\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**BLOG**

<!DOCTYPE html>

<html>

  <head>

    <title>Everyday with Isa</title>

  </head>

  <body>

    <a href="#contact"><img src="https://content.codecademy.com/courses/learn-html/elements-and-structure/profile.jpg"></a>

    <h3>by Isabelle Rodriguez | 1 day ago</h3>

    <h1>An Insiders Guide to NYFW</h1>

    <img src="https://content.codecademy.com/courses/learn-html/elements-and-structure/image-one.jpeg">

    <p>

      <a href="https://en.wikipedia.org/wiki/New\_York\_Fashion\_Week" target="\_blank">NYFW</a> can be both amazingly fun & incredibly overwhelming,   especially if youve never been. Luckily, Im here to give you an insiders guide and make your first show a pleasurable experience. By taking my tips and tricks, and following your gut, youll have an unforgettable experience!

    </p>

    <h2>Getting Tickets & Picking the Shows</h2>

    <img src="https://content.codecademy.com/courses/learn-html/elements-and-structure/image-two.jpeg">

    <p>

    If youre lucky or connected you can get an invite, sans the price tag. But I wasnt so lucky or connected my first 2 years so Im here to help you out. First, plan out which shows are most important to you and make a schedule and this is a biggie: SET A BUDGET. If youre worrying about blowing your cash the whole time you wont have fun. Then check out prices, days, and times and prioritize the designers you want to see most. Lastly, purchase your tickets and get excited

    </p>

    <h2>Dressing for the Shows</h2>

    <img src="https://content.codecademy.com/courses/learn-html/elements-and-structure/image-three.jpeg">

    <p>Always be true to your own sense of style, if you dont youll be uncomfortable the whole time and it will show. Remember, NYFW is about expressing yourself and taking in what the designers have chosen to express through their new lines. Also its important to wear shoes youll be comfortable in all day. Obviously you want to look good, but youll be on your feet all day long, so be prepared.</p>

    <h4>Related Content</h4>

    <ul>

      <li>How To Style Boyfriend Jeans</li>

      <li>When Print Is Too Much</li>

      <li>The Overalls Trend</li>

      <li>Falls It Color: Blush</li>

    </ul>

    <div id="contact">

      <p>

        <strong>email:</strong> isa@fashionblog.com | <strong>phone:</strong> 917-555-1098 | <strong>address:</strong> 371 284th St, New York, NY, 10001

      </p>

    </div>

  </body>

</html>

**MEDIA:**

<!DOCTYPE html>

<html>

  <head>

    <link rel="stylesheet" type="text/css" href="style.css">

  </head>

  <body>

    <header>

      <h1>Navigational Links</h1>

      <nav>

        <ul>

          <li><a href="#home">Home</a></li>

          <li><a href="#posts">Posts</a></li>

          <li><a href="#contact">Contact</a></li>

        </ul>

      </nav>

    </header>

    <main>

      <section>

        <article>

          <h2>Facts About Dogs</h2>

          <p>

          Dogs have a sense of time. It's been proven that they know the difference between a hour and five. If conditioned to, they can predict future events, such as regular walk times.

          </p>

        </article>

        <aside>

          <p>A study was conducted on dogs being away from their owners for varying hours and the studies show that dogs who were away from their owners the longest showed the greatest amount of affection!

          </p>

        </aside>

      </section>

      <figure>

        <img src="https://content.codecademy.com/courses/SemanticHTML/dogimage.jpeg"/>

        <figcaption>A cute dog.</figcaption>

      </figure>

      <audio controls>

        <source src="https://content.codecademy.com/courses/SemanticHTML/dogBarking.mp3" type="audio/mp3">

      </audio>

      <video src="https://content.codecademy.com/courses/SemanticHTML/dog-video.mp4" controls>

      </video>

      <embed src="https://content.codecademy.com/courses/SemanticHTML/dog-on-beach.gif"/>

