Core Computing Concepts (Item 2 - Web)
Discovery Log

Thomas Boxall

November 2022 - December 2022

Contents

Section 1

Week 01

1.1 Setting up Domain

Rather than using the university provided VM to host the content created during this part module, I will setup a subdomain on my own website. This is to prevent the requirement of either Eduroam, a University computer or the University VPN to access the content I've produced. The content I've produced will also always be accessible as well.
To host To do this, I will go to the c-panel of Hostinger, the service I pay to host my website through, and create a new subdomain. I have called the subdomain `ccc`. The full address of the subdomain is `ccc.thomasboxall.net`.
After configuring the subdomain, and ensuring it worked. I created a page which would act as a menu. I called this page `index.html` and placed it at the root of the subdomain folders so that the page would open whenever the subdomain was accessed.

1.2 Lecture

1.2.1 HTML5

When starting a new HTML document, which you want to use HTML5 within. The first line of the document must be `<!doctype html>`. This tells the browser that the document is a HTML5 document and that it should be interpreted as such.

1.2.2 Meta

Meta tags can be used to provide additional information about the document. This information doesn't get rendered on the webpage, however it aids the search algorithm with working out what is on the page.

1.2.3 Document Structure

New to HTML5, you don't have to specify the `<html>`, `<head>` and `<body>` tags. The browser can interpret this automatically.
In the past, documents would have looked like:

LANGUAGE: HTML

```html
<!doctype html>
<html>
  <head>
    <title>HTML Example</title>
    <meta name="author" content="Matt Dennis">
    <meta name="date"   content="March 2020">
  </head>

  <body>
    <header>
      <h1>HTML Example</h1>
      <p>Matt Dennis</p>
      <p>March 2020</p>
    </header>

    <section>
      <h1>Introduction</h1>
      <p>Paragraph Text.</p>
      <p>Isnt it …pretty? <em>Less so.</em></p>
    </section>
  </body>
</html>
```

Whereas, with HTML5, HTML documents can be much more skeletal,

LANGUAGE: HTML

```html
<!doctype html>
<title>HTML Example</title>
<meta name="author" content="Matt Dennis">
<meta name="date"   content="March 2020">

<header>
  <h1>HTML Example</h1>
  <p>Matt Dennis
  <p>March 2020
</header>

<section>
  <h1>Introduction</h1>
  <p>Paragraph Text.
  <p>Isnt it …pretty?
    <em>Less so.</em>
</section>
```

Examples adapted from Lecture01 slides.

1.3 Worksheet Task 2 & 3

Using the template provided, I created the following page.

LANGUAGE: HTML

```html
<!doctype html>
<title>Thomas Boxall</title>
<h1>Thomas Boxall</h1>
<p>contain whatever you would like</p>
```

I then open the page in a browser to ensure it worked.

1.4 Worksheet Task 4

I created a new HTML document and wrote the following code to produce a page giving some basic information about each of the modules I'm currently studying.

LANGUAGE: HTML

```html
<!DOCTYPE html>
<title>Modules</title>
<marquee><h1>Modules</h1></marquee>
<h2>M22731 - Comp Tutorial Level 4</h2>
```

```html
<p>A philosophical approach to learning and Computer Science</p>

<h2>M30220 - Core Computing Concepts</h2>
<p>A 4-part module introducing the history of computing, the web, security and user experience.</p>

<h2>M30231 - Networks</h2>
<p>An introduction to networking.</p>

<h2>M30232 - Database Systems Development</h2>
<p>An introduction to Databases using PostgreSQL.</p>

<h2>M30299 - Programming</h2>
<p>The basic concepts of programming featuring Zelles Graphics package for Python.</p>

<h2>M30943 - Architecture and Operating Systems</h2>
<p>A two part special, featuring the fundamentals of how computers work and maths.</p>
```

