

# Core Computing Concepts (Item 2 - Web) Discovery Log

Thomas Boxall

November 2022 - December 2022

# Contents

1	Week 01	2
2	Week 02	10

# 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
1 <!doctype html>
2 <html>
3   <head>
4     <title>HTML Example</title>
5     <meta name="author" content="Matt Dennis">
6     <meta name="date" content="March 2020">
```

```
7 </head>
8
9 <body>
10   <header>
11     <h1>HTML Example</h1>
12     <p>Matt Dennis</p>
13     <p>March 2020</p>
14   </header>
15
16   <section>
17     <h1>Introduction</h1>
18     <p>Paragraph Text.</p>
19     <p>Isnt it ..pretty? <em>Less so.</em></p>
20   </section>
21 </body>
22 </html>
```

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

```
LANGUAGE: HTML
1 <!doctype html>
2 <title>HTML Example</title>
3 <meta name="author" content="Matt Dennis">
4 <meta name="date" content="March 2020">
5
6 <header>
7   <h1>HTML Example</h1>
8   <p>Matt Dennis
9   <p>March 2020
10 </header>
11
12 <section>
13   <h1>Introduction</h1>
14   <p>Paragraph Text.
15   <p>Isnt it ..pretty?
16     <em>Less so.</em>
17 </section>
```

*Examples adapted from Lecture01 slides.*

## 1.3 Worksheet Task 2 & 3

Using the template provided, I created the following page.

```
LANGUAGE: HTML
1 <!doctype html>
2 <title>Thomas Boxall</title>
3 <h1>Thomas Boxall</h1>
4 <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
1 <!DOCTYPE html>
2 <title>Modules</title>
3 <marquee><h1>Modules</h1></marquee>
```

```

4 <h2>M22731 - Comp Tutorial Level 4</h2>
5 <p>A philosophical approach to learning and Computer Science</p>
6
7 <h2>M30220 - Core Computing Concepts</h2>
8 <p>A 4-part module introducing the history of computing, the web, security and user experience.
   ↪ </p>
9
10 <h2>M30231 - Networks</h2>
11 <p>An introduction to networking.</p>
12
13 <h2>M30232 - Database Systems Development</h2>
14 <p>An introduction to Databases using PostgreSQL.</p>
15
16 <h2>M30299 - Programming</h2>
17 <p>The basic concepts of programming featuring Zelles Graphics package for Python.</p>
18
19 <h2>M30943 - Architecture and Operating Systems</h2>
20 <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
1 <!DOCTYPE html>
2
3 <html lang="en">
4   <head>
5     <title>CatPhotoApp</title>
6   </head>
7   <body>
8     <main>
9       <h1>CatPhotoApp</h1>
10      <section>
11        <h2>Cat Photos</h2>
12        <!-- TODO: Add link to cat photos -->
13        <p>Click here to view more <a target="_blank" href="https://freecatphotoapp.com">cat
   ↪ photos</a>.</p>
14        <a href="https://freecatphotoapp.com"></
   ↪ a>
15      </section>
16      <section>
17        <h2>Cat Lists</h2>
18        <h3>Things cats love:</h3>
19        <ul>
20          <li>cat nip</li>
21          <li>laser pointers</li>
22          <li>lasagna</li>
23        </ul>
24        <figure>
25          
26          <figcaption>Cats <em>love</em> lasagna.</figcaption>
27        </figure>
28        <h3>Top 3 things cats hate:</h3>
29        <ol>
30          <li>flea treatment</li>
31          <li>thunder</li>
32          <li>other cats</li>
33        </ol>
34        <figure>
35          
36          <figcaption>Cats <strong>hate</strong> other cats.</figcaption>
37        </figure>
38      </section>
39      <section>
40        <h2>Cat Form</h2>
41        <form action="https://freecatphotoapp.com/submit-cat-photo">

```

