Matt Tennison
(140197385)

CSC1026: Assessment 2 Report

My final website has a number of key features that differentiate it from other tutorial websites:

- Responsive design: I've made two separate stylesheets for small browsers (phones, vertical tablets etc) and large browsers (desktops, laptops and horizontal tablets). This allows users to access it from their mobile devices - a useful feature in a world where mobile web browsing continues to increase at a rapid rate.
- Syntax highlighter: Admittedly this isn't self-coded, I've used a Javascript library (prism.js). Never the less it's a very useful feature for users, and allows them to quickly read and understand code examples. Without the syntax highlighting, users might struggle to differentiate between CSS properties and the values (property: value;)
- jQuery animations and toggles: I've built a number of buttons into my website that allow users to toggle between two different styles on an element (for an example see: learn/CSSBoxModel.html). This allows users to easily tell the difference between two sets of properties. For example, in the box model page this makes it clear that whilst padding includes the background, margins don't.

My key features don't differ from the pitch's "3 technical features you intend to include…". In this regard I believe I've satisfied my pitch.

I've applied a lot of knowledge from CSC1026 lectures to my tutorial website. Through the lectures I've built on my understanding of Javascript, allowing me to implement more ambitious features. I've also learnt to modularise my work so there is a clear distinction between HTML, CSS and Javascript. For example, I've tried to avoid using the style attribute in HTML, instead using IDs and classes in CSS. Thanks to the lectures I finally clearly understood the distinction between the display types (block, inline and inline-block). This proved very useful since I based an entire portion of my tutorial website on it! Likewise my understanding of the box model improved thanks to the lectures, which was very useful.

I also learnt about using Google Fonts offline, since one of the requirements for the project was that the site had to run offline. In my personal usage of Google Fonts I've always used the online imports.

I've used the following external libraries and frameworks:

- jQuery. This Javascript framework makes it easier to manipulate the DOM. For example, instead of writing document.getElementByID('id') you can write $('#id'). I used jQuery extensively, since I prefer it to writing vanilla Javascript.
  - I downloaded it on the 2nd of May.
  - The version is: 2.13
  - It uses the MIT License, allowing for use in any project, personal or commercial.
  - The source is: https://jquery.com/download/

- Google Fonts, "Open Sans". This allowed me to use a suitable font, "Open Sans" in my project. Whilst not essential, I believe it added to the visual appearance of the website.
  - I downloaded it on the 11th of May (previously used it through online @imports, but this wouldn't work with the pitch).
  - It uses the Apache 2.0 License.
  - The source is: https://www.google.com/fonts/specimen/Open+Sans#charset

- Font Awesome. This is a 'font' that consists of hundreds of vector icons. Since it's free to use (MIT License) and all the icons are of a consistent nature, I felt it was ideal for this project where visuals are important.
  - I downloaded it on the 23rd of April
  - The version is: 4.3.0
  - It uses the following licenses: SIL OFL 1.1 (font), MIT License (CSS, Less, Sass)

  - The source is: http://fortawesome.github.io/Font-Awesome/

- Animate.css. This is a stylesheet that makes it easy to add animations to elements. I used this class to avoid the hassle that is CSS3 animations myself.
  - I downloaded it on the 23rd of April
  - The version is: 3.2.5
  - It uses the MIT License
  - The source is: http://daneden.github.io/animate.css/

- Prism.js. This is a syntax highlighter with support for many languages. It makes it trivial to add syntax highlighting to any code block.
  - I downloaded it on the 1st of May
  - It uses the MIT License.
  - The source is: http://prismjs.com

- Normalize.css. This is a stylesheet that ensures your website looks similar in different browsers, by resetting all browsers to a standard stylesheet.
  - I downloaded it on the 23rd of April
  - It uses the MIT License.
  - The source is: https://github.com/necolas/normalize.css

- The Fisher-Yates (aka Knuth) shuffle. This is a simple JS function that allows you to shuffle a array. I've used this in my questions list, to ensure that all the answers switch around each time.
  - I downloaded it on the 10th of May.
  - It uses the Apache License.
  - The source is: https://github.com/coolaj86/knuth-shuffle

I believe my tutorial website is of high quality and easy to navigate. Upon first load, I believe most users would be able to navigate and access all the content. I've designed my website to be accessible on a range of browser sizes, including as far down as mobile devices. I'm especially proud of the quiz, which features animations (courtesy of animations.css and it's excellent documentation) and is easy to use. At the end the user gets a clear score, and whilst taking the quiz they can follow their score thanks to the overlapping scoreButton div.

One problem I encountered was my topic areas were too specific; it limited the amount of detail I could go into drastically. There's only so much content you can cover on the box model for example, and whilst I believe visitors will be able to follow and learn from my content, the scope could be improved. If I had to repeat this task in the future, I would likely prepare the content (or at least a draft) before finalising the pitch. This way I would have been able to notice how limited my content areas were and either change them or expand the scope to include more CSS topics.

Another area I feel my website is lacking is in diagrams - in certain topics (box model especially), diagrams would help cover the topic in more detail. Whilst I believe my text goes into enough detail, it might be more accessible to include diagrams. This would also allow users to quickly check my website for reference when they forget (almost like a cheatsheet).

In conclusion, I believe my website satisfies the specification and I have met my aims laid out in the pitch. I have implemented all the technical features, and covered all of the topics mentioned. I have also used a mixture of HTML, CSS and Javascript to deliver a interactive experience that teaches users the content they're after in a clear and concise manner.