Practical 2: Introduction to HTML and CSS

Aims

- Familiarising yourself with web frameworks and code editors
- Familiarising yourself with languages for structuring and styling web pages: HTML and CSS

Task 1.1

Search for and install on your machines the **web framework** and **code editor** we agreed on during the Friday lecture.
Explore the tools' features – you will use them in the next task.

Task 1.2

HTML Resources

You can read more about HTML and find tutorials for it on:

- MDN: https://developer.mozilla.org/en-US/docs/Web/HTML
- The Odin Project: https://www.theodinproject.com/courses/web-development-101/lessons/html-and-css-basics
- W3Schools: https://www.w3schools.com/html/default.asp
- CodePen: https://codepen.io/

CSS Resources

You can read more about CSS and find tutorials for it on:

- MDN: https://developer.mozilla.org/en-US/docs/Web/CSS
- The Odin Project: https://www.theodinproject.com/courses/web-development-101/lessons/html-and-css-basics
- W3Schools: https://www.w3schools.com/css/default.asp
- CodePen: https://codepen.io/

To do:

1. Write an HTML document

Write a one-page HTML document. The page should present a CS/IT application that has made a significant contribution to human society or had a profound impact on the global community. You are free to choose what that can be – an application that inspires you.

The web page should include a:

- *Heading* with text on a coloured background
- *List*
- *Hyperlinks*
- Consistent, but non-default *font*

2. CSS file

All the style information should be included in a separate CSS file.

- Style your page according to your liking.
- You are to reference/link the CSS file in the HTML document.

3. Use a code editor

To write your HTML and CSS code, use the agreed on code editor.

You can view, test and develop your page by accessing the page from within your file browser. It will launch a web browser and access it via the `file:` scheme.

Note: For web pages hosted on the Internet, `http:` scheme is used – you can learn more on: List of URI schemes

4. Use a framework

Make a copy of your HTML page and change it so that it uses the agreed on web framework.

- You should have to change very little of the HTML content, only the CSS classes referred to.
- Use templates that are provided in the documentation where relevant.

5. Make your HTML page mobile-friendly

Make your page responsive for mobile devices, by including:

- a navbar (collapsible)
- a grid layout for content

Look at the HTML and CSS resources for instructions on how to do this.

6. Publishing your page (next practical)

Next time you will learn about git, GitHub, and how to publish your web page.

You can publish your web page by using gitpages. Instructions for this are on: work with GitHub pages.