```

42     <fieldset>
43       <legend>Is your cat an indoor or outdoor cat?</legend>
44       <label><input id="indoor" type="radio" name="indoor-outdoor" value="indoor"
↪ checked> Indoor</label>
45       <label><input id="outdoor" type="radio" name="indoor-outdoor" value="outdoor">
↪ Outdoor</label>
46     </fieldset>
47     <fieldset>
48       <legend>What's your cat's personality?</legend>
49       <input id="loving" type="checkbox" name="personality" value="loving" checked> <
↪ label for="loving">Loving</label>
50       <input id="lazy" type="checkbox" name="personality" value="lazy"> <label for="
↪ lazy">Lazy</label>
51       <input id="energetic" type="checkbox" name="personality" value="energetic"> <
↪ label for="energetic">Energetic</label>
52     </fieldset>
53     <input type="text" name="catphotourl" placeholder="cat photo URL" required>
54     <button type="submit">Submit</button>
55   </form>
56 </section>
57 </main>
58 <footer>
59 <p>
60   No Copyright - <a href="https://www.freecodecamp.org">freeCodeCamp.org</a>
61 </p>
62 </footer>
63 </body>
64 </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
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>Cafe Menu</title>
7     <link href="styles.css" rel="stylesheet"/>
8   </head>
9   <body>
10    <div class="menu">
11      <main>
12        <h1>CAMPER CAFE</h1>
13        <p class="established">Est. 2020</p>
14        <hr>
15        <section>
16          <h2>Coffee</h2>
17          
18          <article class="item">
19            <p class="flavor">French Vanilla</p><p class="price">3.00</p>
20          </article>
21          <article class="item">
22            <p class="flavor">Caramel Macchiato</p><p class="price">3.75</p>
23          </article>
24          <article class="item">
25            <p class="flavor">Pumpkin Spice</p><p class="price">3.50</p>
26          </article>
27          <article class="item">
28            <p class="flavor">Hazelnut</p><p class="price">4.00</p>
29          </article>
30          <article class="item">

```

```

31     <p class="flavor">Mocha</p><p class="price">4.50</p>
32   </article>
33 </section>
34 <section>
35   <h2>Desserts</h2>
36   
37   <article class="item">
38     <p class="dessert">Donut</p><p class="price">1.50</p>
39   </article>
40   <article class="item">
41     <p class="dessert">Cherry Pie</p><p class="price">2.75</p>
42   </article>
43   <article class="item">
44     <p class="dessert">Cheesecake</p><p class="price">3.00</p>
45   </article>
46   <article class="item">
47     <p class="dessert">Cinnamon Roll</p><p class="price">2.50</p>
48   </article>
49 </section>
50 </main>
51 <hr class="bottom-line">
52 <footer>
53   <p>
54     <a href="https://www.freecodecamp.org" target="_blank">Visit our website</a>
55   </p>
56   <p class="address">123 Free Code Camp Drive</p>
57 </footer>
58 </div>
59 </body>
60 </html>

```

## CSS:

```

LANGUAGE: CSS
1 body {
2   background-image: url(https://cdn.freecodecamp.org/curriculum/css-cafe/beans.jpg);
3   font-family: sans-serif;
4   padding: 20px;
5 }
6 h1 {
7   font-size: 40px;
8   margin-top: 0;
9   margin-bottom: 15px;
10 }
11 h2 {
12   font-size: 30px;
13 }
14 .established {
15   font-style: italic;
16 }
17 h1, h2, p {
18   text-align: center;
19 }
20 .menu {
21   width: 80%;
22   background-color: burlywood;
23   margin-left: auto;
24   margin-right: auto;
25   padding: 20px;
26   max-width: 500px;
27 }
28 img {
29   display: block;
30   margin-left: auto;
31   margin-right: auto;
32   margin-top: -25px;
33 }
34 hr {
35   height: 2px;
36   background-color: brown;
37   border-color: brown;
38 }
39 .bottom-line {

```