    </main>

    <footer>

      <p>Contact me at +1 234 567 8910 </p>

    </footer>

  </body>

</html>

**BLOG 2 :**

<!DOCTYPE html>

<html>

  <head>

    <link rel="stylesheet" href="style.css">

  </head>

  <body>

    <nav>

      <ul>

        <li>  <a href="#blog"> Blog </a> </li>

        <li>  <a href="#media"> Media </a> </li>

        <li>  <a href="#footer"> About </a> </li>

      </ul>

    </nav>

    <header>

      <h1> New York City</h1>

    </header>

    <main>

      <section id="blog">

        <article>

          <p> New York City is made up of five boroughts which include Queens, Manhattan, Brooklyn, the Bronx, and Staten Island. The city is the home of approximately 8 million people. In 1876, France gifted the City of New York what is known as the Statue of Liberty, which is currently located on Liberty Island and commonly visited by tourists. However, it took 10 years to assemble and therefore wasn’t unveiled until 1886. Another tourist destination is Times Square. Times Square is commonly known for the big buildings, Broadway shows, and bright neon signs. This famous location was named after The New York Times after the Times moved to that location. Prior to that, it was named Longacre Square. New York City is also known for its bridges that connect the boroughs and allow ease of transportation. </p>

        </article>

      </section>

      <figure>

        <img src="https://content.codecademy.com/courses/Semantic%20HTML/statue-of-liberty.jpeg"/>

        <figcaption> This is the Statue of Liberty, a popular tourist attraction located on Liberty Island</figcaption>

      </figure>

      <aside> <p> New York City is very popular for the variety of great food it has. Some of the top food items in NYC include :</p>

      <ol>

        <li> Pizza </li>

        <li> Bagels </li>

        <li> Burgers and Sandwiches </li>

        <li> Ramen </li>

        <li> Tacos </li>

        <li> Pasta </li>

        <li> Desserts </li>

      </ol>

      </aside>

      <section id="media">

        <article>

          <h2>The scenery in NYC</h2>

          <p>

            While the view in the city is beautiful, the sounds are not as lvoely. Below you will see an exemple of the view and the osunds you will deal with in NYC on a daily basis.

          </p>

        </article>

        <video src="https://content.codecademy.com/courses/Semantic%20HTML/nyc-skyline-timelapse.mp4" controls>

        </video>

        <embed src="https://content.codecademy.com/courses/Semantic%20HTML/nyc-skyline.jpeg">

        <audio src="https://content.codecademy.com/courses/Semantic%20HTML/nyc-sounds.mov" controls>

        </audio>

      </section>

    </main>

    <footer id="footer">

      <p> "Posted by : moroccan"</p>

      <p> contact information: blogger@test.com </p>

    </footer>

  </body>

</html>

<link href='./style.css' rel='stylesheet'>

<style>

    p{

      color: green;

    }

  </style>

* Providing a dynamic user experience by offering content that responds to user input without forcing the page to reload. In the early web, user input would typically take the user to a new page — and they would have to wait for the new page to load. In Web 2.0, websites could just update selected regions of the page, avoiding the interruption caused by reloading.
* jQuery was the first JavaScript framework that could fetch data while the web page is running.
* The rise of web frameworks that connected to databases, like Spring, Django, and Ruby-on-Rails, enabled user-generated content to effectively be created, stored, and displayed.

**Responsive Web Design**

The rise of *responsive web design* has changed how websites are built. Responsive web design was enabled by additions to the CSS language, like media queries and relative units. These additions allow the presentation of websites to adjust based on the size of the window in which they are displayed.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Pseudo-class**

You may have observed how the appearance of certain elements can change, or be in a different state, after certain user interactions. For instance:

* When you click on an <input> element, and a blue border is added showing that it is in *focus*.
* When you click on a blue <a> link to *visit* to another page, but when you return the link’s text is purple.
* When you’re filling out a form and the submit button is grayed out and *disabled*. But when all of the fields have been filled out, the button has color showing that it’s *active*.

These are all examples of pseudo-class selectors in action! In fact, :focus, :visited, :disabled, and :active are all pseudo-classes. Factors such as user interaction, site navigation, and position in the document tree can all give elements a different state with pseudo-class.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<!DOCTYPE html>

<html>

<head>

  <title>Vacation World</title>

  <link href='style.css' rel='stylesheet'>

</head>

<body>

  <img src='https://content.codecademy.com/courses/freelance-1/unit-2/explorer.jpeg' />

  <h1 class='title uppercase' id='article-title'>Top Vacation Spots</h1>

  <h5 class='author-class' id='author-id'>By: Stacy Gray</h5>

  <h6 id='publish-time'>Published: 2 Days Ago</h6>

  <p>The world is full of fascinating places. Planning the perfect vacation involves packing up, leaving home, and experiencing something new.</p>

  <h2 class='destination heading-background'>1. Florence, Italy</h2>

  <div class='description'>A city-size shrine to the Renaissance, Florence offers frescoes, sculptures, churches, palaces, and other monuments from the richest cultural flowering the world has known. Names from its dazzling historical past; Dante, Michelangelo, Galileo, Machiavelliare are some of the most resonant of the medieval age. <a href='https://www.nationalgeographic.com/travel/destination/florence' target='\_blank'>Learn More</a>.

