

# CSCU9B2 Practical 1: Introduction to HTML 5

---

**Aim: To learn the basics of creating web pages with HTML5.**

Please register your practical attendance: Go to the GROUPS\CSCU9B2 folder in your Computer folder and double-click on the Register icon. Ask a demonstrator if you need help or something goes wrong.

This sheet contains one CHECKPOINT (see the end of the sheet).

## Further Documentation

There is a short summary of basic HTML on the CSC9B2 module **Resources** web page and the lecture slides should also be a handy reference point.

You will find it helpful to consult *Creating a Website: the Missing Manual* (MacDonald 2011, available as an e-book from the library). Chapters 1, 2, 7 and 8 are relevant to this worksheet.

See also the W3Schools tutorials on HTML at <http://www.w3schools.com/html/>.

## Creating a Simple Web Page

*(I realise that quite a few of you will have done much of this in CSCU9B1: but bear with us, as some have not. Even if you have, quickly read through this sheet – there is some new material about IE11, specifically about testing web pages; and you should prepare for the checkpoint.)*

The tools for use in the lab are **TextPad** and **Internet Explorer 11 (IE11)**. TextPad is a fairly intuitive text editor, and provides syntax colouring (that is, different parts of the text are coloured differently so that you can easily differentiate, for example between HTML tags and text).

## Preparatory work

*(If you already have created web pages in your University account, and if these already work, please ignore this section).*

In order for you to be able to make web pages visible, you need to start off by creating a folder called **web**, and fixing its permissions appropriately. The University's systems will do the rest.

1. Create a folder called **web** at the top level on your H: drive (your home directory)
2. Open GROUPS\CSC9B2\ then double click on the **setweb** (maybe setweb.cmd) icon.

These steps will create a web folder and set the permissions on it appropriately.

Now we will create a basic web page. Bring up the TextPad program from the Start menu:

⇒ **All Programs : Applications : TextPad**

then type in the HTML required to generate a simple page, for example:

```
<!DOCTYPE html>
<html>
<head>
  <title>All About Me</title>
</head>
<body>
<p>
  This is <b>my very own web page</b> all about me!
</p>
<p>
  I wonder what I shall write in it...
</p>
</body>
</html>
```

Use **File : Save As** to save this file in your **web** folder with the name **index.html**. Saving it with the **.html** extension ensures that TextPad displays the HTML key words in a colour-coded format.

Display this page in the Internet Explorer (IE) browser by opening IE and doing one of:

- using the **File** menu : **Open** option of IE,
- *or by dragging and dropping* the aboutMe.html file *on to an already open IE window,*
- *or by double-clicking* on the icon for index.html.

The address bar near the top of the browser window will show a "URL" something like:

H:\...\web\index.html. This is **not** a proper network URL, but indicates that the browser has fetched a file directly accessible on the local computer. This is the easiest way to check the appearance of Web pages that you are creating. You will see later how to create a page accessible via a full URL and HTTP request from anywhere in the wired world.

Take a look at the elements that make up your text file index.html. The document follows a structure that is standard for all HTML files:

```
<!DOCTYPE html>
<html>
<head>
  <title>...</title>
</head>
<body>
  ...
</body>
</html>
```

The first line names a *Document Type Definition (DTD)*, saying that this is an HTML document: for HTML5 this is all that is required (earlier versions of HTML had different requirements). The rest of the document consists of an <html> element, which encloses a <head> element followed by a

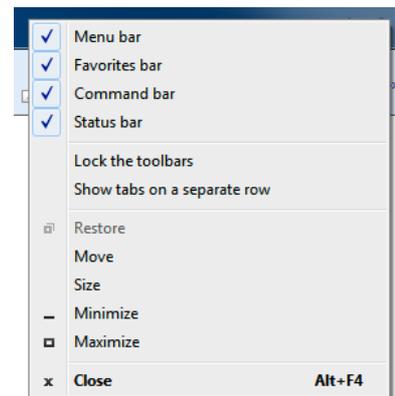
<body> element. The head contains a <title> element. All well-structured HTML documents should contain all these parts.