1.5 FreeCodeCamp tutorial

1.5.1 Learn HTML by Building a Cat Photo App

LANGUAGE: HTML

```html
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>CatPhotoApp</title>
  </head>
  <body>
    <main>
      <h1>CatPhotoApp</h1>
      <section>
        <h2>Cat Photos</h2>
        <!-- TODO: Add link to cat photos -->
        <p>Click here to view more <a target="_blank" href="https://freecatphotoapp.com">cat photos</a>.</p>
        <a href="https://freecatphotoapp.com"><img src="https://cdn.freecodecamp.org/curriculum/cat-photo-app/relaxing-cat.jpg" alt="A cute orange cat lying on its back."></a>
      </section>
      <section>
        <h2>Cat Lists</h2>
        <h3>Things cats love:</h3>
        <ul>
          <li>cat nip</li>
          <li>laser pointers</li>
          <li>lasagna</li>
        </ul>
        <figure>
          <img src="https://cdn.freecodecamp.org/curriculum/cat-photo-app/lasagna.jpg" alt="A slice of lasagna on a plate.">
          <figcaption>Cats <em>love</em> lasagna.</figcaption>
        </figure>
        <h3>Top 3 things cats hate:</h3>
        <ol>
          <li>flea treatment</li>
          <li>thunder</li>
          <li>other cats</li>
        </ol>
        <figure>
          <img src="https://cdn.freecodecamp.org/curriculum/cat-photo-app/cats.jpg" alt="Five cats looking around a field.">
          <figcaption>Cats <strong>hate</strong> other cats.</figcaption>
        </figure>
      </section>
      <section>
        <h2>Cat Form</h2>
        <form action="https://freecatphotoapp.com/submit-cat-photo">
          <fieldset>
            <legend>Is your cat an indoor or outdoor cat?</legend>
            <label><input id="indoor" type="radio" name="indoor-outdoor" value="indoor" checked> Indoor</label>
            <label><input id="outdoor" type="radio" name="indoor-outdoor" value="outdoor"> Outdoor</label>
          </fieldset>
          <fieldset>
            <legend>What's your cat's personality?</legend>
            <input id="loving" type="checkbox" name="personality" value="loving" checked> <label for="loving">Loving</label>
            <input id="lazy" type="checkbox" name="personality" value="lazy"> <label for="lazy">Lazy</label>
            <input id="energetic" type="checkbox" name="personality" value="energetic"> <label for="energetic">Energetic</label>
          </fieldset>
          <input type="text" name="catphotourl" placeholder="cat photo URL" required>
          <button type="submit">Submit</button>
        </form>
      </section>
    </main>
    <footer>
      <p>
        No Copyright - <a href="https://www.freecodecamp.org">freeCodeCamp.org</a>
      </p>
    </footer>
  </body>
</html>
```

Through this course, I was introduced to the `<section>`, `<figure>` and `<footer>` elements. I was also introduced to basically all of the form elements.
I was surprised that this course enforced the writing style of writing every tag out in full while that is not required as part of HTML5.

1.5.2 Cafe Menu

HTML:

LANGUAGE: HTML

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Cafe Menu</title>
    <link href="styles.css" rel="stylesheet"/>
  </head>
  <body>
    <div class="menu">
      <main>
        <h1>CAMPER CAFE</h1>
        <p class="established">Est. 2020</p>
        <hr>
        <section>
          <h2>Coffee</h2>
          <img src="https://cdn.freecodecamp.org/curriculum/css-cafe/coffee.jpg" alt="coffee icon"/>
          <article class="item">
            <p class="flavor">French Vanilla</p><p class="price">3.00</p>
          </article>
          <article class="item">
            <p class="flavor">Caramel Macchiato</p><p class="price">3.75</p>
          </article>
          <article class="item">
            <p class="flavor">Pumpkin Spice</p><p class="price">3.50</p>
          </article>
          <article class="item">
            <p class="flavor">Hazelnut</p><p class="price">4.00</p>
          </article>
          <article class="item">
            <p class="flavor">Mocha</p><p class="price">4.50</p>
          </article>
        </section>
        <section>
          <h2>Desserts</h2>
          <img src="https://cdn.freecodecamp.org/curriculum/css-cafe/pie.jpg" alt="pie icon"/>
          <article class="item">
            <p class="dessert">Donut</p><p class="price">1.50</p>
          </article>
          <article class="item">
            <p class="dessert">Cherry Pie</p><p class="price">2.75</p>
          </article>
          <article class="item">
            <p class="dessert">Cheesecake</p><p class="price">3.00</p>
          </article>
          <article class="item">
            <p class="dessert">Cinnamon Roll</p><p class="price">2.50</p>
          </article>
        </section>
      </main>
      <hr class="bottom-line">
      <footer>
        <p>
          <a href="https://www.freecodecamp.org" target="_blank">Visit our website</a>
        </p>
        <p class="address">123 Free Code Camp Drive</p>
      </footer>
    </div>
  </body>
</html>
```

CSS:

LANGUAGE: CSS

```css
body {
  background-image: url(https://cdn.freecodecamp.org/curriculum/css-cafe/beans.jpg);
  font-family: sans-serif;
  padding: 20px;
}
h1 {
  font-size: 40px;
  margin-top: 0;
  margin-bottom: 15px;
}
h2 {
  font-size: 30px;
}
.established {
  font-style: italic;
}
h1, h2, p {
  text-align: center;
}
.menu {
  width: 80%;
  background-color: burlywood;
  margin-left: auto;
  margin-right: auto;
  padding: 20px;
  max-width: 500px;
}
img {
  display: block;
  margin-left: auto;
  margin-right: auto;
  margin-top: -25px;
}
hr {
  height: 2px;
  background-color: brown;
  border-color: brown;
}
.bottom-line {
  margin-top: 25px;
}
```