```
40 margin-top: 25px;
41 }
42 h1, h2 {
43   font-family: Impact, serif;
44 }
45 .item p {
46   display: inline-block;
47   margin-top: 5px;
48   margin-bottom: 5px;
49   font-size: 18px;
50 }
51 .flavor, .dessert {
52   text-align: left;
53   width: 75%;
54 }
55 .price {
56   text-align: right;
57   width: 25%;
58 }
59 /* FOOTER */
60 footer {
61   font-size: 14px;
62 }
63 .address {
64   margin-bottom: 5px;
65 }
66 a {
67   color: black;
68 }
69 a:visited {
70   color: black;
71 }
72 a:hover {
73   color: brown;
74 }
75 a:active {
76   color: brown;
77 }
```

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
1 h1 {
2   text-align: center;
3 }
4 .container {
5   background-color: rgb(255, 255, 255);
6   padding: 10px 0;
7 }
8 .marker {
9   width: 200px;
10  height: 25px;
11  margin: 10px auto;
12 }
13 .cap {
14  width: 60px;
15  height: 25px;
16 }
17 .sleeve {
18  width: 110px;
19  height: 25px;
```



```
20 background-color: rgba(255, 255, 255, 0.5);
21 border-left: 10px double rgba(0, 0, 0, 0.75);
22 }
23 .cap, .sleeve {
24 display: inline-block;
25 }
26 .red {
27 background: linear-gradient(rgb(122, 74, 14), rgb(245, 62, 113), rgb(162, 27, 27));
28 box-shadow: 0 0 20px 0 rgba(83, 14, 14, 0.8);
29 }
30 .green {
31 background: linear-gradient(#55680D, #71F53E, #116C31);
32 box-shadow: 0 0 20px 0 #3B7E20CC;
33 }
34 .blue {
35 background: linear-gradient(hsl(186, 76%, 16%), hsl(223, 90%, 60%), hsl(240, 56%, 42%));
36 box-shadow: 0 0 20px 0 hsla(223, 59%, 31%, 0.8);
37 }
```

## HTML:

```
LANGUAGE: HTML
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Colored Markers</title>
7     <link rel="stylesheet" href="styles.css">
8   </head>
9   <body>
10    <h1>CSS Color Markers</h1>
11    <div class="container">
12      <div class="marker red">
13        <div class="cap"></div>
14        <div class="sleeve"></div>
15      </div>
16      <div class="marker green">
17        <div class="cap"></div>
18        <div class="sleeve"></div>
19      </div>
20      <div class="marker blue">
21        <div class="cap"></div>
22        <div class="sleeve"></div>
23      </div>
24    </div>
25  </body>
26 </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 RGB 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, `co11` is the first colour, `co12` 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

```
LANGUAGE: HTML
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>Registration Form</title>
6     <link rel="stylesheet" href="styles.css" />
7   </head>
8   <body>
9     <h1>Registration Form</h1>
10    <p>Please fill out this form with the required information</p>
11    <form method="post" action='https://register-demo.freecodecamp.org'>
12      <fieldset>
13        <label for="first-name">Enter Your First Name: <input id="first-name" name="first-name"
14        ↪ type="text" required /></label>
15        <label for="last-name">Enter Your Last Name: <input id="last-name" name="last-name"
16        ↪ type="text" required /></label>
17        <label for="email">Enter Your Email: <input id="email" name="email" type="email"
18        ↪ required /></label>
19        <label for="new-password">Create a New Password: <input id="new-password" name="new-
20        ↪ password" type="password" pattern="[a-z0-5]{8,}" required /></label>
21      </fieldset>
22      <fieldset>
23        <label for="personal-account"><input id="personal-account" type="radio" name="account-
24        ↪ type" class="inline" /> Personal Account</label>
25        <label for="business-account"><input id="business-account" type="radio" name="account-
26        ↪ type" class="inline" /> Business Account</label>
27        <label for="terms-and-conditions" name="terms-and-conditions">
28        ↪ <input id="terms-and-conditions" type="checkbox" required name="terms-and-conditions"
29        ↪ class="inline" /> I accept the <a href="https://www.freecodecamp.org/news/terms-of-
30        ↪ service/">terms and conditions</a>
31      </label>
32    </fieldset>
33    <fieldset>
34      <label for="profile-picture">Upload a profile picture: <input id="profile-picture" type
35      ↪ ="file" name="file" /></label>
36      <label for="age">Input your age (years): <input id="age" type="number" name="age" min="
37      ↪ 13" max="120" /></label>
38      <label for="referrer">How did you hear about us?
39        <select id="referrer" name="referrer">
40          <option value="">(select one)</option>
41          <option value="1">freeCodeCamp News</option>
42          <option value="2">freeCodeCamp YouTube Channel</option>
43          <option value="3">freeCodeCamp Forum</option>
44          <option value="4">Other</option>
45        </select>
46      </label>
47      <label for="bio">Provide a bio:
48      <textarea id="bio" name="bio" rows="3" cols="30" placeholder="I like coding on the
```