In this particular example, the body contains two paragraph elements, enclosed within <p> and </p> tags. Paragraph elements act as containers for text. The first paragraph contains some text highlighted in bold. Make sure you can see how each of the elements in the code is rendered by the browser.

## Using the Browser effectively

To use the browsers effectively you need to be able to see the menu line. As initially configured in the lab, IE11 is set up assuming that you are only a consumer of web pages. To improve things:

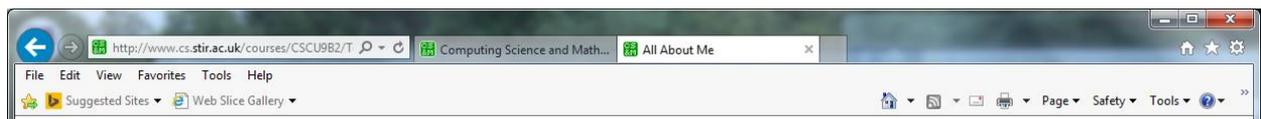
- Right-click in the dark blue area near the top of the IE11 page. This should bring up a menu that looks something like this:



You should ensure that the top four bars are ticked – that way, interacting with the browser for development of web pages is considerably easier.

Also to ensure that the capabilities of HTML5 are enabled, bring up the **Tools:Compatibility View Settings** panel. Ensure that “Display Intranet Sites in Compatibility View” is un-ticked (it probably already is). That way, IE11 should be able to deal with HTML5 web pages reasonably correctly.

The top of your IE11 window should look something like this:



## Creating a Home Page

Open the file **index.html** within TextPad (if it is not already open). You are going to use this to create your own personal web page, using the suggestions given below. There is a short summary of basic HTML on the CSC9B2 module **Resources** web page and the lecture slides should also be a handy reference point. For more detailed information consult the references listed at the start of this sheet.

Note that because index.html is stored in your **web** folder, your web page will be visible throughout the Internet, so take care what you write on it! Do not reveal your contact details or any private information. In this practical we will add content to this page. Later, we will try to make it look good too... (Note that if you save your file anywhere other than in your “web” folder, and then drag and drop it into “web”, it may not work correctly – though this can be fixed (see end of this lab sheet).)

Your web page (index.html), and any other pages you create in your web folder are “browseable” on the Internet. The full URL for the page will be:

`http://www.students.stir.ac.uk/~qjx001/index.html`

where you must replace qjx001 with your own username. The browser/web server will automatically convert the URL to the following form:

`http://www.students.stir.ac.uk/users/x/qjx001/Web/index.html`

which is the more precise location of the file in the Wide file store.

To make the browser load this page using a full HTTP call via the network, type either URL above into the browser's address bar. You should normally use the first form of URL (in case Information Services changes the location it converts to). If you receive an "error 404", see the comments at the end of the sheet.

*Documents in your Web folder can be browsed from throughout the Internet using both these forms of URL – so you could try reading your home page from home (if you have Internet access there), or your friends at other locations can read your home page. Note also that the file name index.html is special: a browser will look for this file if no filename is given in the URL, so you can use the form:*

`http://www.students.stir.ac.uk/~qjx001/`

## Content

What should your home page look like? That's really up to you, but please follow these guidelines. Note that for now we would like you to create the page using simple HTML: no CSS (we'll add that later). To get the **checkpoint** your page must use some **lists** (numbered and/or un-numbered and/or description), as well as a **table**, and a variety of **headings**. Your page must also include some **images** and some links to other **websites**.

- ◆ Make the title reflect what the page is: call it something like:

"Fred Bloggs Home Page"

- ◆ Make the top level heading the same as the title.

