



Edinburgh Napier University

**SETo8101/SETo8401 Web Tech**

---

## **Lab 3 - Semantic HTML**

---

**Dr Simon Wells**

# 1 Aims

By the end of the practical portion of this topic we will have:

- Created Semantic HTML documents through appropriate use of HTML5 elements
- Practised our use of standard HTML
- Investigated and considered how existing sites use semantic HTML (or don't)
- Elaborated on some of the sites you started developing in previous weeks, helping partly to kick-start our development of a personal portfolio, and partly to stimulate our interest in the variety of types of sites we can implement for the web.

**NOTICE: You should be putting your lab work into Git (Add, Commit, Push) as you develop it. This is active & deliberate practise and means that you are constantly building skills so that you no longer have to waste cognitive energy on them; you just use them. If you are looking for a good Git tool then Git-Bash is probably the best and available from here: <https://git-scm.com/download/win>. Rather than the default that tries to auto-download, I'd choose the portable "thumbdrive" edition. You can then follow this tutorial to use git-bash with GitHub: <http://vastinfos.com/2016/09/quick-github-and-gitbash-basics-for-beginners-tutorial/>**

## 2 Activities

You should, unless otherwise indicated, try out all of the code examples shown on lab sheets for yourself. Whilst this lab sheet can describe an outcome, this is no substitute for experiencing that outcome for yourself and allowing it to spur some experimental exploration.

### 2.1 HTML Documents

We've previously seen simple HTML documents that perhaps looked like this:

```
1 <!DOCTYPE html >
2 <html >
3   <head >
4     <title >SET08101 - Web Tech</title >
5   </head >
6   <body >
7     <h1>Hello Web Tech</h1>
8     <p>Welcome to the class</p>
9   </body >
10 </html >
```

Now let's consider a simple page for a cake baking site that uses semantic HTML tags to organise the content:

```
1 <!DOCTYPE html >
2 <html >
3 <head >
4 <meta charset="UTF-8" />
5 <title>Untitled Document</title >
6 </head >
7 <body >
8
9
10 <header >
11 Baking Cakes
12 <hr />
13 </header >
14 <nav >
15 <a href="/Baking/" target="_blank">Baking</a> |
16 <a href="/ingredients/" target="_blank">Ingredients</a> |
17 <a href="/mixing/" target="_blank">Mixing</a> |
18 <a href="/toppings/" target="_blank">Toppings</a>
```

```

19 </nav>
20 <section>
21 <h1>Cake Tips and Techniques</h1>
22 <h2>Professional Tips</h2>
23 <p>
24 When baking a cheesecake, it is important not to over bake it. You only want to
    bake it until the middle has a slight wiggle, not until it is rock solid.
25 </p>
26 <p>It is important that you use a water bath, discussed at the right, to ensure
    even baking of your cheesecake.</p>
27 <aside>
28 To create a water bath, use a pan that will allow you to fill it with boiling water
    that
29 goes halfway up the spring form pan in which the cake is placed.
30 </aside>
31 </section>
32
33 <footer> &copy; 2020 - Better Made Cakes - All rights reserved </footer>
34
35 </body>
36 </html>

```

Notice that we have a mix of semantic and regular HTML, to both organise the page into thematic groups of tags, but also to mark-up the content of the page. For example, each section, in a larger page, might have a set of heading, so regular HTML is still needed, even though we are now adding in these organisational, *semantic*, elements. Try to pick out the various sections. Use this as the basis for building a few pages for a cookery site (changing as much as necessary). Make sure to include a few recipes so that you can consider how you might collate and organise the ingredients list, the procedure, and the description of the resulting dish.

In the following, don't worry too much about making your pages look nice. We will address that in subsequent labs, but for now, think about structure and content, presenting your ideas in a clear way using HTML5 and focussing on the application of semantic tags to group and organise your data. We will return to these challenges over subsequent weeks to elaborate on them with design and interactive elements as appropriate so don't ignore them. By the end of this module you will have developed your own set of websites of various types and purposes. Think of these not just as a framework for meeting the various learning objectives of the module, but also as a way to starting building your own portfolio of work. There is nothing to stop you elaborating on any of these exercises and making your own, unique demonstrators of your skills as we progress. If you are happy with what you've built then you can consider deploying some, or all<sup>1</sup> of your sites<sup>2</sup>.