```
↔ beach..."></textarea>
39 </label>
40 </fieldset>
41 <input type="submit" value="Submit" />
42 </form>
43 </body>
44 </html>
```

LANGUAGE: CSS

```
1 body {
2   width: 100%;
3   height: 100vh;
4   margin: 0;
5   background-color: #1b1b32;
6   color: #f5f6f7;
7   font-family: Tahoma;
8   font-size: 16px;
9 }
10
11 h1, p {
12   margin: 1em auto;
13   text-align: center;
14 }
15
16 form {
17   width: 60vw;
18   max-width: 500px;
19   min-width: 300px;
20   margin: 0 auto;
21   padding-bottom: 2em;
22 }
23
24 fieldset {
25   border: none;
26   padding: 2rem 0;
27   border-bottom: 3px solid #3b3b4f;
28 }
29
30 fieldset:last-of-type {
31   border-bottom: none;
32 }
33
34 label {
35   display: block;
36   margin: 0.5rem 0;
37 }
38
39 input,
40 textarea,
41 select {
42   margin: 10px 0 0 0;
43   width: 100%;
44   min-height: 2em;
45 }
46
47 input, textarea {
48   background-color: #0a0a23;
49   border: 1px solid #0a0a23;
50   color: #ffffff;
51 }
52
53 .inline {
54   width: unset;
55   margin: 0 0.5em 0 0;
56   vertical-align: middle;
57 }
58
59 input[type="submit"] {
60   display: block;
61   width: 60%;
62   margin: 1em auto;
63   height: 2em;
```

```

64     font-size: 1.1rem;
65     background-color: #3b3b4f;
66     border-color: white;
67     min-width: 300px;
68 }
69
70 input[type="file"] {
71     padding: 1px 2px;
72 }
73
74 a{
75     color: #dfdfe2
76 }

```

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.

## 2.1.2 Learnn The CSS Box Model By Building A Rothko Painting

```

LANGUAGE: HTML
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>Rothko Painting</title>
6     <link href="./styles.css" rel="stylesheet">
7   </head>
8   <body>
9     <div class="frame">
10      <div class="canvas">
11        <div class="one"></div>
12        <div class="two"></div>
13        <div class="three"></div>
14      </div>
15    </div>
16  </body>
17 </html>

```

```

LANGUAGE: CSS
1 .canvas {
2   width: 500px;
3   height: 600px;
4   background-color: #4d0f00;
5   overflow: hidden;
6   filter: blur(2px);
7 }
8
9 .frame {
10  border: 50px solid black;
11  width: 500px;
12  padding: 50px;
13  margin: 20px auto;
14 }
15
16 .one {
17  width: 425px;
18  height: 150px;
19  background-color: #efb762;
20  margin: 20px auto;
21  box-shadow: 0 0 3px 3px #efb762;
22  border-radius: 9px;
23  transform: rotate(-0.6deg);

```