What about the actual content? Take some care what you write (see below), remember that what you write will be visible. You could (for example)

- ◆ Say that you are a student
- ◆ Say what modules you are studying (nice place for a table?)
- ◆ Write about your interests (nice place for some lists, perhaps an image as well?)
- ◆ Write about where you come from (perhaps you could find some images, and use them as links?)
- ◆ We used to say: "Put your name and email address at the bottom, so that you can be contacted. (In a later practical you will see how to insert a clickable "mailto" link in there too.)". However, there are now programs that go and "harvest" email addresses for use by email spammers, so that this is no longer a good idea. So now we say:

- *Don't* put your email address, phone number or postal address on the page (or if you do put your email address on it, disguise it, for example writing something like:

**xyz001 (no spam) at (no spam) students dot stir dot ac dot uk.**

This *should* stop it from being harvested by would-be spammers.)

Later on in this module you will look at using CSS to improve the appearance of your page(s).

Whatever you do write about, please **take care what you put in your home pages:**

**Do not use anything which is, or could be taken to be ageist, racist, sexist, inflammatory or indecent. Do not use anything which is, or could be taken to be, libellous or slanderous. Remember that these home pages are visible throughout the Internet – and if inappropriate information is placed on them, the University will be unwilling to continue to provide this facility.**

Ask your neighbour what their username is, and try to view *their* home page via the browser – you will again need to customise one or other of the two URL forms given above. With their home page on view, select **Page: View Source** – you can then see *your* browser's copy of *their* HTML: think about whether it is laid out in a readable fashion (but it may be better to keep any thoughts you have to yourself!).

## Validating HTML

The homepage that you have just constructed probably looks fine on your browser. However, perhaps it contained some hidden problems that your browser was tolerant of, but that other browsers would not be tolerant of and so would not display the document as you intended. The World Wide Web Consortium (W3C) has an HTML validation service that you can use. Below we describe how to use it. Once you are used to it, it is a good idea to use it *before* you demonstrate your checkpoints in all these practicals.

Type the URL <http://validator.w3.org/> into the address bar of your browser window. There are various information resources and services available from this page. You will find a form for requesting validation checks. Before you try this on your own page, try it on one we created earlier: type the following address into the Address box and click Check:

<http://www.cs.stir.ac.uk/courses/CSCU9B2/testcode/cake.html>

This should validate with a couple of errors. You can look at the source of this page if you like: display the page in the browser, then use **Page: View Source** to display the source HTML. Later in this practical we will try debugging this page ie find and fix the errors.

Now type the URL of your home page into the top text box (URI) in the form and click on the button "Validate this page". After a pause (during which the W3C server fetches your home page and analyses it) you will get a report on the screen – you will probably need to scroll down to see the whole report. You will probably see *at least* one complaint reported:

- No "character encoding" was found: you can fix the problem by adding the following line of HTML into the <head> part of your home page:

```
<meta charset="UTF-8">
```

There may be other problems or weaknesses reported too – try looking at them and fixing them (by editing the HTML of your home page using TextPad). After you think that you have fixed the problems, click the “Check” button on the report page and see if all is then OK – if not, then try again...!

More validation things to try:

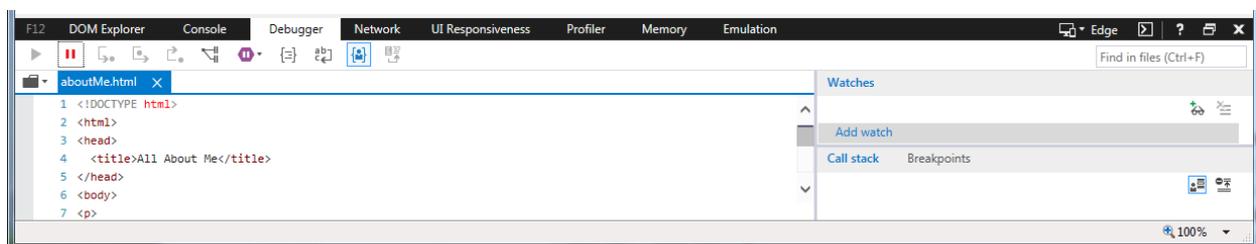
- You could try introducing HTML errors (such as omitting `< ... >` tags) and then revalidating to see what complaint is produced in the report.
- You could try validating other pages to check the “quality” of their HTML content (for example, your neighbour’s home page, Stirling University home page (<http://www.stir.ac.uk/>), the Computing Science and Maths Department home page (<http://www.cs.stir.ac.uk/>), the CSCU9B2 home page... You will probably see lots of errors on some of these as they have not been fully converted to HTML5 from older versions of HTML).