1. Revisit the web pages you built to describe yourself and your career at Napier. Create a new version that applies the HTML5 semantic tags in order to organise the content of your page so that your page's content is sensibly grouped and organised. There is no *absolutely correct* way to do this, and the on-screen effect, at this point, won't be hugely different from the earlier pages. However, you should notice that it is immediately easier to find your way around your pages HTML code. This is compounded when we start to interact with our sites from CSS or JavaScript.
2. Build a set of pages, using semantic HTML to describe a hobby or some other topics that you are interested in. You should be carefully considering how to break things down into separate pages and how each page can be organised in terms of semantic notions like header, footer, section, article, aside, and nav.
3. Return to the CIA world fact book website. As we saw before, the factbook contains information about various countries around the world. Have another look at the kind of information available and this time design a simple set of pages that reproduces some of the factbooks content but ensuring that each page is structured semantically.

Don't worry if you're not sure how to do all of what you want to do yet. This is why these exercises are challenges. It is also fine to "build one to throw away". Try out a test spike to get something working on screen, then, early in the process, before you've put in too much effort, decide whether things are developing the way you want and whether to continue or start again.

<sup>1</sup>or even none :D

<sup>2</sup>We'll consider deployment of websites later in the module

Often in trying a test spike we learn enough about the problem and potential solutions that we are better equipped to make a second, improved, attempt.

Speak to your classmates about how they are approaching things and see if you can come up with good ideas. You can also research how others have built similar sites online to see if you can gain inspiration. Remember that when you look at websites online, you can also inspect the code that drives the website, the HTML sourcecode, and in many cases, although developers will sometimes try to hinder you, the CSS, JavaScript and other resources, are also available to inspect, so you can see how others have achieved things. If you find something that you like then you should consider how to incorporate those aspects into your own sites where it is appropriate to do so<sup>3</sup>.

## 2.2 Challenges

1. Visit a web-site that you are familiar with and use reasonably frequently. Without looking at the underlying HTML just yet, build your own version of the site, but applying semantic HTML tags to organise the content. You should be carefully considering how the different “types” of information that make up the page(s) work together. Obviously your site won’t look anything like the original, because we won’t be styling it, but the idea is to gain extra practise considering how sites of different types can organise their content.
2. Research a variety of web sites, investigating their source-code, using right-click “View Page Source”, to gain an understanding of the range of ways that different web designers will organise and present their work using HTML. Do you notice anything? How many sites seem to have clean and easily understandable structures for their content? How many are using semantic HTML consistently? How many are using div tags to differentiate between logical sections? Of these, how often might those same sections be represented using standard semantic HTML tags rather than custom tags. You should notice that many sites organisation are a result of being ‘slaves to their design’ and ultimately their CSS. Even though we haven’t looked at CSS yet you should notice that many sites are much more complicated than perhaps necessary.
3. Create individual Git repositories for each of your sites and push them to your GitHub account. Use “GitHub pages” as your web server and ensure that each “site” is available from its own github based URL. How you organise your Git repositories is up to you, but you should be considering how your pages are organised from a source code perspective. For example, rather than having everything together in a single folder you might want to have various file-types organised into different sub-folders. For example, placing images together in a sub-folder called images. Later, as we accumulate files of different types, like CSS and JS, we will see how organisation of our resources can make life much easier as a developer<sup>4</sup>.

## 2.3 Finally

If you want some additional practise then the W3Schools HTML exercises are a great place to start:

- HTML Exercises: <https://www.w3schools.com/html/exercise.asp>

---

<sup>3</sup>Note that this doesn’t mean to copy and paste another’s code, but to take inspiration for how to implement your own solution to a given problem.

<sup>4</sup>and if it’s easier for us as the creators then it must be much easier for anyone who might inherit our work in the future (or if we return to our work in the future)