```
h1, h2 {
  font-family: Impact, serif;
}
.item p {
  display: inline-block;
  margin-top: 5px;
  margin-bottom: 5px;
  font-size: 18px;
}
.flavor, .dessert {
  text-align: left;
  width: 75%;
}
.price {
  text-align: right;
  width: 25%;
}
/* FOOTER */
footer {
  font-size: 14px;
}
.address {
  margin-bottom: 5px;
}
a {
  color: black;
}
a:visited {
  color: black;
}
a:hover {
  color: brown;
}
a:active {
  color: brown;
}
```

This project further expanded some of the knowledge I had from tinkering with web development in my own time.
Using the `<article>` tag to encase elements which work together as well as the setting the item's width and `display` using CSS gives an extremely elegant solution which I will take forward to use in my own personal web projects.

1.5.3 CSS Markers

CSS:

LANGUAGE: CSS

```
h1 {
  text-align: center;
}
.container {
  background-color: rgb(255, 255, 255);
  padding: 10px 0;
}
.marker {
  width: 200px;
  height: 25px;
  margin: 10px auto;
}
.cap {
  width: 60px;
  height: 25px;
}
.sleeve {
  width: 110px;
  height: 25px;
  background-color: rgba(255, 255, 255, 0.5);
  border-left: 10px double rgba(0, 0, 0, 0.75);
```

```css
}
.cap, .sleeve {
  display: inline-block;
}
.red {
  background: linear-gradient(rgb(122, 74, 14), rgb(245, 62, 113), rgb(162, 27, 27));
  box-shadow: 0 0 20px 0 rgba(83, 14, 14, 0.8);
}
.green {
  background: linear-gradient(#55680D, #71F53E, #116C31);
  box-shadow: 0 0 20px 0 #3B7E20CC;
}
.blue {
  background: linear-gradient(hsl(186, 76%, 16%), hsl(223, 90%, 60%), hsl(240, 56%, 42%));
  box-shadow: 0 0 20px 0 hsla(223, 59%, 31%, 0.8);
}
```

HTML:

LANGUAGE: HTML

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Colored Markers</title>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>CSS Color Markers</h1>
    <div class="container">
      <div class="marker red">
        <div class="cap"></div>
        <div class="sleeve"></div>
      </div>
      <div class="marker green">
        <div class="cap"></div>
        <div class="sleeve"></div>
      </div>
      <div class="marker blue">
        <div class="cap"></div>
        <div class="sleeve"></div>
      </div>
    </div>
  </body>
</html>
```

This taught me quite a lot about colours and the way in which they interact with one another.

Before this tutorial, I had an understanding of how RBG colours work, however now I know that they can be used in CSS with the syntax `RGB(x,y,z)`, where `x`, `y` and `z` represent red, green and blue values respectively. They values range from 0 to 255.

I have also used HEX colours before. The syntax for this is `#XXYYZZ` where where `XX`, `YY` and `ZZ` represent hexadecimal values from 00 to FF for red, green and blue respectively.

A new colour system to me is HSL. Hue, Saturation, Lightness (HSL) is another colour model which can be used to represent colours. It contains three values. Hue accepts a number from 0 to 360 representing a colour where 0 is red, 120 is green and 240 is blue. Saturation is the intensity of a colour, as a percentage. This ranges from 0 (gray) to 100 (pure colour). Lightness is the brightness of the colour, as a percentage. This ranges from 0 (black) to 100 (complete white), passing through 50 (neutral). The syntax for this is `hsl(aaa, bbb%, cc%)`, where `aaa` represent H, `bbb` represent S and `cc` represent L. The percentage signs are essential.

To create a coloured background of a HTML element. Set the `background` property to the value `linear-gradient(direction, col1, col2...)` where `direction` is the direction in which the gradient goes, `col1` is the first colour, `col2` is the second. There can be as many

colours as you wish. Colours can be specified
Colour stops can be specified by including a percentage value after the colour. For example `linear-gradient(90deg, rgb(255, 0, 0) 75%, rgb(0, 255, 0), rgb(0, 0, 255))` will produce a gradient from red, to green to blue where 75% of the space will be filled by red.

Section 2

Week 02

2.1 FreeCodeCamp

2.1.1 Registration Form

To remove the scrollbar present in some browsers, add the line `margin:0;` to the `body` CSS styling.
Sometimes you will need to specify the last of a type, to do this use `p:last-of-type` . This would select the last `<p>` tag and allow that to have different styling, for example a different amount of padding or a different bottom border.
The `unset` property applied to width will reset the value so that its inherited from its parent.