**CAUTION:** If you ask the validator to check a page that needs a username and password to authorise the access (e.g. the CSC9B2 lecture notes), then it will ask *you* to tell *it* your username and password so that it can pass them on to the server actually needing the username and password – this in general is **not a good idea**, so, if it happens, click on cancel (and read the explanation that you will see on the screen – it’s quite well written).

## Testing your HTML

Browsers can do more than simply display the web pages you are writing, it may also help you fix problems with your HTML (or CSS and JavaScript), such as those revealed by validation.

In IE11, as you did above, press the F12 key (or, alternatively, navigate to **Tools: F12 Developer Tools**). Now you should see another panel, at the bottom of the browser window, in addition to the one that you are using to display the web page itself. This window panel has a great deal of functionality, and we shall use it in a moment. It should look something like:



Look at what you can see in this window: it is primarily intended for detecting errors in JavaScript scripts, of which more later. It can also detect some HTML and CSS errors as well.

To try some debugging of HTML for yourself, load the previous example file again (if necessary):

<http://www.cs.stir.ac.uk/courses/CSCU9B2/testcode/cake.html>

Press F12 to bring up the Developer Tools window for this web page. Select the **DOM Explorer** tab and you will see the structure of the HTML. You can click on any of the arrow heads to expand a section and see its content: try this now e.g. the open arrow head to the left of <h1>.

Now click the **Console** tab. This window reports any messages from the browser and in this case it indicates that there is a problem with the HTML (much like the Validator, but the Validator is usually more comprehensive in its error reporting). Read the warning message: in this case this message is due to a mistake in the HTML.

Go back to the DOM Explorer tab and see if you can find what might be wrong. If you spot something that you would like to fix, right-click on the HTML element to be changed and select “Edit as HTML”: you can now make any changes that you wish and if you then click somewhere outside the editing box, the changes will be applied and the web page display will be updated. (NOTE: these changes are NOT applied to the source code, but only internally in the browser, so if you refresh the page the error will return).

What mistake did you find and fix for this web page? Ask a demonstrator if you want confirmation.

### CHECKPOINT [HTML]

Please demonstrate your validated homepage to laboratory staff. You should have correctly added a title, some headers, lists and tables, an image, some links, as well as some paragraphs in the body about yourself.

### HINT: cannot access your web page?

You may occasionally find that there are files that you can clearly see in your Web folder, but that the Web server refuses to, or cannot, fetch when requested - you type their URL into a browser but you get an error page sent back, usually mentioning “HTTP error 404 - File not found” or “HTTP error 403 – Forbidden” (for example, try <http://www.stir.ac.uk/x>). Here is how it might have happened, and how to fix it:

The HTTP (Web) server program has few or no special privileges, and requires all the files in your Web folder to be accessible to “Everyone” in order to fetch them to respond to a request. On the other hand, everything else in your home folder is automatically protected from access by anyone else - in particular if you initially create them **outside** the Web folder and then move (drag) them to inside it. In this case the server will report back to the browser that the request cannot be satisfied, and the browser will report failure or perhaps pop up an authorisation dialogue. You can usually fix the problem as follows: Open GROUPS\CSC9B2\ then double click on the setweb (maybe setweb.cmd) icon. After a little dazzling magic, the permissions in your Web folder will have been fixed to be “Readable to everyone”. The SetWeb action is also available via the Windows Start menu: **Start / Programs / Internet / SetWeb**.