    <h5>Top Attractions</h5>

    <ul>

      <li>Museums</li>

      <li>Bike Tours</li>

      <li>Historical Monuments</li>

    </ul>

  </div>

  <h2 class='destination heading-background'>2. Beijing, China</h2>

  <div class='description'>A city in the midst of reinventing itself and continuing to build on the success of the 2008 Summer Olympics, Beijing is a place of frenzied construction. New housing, new roads, and new sports venues seem to spring up overnight. At the same time, the capital of the Peoples Republic of China remains an epicenter of tradition, with the treasures of nearly 2,000 years as the imperial capital still on view in the famed Forbidden City and in the luxuriant pavilions and gardens of the Summer Palace.

    <a href='https://www.nationalgeographic.com/travel/destination/beijing' target='\_blank'>Learn More</a>.

    <h5>Top Attractions</h5>

    <ul>

      <li>Biking</li>

      <li>Historical Sites</li>

      <li>Restaurants and Dining</li>

    </ul>

  </div>

  <h2 class='destination heading-background'>3. Seoul, South Korea</h2>

  <div class='description'>The Korean capital is a city of contrasts. Fourteenth-century city gates squat in the shadow of 21st-century skyscrapers, while the broad Han River is back-dropped by granite mountains rising in the city center complete with alpine highways speeding around their contours and temples nestling among their crags. Fashionable, gadget-laden youths battle for sidewalk space with fortune-tellers and peddlers, while tiny neighborhoods of traditional cottages contrast with endless ranks of identical apartments.

    <a href='https://www.nationalgeographic.com/travel/destination/seoul' target='\_blank'>Learn More</a>.

    <h5>Top Attractions</h5>

    <ul>

      <li>Parasailing</li>

      <li>Segway Tours</li>

      <li>Spas and Resorts</li>

    </ul>

  </div>

  <h2 class='heading-background'> More Destinations </h2>

  <ul>

    <li><h4 class='destination'>Jackson Hole, Wyoming</h4></li>

    <li><h4 class='destination'>Cape Town, South Africa</h4></li>

    <li><h4 class='destination'>La Paz, Bolivia</h4></li>

  </ul>

  <p>&mdash;Best of luck with your travels, and be sure to send pictures and stories. We'd love to hear them!</p>

</body>

</html>

CSS

\* {

  border: 1px solid red;

}

p {

  color: green;

}

h1 {

  color: maroon;

}

.title {

  color: teal;

}

.uppercase {

  text-transform: uppercase;

}

#article-title {

  font-family: cursive;

}

a[href\*='florence'] {

  color: lightgreen;

}

a[href\*='beijing'] {

  color: lightblue;

}

a[href\*='seoul'] {

  color: lightpink;

}

a:hover {

  color:darkorange;

}

.heading-background {

  background-color: aqua;

}

#publish-time {

  color: lightgray;

}

h5 {

  color: yellow;

}

.author-class {

  color: pink;

}

#author-id {

  color: cornflowerblue;

}

h2.destination {

  font-family: Tahoma;

}

.description h5 {

  color: blueviolet;

}

li h4 {

  color: gold;

}

h4 {

  color: dodgerblue;

}

li,

h5 {

  font-family: monospace;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

HTML

<!DOCTYPE html>

<html>

<head>

  <title>The Rise of Soccer in The US</title>

  <link href='style.css' rel='stylesheet'>

  <link href='style-library.css' rel='stylesheet'>

</head>

<body>

  <div class='content'>

    <img src='https://content.codecademy.com/courses/web-101/unit-4/htmlcss1-img\_writer-avatar.jpg' class='writer-img'>

    <h3 class='byline'>Article By: Jane Dover</h3>

    <h1>How the Rise of Soccer in the US Is Changing the Face of Youth Sports</h1>

    <h2>The focus on soccer in youth sports programs is exploding nation-wide</h2>

    <p>When the first World Cup arrived in the US in the 90's everyone officially declared that soccer was it. Well it's taken it's time but we can definitely see the influence of soccer, especially women's soccer, across the US. This year, 3 million kids

      played in youth soccer leagues with 2/3 of those leagues for girls. In fact, in the 12-17 age range the MLS has surpassed the MLB and NFL in popularity.</p>

