public_html/page01.html
```

e. Open a web browser, i.e. Google Chrome or Mozilla Firefox, and access the URL

`https://student.csc.liv.ac.uk/~<user>/page01.html`

where you need to replace `<user>` with your username. Figure 1 shows you what you should see. If this does not work for technical reasons, then on the Windows side, locate `page01.html` in a file manager and open it with a web browser.

f. The web browser can show you the HTML markup it is rendering. For example, in Google Chrome, right-click somewhere on the web page and select the option 'View page source' in the menu that opens; in Mozilla Firefox, right-click somewhere on the web page and select the option 'View Page Source'. A new tab should open in the web browser showing your HTML markup. For `page01.html` you should see exactly the contents of the file `page01.html`. Later, when we move on to dynamic web pages, you will find that this correspondence no longer holds.

2. While web browsers will render almost anything that looks remotely like HTML markup, there is a notion of well-formed, valid HTML markup versus ill-formed, invalid HTML markup. While you might think that `page01.html` looks perfectly fine, it is not well-formed.

   To see what is wrong, open the URL

   `https://validator.w3.org/`

   in a new browser tab (Figure 2). Enter the URL of your page into the 'Address' textfield and click on the 'Check' button. The result should inform you that there is one error and at least one warning relating to your web page, the error should be the one shown in Figure 3.

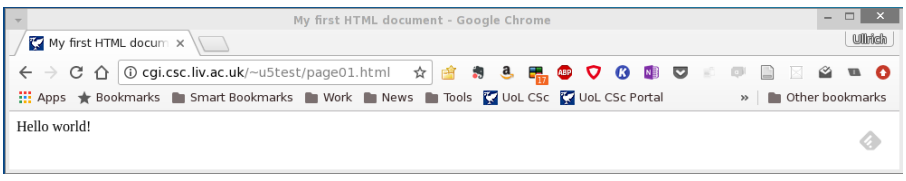

Figure 1: Your first web page in a web browser

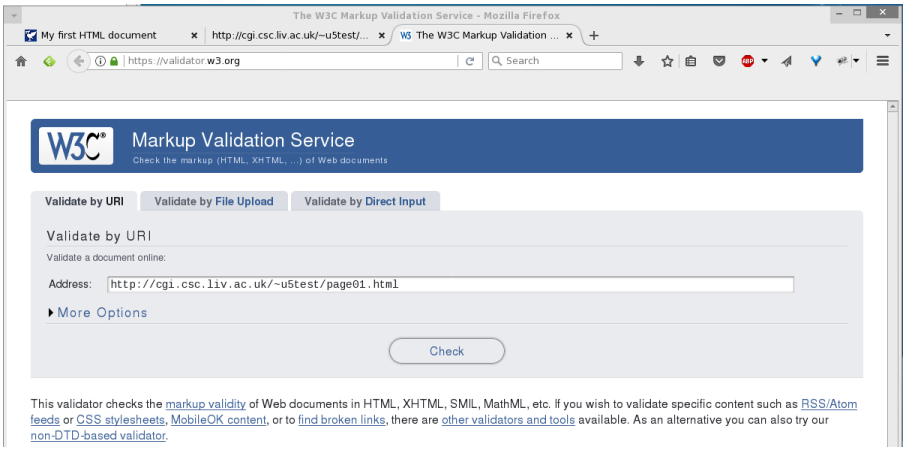

Figure 2: The W3C Markup Validation Service

3. Rectify that error by adding

```
<!DOCTYPE html>
```

   as the first line to the file `page01.html`. Save the file and re-check it.

4. If there are still any warnings, then try to fix those yourself.

5. Our web server might not always immediately pick up changes to your files. It is therefore a good idea to have a version number in the file that you increment every time you save the file after making changes. A version number or an indication of the time of last modification of the file would make it easier to spot whether recent changes to a file are already reflected by the web server.

   Add the following line

```
Last modification <!--#flastmod file="page01.html" -->
```

   just before the closing `</body>` tag in `page01.html` and save the file.

   We are using an Apache Server Side Include here. For more information on server side includes see [1].

6. Now test whether the markup added in the previous step has the desired effect. It does not. You see that the web browser show the additional text ‘Last modification’ but nothing else.

   The reason is that the current version of the Apache web server software that the Department is using only processes server side includes in files with the extension ‘`.shtml`’.

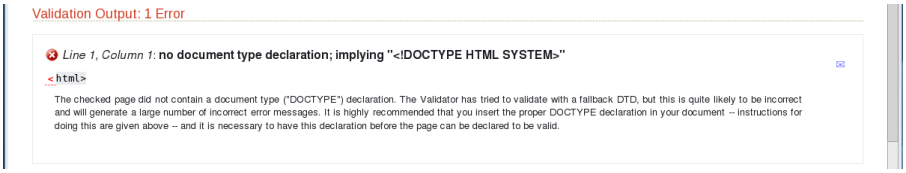

Figure 3: Validation result for your web page

7. We could solve this problem by simply change the name of the file from `page01.html` to `page01.shtml`. But instead we will tell the web server to also process files with the extensions '`.html`' and '`.htm`'. To do so, in the `public_html` directory create a file `.htaccess` with the following content:

   ```
   addOutputFilter INCLUDES .html .htm
   ```

   Make the file `.htaccess` world-readable by executing the command

   ```
   chmod a+r ~/public_html/.htaccess
   ```

8. Now reload `page01.html` in your web browser. It should now correctly display the last modification time of that file.

References

[1] The Apache Software Foundation. *Apache httpd Tutorial: Introduction to Server Side Includes*. 01 January 2023. URL: https://httpd.apache.org/docs/current/howto/ssi.html (accessed 24 January 2023).