```

24 }
25
26 .two {
27   width: 475px;
28   height: 200px;
29   background-color: #8f0401;
30   margin: 0 auto 20px;
31   box-shadow: 0 0 3px 3px #8f0401;
32   border-radius: 8px 10px;
33   transform: rotate(0.4deg);
34 }
35
36 .one, .two {
37   filter: blur(1px);
38 }
39
40 .three {
41   width: 91%;
42   height: 28%;
43   background-color: #b20403;
44   margin: auto;
45   filter: blur(2px);
46   box-shadow: 0 0 5px 5px #b20403;
47   border-radius: 30px 25px 60px 12px;
48   transform: rotate(-0.2deg);
49 }

```

## Box Model

The content is the HTML elements which we are looking at on the page.

The padding is space which wraps around the content. It can be used to control white-space around elements

The border encapsulates the padding, this has properties which can be used to show or hide it in various states.

The Margin is the area outside of the box and can be used to control the space between both boxes or elements.

## CSS Properties

The `filter: blur([val])` property allows a blur to be placed over a element. This gives a visual blur appearance. As the CSS styles stack on top of each other, the `blur` effects add. This means that if you had `blur(1px)` on your body and you wanted a blur value of `3px` on a particular element within the body, you would need to give that element a value of `blur(2px)`.

`border-radius` can either accept a single value which it will use for all for corners or, alternatively, you can feed it 4 values where each corresponds to a different corner. Their order is: top-left, top-right, bottom-right, bottom-left.

`transform: rotate([val])` can be used to rotate an element by the specified value. Degrees can be represented with the unit `deg`.

### 2.1.3 CSS Flexbox Photo Gallery

LANGUAGE: HTML

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Photo Gallery</title>
7     <link rel="stylesheet" href="styles.css">
8   </head>
9   <body>

```

```
10 <header class="header">
11   <h1>css flexbox photo gallery</h1>
12 </header>
13 <div class="gallery">
14   
16   
18   
20   
22   
24   
26   
28   
30   
32 </div>
33 </body>
34 </html>
```

LANGUAGE: CSS

```
1 * {
2   box-sizing: border-box;
3 }
4
5 body {
6   margin: 0;
7   font-family: sans-serif;
8   background: #f5f6f7;
9 }
10
11 .header {
12   text-align: center;
13   text-transform: uppercase;
14   padding: 32px;
15   background-color: #0a0a23;
16   color: #fff;
17   border-bottom: 4px solid #fdb347;
18 }
19
20 .gallery {
21   display: flex;
22   flex-direction: row;
23   flex-wrap: wrap;
24   justify-content: center;
25   align-items: center;
26   gap: 16px;
27   max-width: 1400px;
28   margin: 0 auto;
29   padding: 20px 10px;
30 }
31
32 .gallery img {
33   width: 100%;
34   max-width: 350px;
35   height: 300px;
36   object-fit: cover;
37   border-radius: 10px;
38 }
39
40 .gallery::after {
41   content: "";
42   width: 350px;
43 }
```

## Flexbox

Unless otherwise specified, all properties shown below are added to the container which will become the flexbox (e.g. `<div>`).

Flexbox is a one dimensional CSS layout that can control the way items are spaced out and aligned within the container. To use it, add `display: flex;` to the element which you wish to be a flexbox.

The main axis of the flexbox is defined by the `flex-direction` property. It has 4 possible values (`row`, `row-reverse`, `column`, `column-reverse`)

The `flex-wrap` property will determine how the items within the flex container will behave when the container is too smaller. `wrap` allows the items to wrap onto a new line; `no-wrap` prevents the items from wrapping and shrinks them if needed, this is the default.

The `justify-content` property determines how the items inside the flexbox align.

The `align-items` property positions the items in the flex box along the cross axis (the opposite axis to the to that specified with `flex-direction`).

The `object-fit` property determines how images should behave. The value `cover` will specify that images should fit the specified width and height and should be cropped to fit so they can be left in their original aspect ratio. NB: this gets added to the image CSS, not the Flexbox css.

The `gap` property allows the gap sizes between the rows and columns to be specified. `gap` is shorthand for `row-gap` and `column-gap`.

The `::after` pseudo-element creates an element that is the last child of the selected element. They can be used to add an empty element after the last.

### 2.1.4 Typography Nutrition Label

```

LANGUAGE: HTML
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>Nutrition Label</title>
6     <link href="https://fonts.googleapis.com/css?family=Open+Sans:400,700,800" rel="stylesheet"
7     ↪ >
8     <link href="./styles.css" rel="stylesheet">
9   </head>
10  <body>
11    <div class="label">
12      <header>
13        <h1 class="bold">Nutrition Facts</h1>
14        <div class="divider"></div>
15        <p>8 servings per container</p>
16        <p class="bold">Serving size <span class="right">2/3 cup (55g)</span></p>
17      </header>
18      <div class="divider lg"></div>
19      <div class="calories-info">
20        <p class="bold sm-text">Amount per serving</p>
21        <h1>Calories <span class="right">230</span></h1>
22      </div>
23      <div class="divider md"></div>
24      <div class="daily-value sm-text">
25        <p class="right bold no-divider">% Daily Value *</p>
26        <div class="divider"></div>
27        <p><span class="bold">Total Fat</span> 8g <span class="bold right">10%</span></p>
28        <p class="indent no-divider">Saturated Fat 1g <span class="bold right">5%</span></p>
29        <div class="divider"></div>
30        <p class="indent no-divider"><i>Trans</i> Fat 0g</p>
31        <div class="divider"></div>
32        <p><span class="bold">Cholesterol</span> 0mg <span class="right bold">0%</span></p>
33        <p><span class="bold">Sodium</span> 160mg <span class="right bold">7%</span></p>
34        <p><span class="bold">Total Carbohydrate</span> 37g <span class="right bold">13%</span>
35        ↪ </p>
36        <p class="indent no-divider">Dietary Fiber 4g</p>
37      </div>
38    </div>

```



```

36     <p class="indent no-divider">Total Sugars 12g</p>
37     <div class="divider dbl-indent"></div>
38     <p class="dbl-indent no-divider">Includes 10g Added Sugars <span class="right bold">20%
↪ </span>
39     <div class="divider"></div>
40     <p class="no-divider"><span class="bold">Protein</span> 3g</p>
41     <div class="divider lg"></div>
42     <p>Vitamin D 2mcg <span class="right">10%</span></p>
43     <p>Calcium 260mg <span class="right">20%</span></p>
44     <p>Iron 8mg <span class="right">45%</span></p>
45     <p class="no-divider">Potassium 235mg <span class="right">6%</span></p>
46     </div>
47     <div class="divider md"></div>
48     <p class="note">* The % Daily Value (DV) tells you how much a nutrient in a serving of
↪ food contributes to a daily diet. 2,000 calories a day is used for general nutrition
↪ advice.</p>
49     </div>
50 </body>
51 </html>

```

LANGUAGE: CSS

```

1 * {
2   box-sizing: border-box;
3 }
4
5 html {
6   font-size: 16px;
7 }
8
9 body {
10  font-family: 'Open Sans', sans-serif;
11 }
12
13 .label {
14  border: 2px solid black;
15  width: 270px;
16  margin: 20px auto;
17  padding: 0 7px;
18 }
19
20 header h1 {
21  text-align: center;
22  margin: -4px 0;
23  letter-spacing: 0.15px
24 }
25
26 p {
27  margin: 0;
28 }
29
30 .divider {
31  border-bottom: 1px solid #888989;
32  margin: 2px 0;
33  clear: right;
34 }
35
36 .bold {
37  font-weight: 800;
38 }
39
40 .right {
41  float: right;
42 }
43
44 .lg {
45  height: 10px;
46 }
47
48 .lg, .md {
49  background-color: black;
50  border: 0;
51 }

```

```
52
53 .md {
54   height: 5px;
55 }
56
57 .sm-text {
58   font-size: 0.85rem;
59 }
60
61 .calories-info h1 {
62   margin: -5px -2px;
63   overflow: hidden;
64 }
65
66 .calories-info span {
67   font-size: 1.2em;
68   margin-top: -7px;
69 }
70
71 .indent {
72   margin-left: 1em;
73 }
74
75 .dbl-indent {
76   margin-left: 2em;
77 }
78
79 .daily-value p:not(.no-divider) {
80   border-bottom: 1px solid #888989;
81 }
82
83 .note {
84   font-size: 0.6rem;
85   margin: 5px 0;
86   padding: 0 8px 0 8px;
87   text-indent: -8px;
88 }
```

`float` is used to place an element on the left or right of the container, allowing the other content (e.g. text) to wrap around it.

The `:not` pseudo-selector can be used to select all elements that do not match the given CSS rule.