    <p>Part of this meteoric rise can be attributed to the impressively soaring ad dollars being pumped into the Women's World Cup games in 2014. The women's games generated $40 million for Fox, that's definitely not chump change. And those advertisers,

      like ATT, Coca Cola, Verizon, Nike, Visa, and other heavy hitters, are working to make sure they see those numbers grow year after year by investing in youth soccer facilities and promoting programs across the country. </p>

    <p>Now that big business is involved you can be assured you'll see a continued rise in popularity in soccer across the country for years to come. </p>

  </div>

  <div class='image'>

    <p class='caption'>The local semi- pro soccer team in Seattle, WA plays an international friendly</p>

  </div>

</body>

</html>

CSS

body {

  /\* Old browsers \*/

  background: #141E30;

  /\* Chrome 10-25, Safari 5.1-6 \*/

  background: -webkit-linear-gradient(-45deg, #35577D, #141E30);

  /\* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+ \*/

  background: linear-gradient(-45deg, #35577D, #141E30);

  margin: 0;

  padding: 0;

}

h1 {

  color: #FFF !important;

  font-size: 2em;

  padding-top: 100px;

  width: 100%;

  font-family: Georgia;

  text-align: center;

}

h2 {

  border-bottom: 1px solid rgba(255, 255, 255, 0.5);

  color: rgba(255, 255, 255, 0.5);

  font-weight: 100;

  font-size: 22px;

  line-height: 24px;

  padding-bottom: 30px;

  text-align: left;

  width: 70%;

  font-family: Georgia;

}

p {

  color: aliceblue;

  line-height: 1.3em;

  text-align: left;

  width: 100%;

  font-family: Helvetica;

  font-size: 18px;

  font-weight: bold;

}

.byline {

  font-family: Helvetica;

  color: rgba(255, 255, 255, 0.5);

  float: left;

  font-size: 14px;

  padding-left: 10px;

  text-transform: uppercase;

}

.caption {

  display: block;

  font-family: 'Playfair Display', serif;

  font-size: 14px;

  font-style: italic;

  line-height: 14px;

  margin-left: 20px;

  padding: 10px;

  position: relative;

  top: 80%;

  width: 60%;

  background-color: white;

  color: black;

}

.content {

  padding: 40px;

}

.image {

  background-image: url('https://content.codecademy.com/courses/freelance-1/unit-2/soccer.jpeg');

  background-size: cover;

  background-position: center;

  height: 300px;

}

.writer-img {

  -webkit-box-shadow: 5px 0px 5px 0px rgba(0, 0, 50, 0.97);

  -moz-box-shadow: 5px 0px 5px 0px rgba(0, 0, 50, 0.97);

  box-shadow: 5px 0px 5px 0px rgba(0, 0, 50, 0.97);

  float: left;

  width: 50px;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

p.citation a.external-link{

  color: SeaGreen;

}

h1, h2, p, li{

  font-family: Helvetica;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Borders**

A *border* is a line that surrounds an element, like a frame around a painting. Borders can be set with a specific width, style, and color:

* width—The thickness of the border. A border’s thickness can be set in pixels or with one of the following keywords: thin, medium, or thick.
* style—The design of the border. Web browsers can render any of [10 different styles](https://developer.mozilla.org/en-US/docs/Web/CSS/border-style#Values). Some of these styles include: none, dotted, and solid.
* color—The color of the border. Web browsers can render colors using a few different formats, including [140 built-in color keywords](https://developer.mozilla.org/en-US/docs/Web/CSS/color_value).

p {  
  border: 3px solid coral;  
}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Review**

In this lesson, we covered the four properties of the box model: height and width, padding, borders, and margins. Understanding the box model is an important step towards learning more advanced HTML and CSS topics. Let’s take a minute to review what you learned:

* The box model comprises a set of properties used to create space around and between HTML elements.
* The height and width of a content area can be set in pixels or percentages.
* Borders surround the content area and padding of an element. The color, style, and thickness of a border can be set with CSS properties.
* Padding is the space between the content area and the border. It can be set in pixels or percent.
* Margin is the amount of spacing outside of an element’s border.
* Horizontal margins add, so the total space between the borders of adjacent elements is equal to the sum of the right margin of one element and the left margin of the adjacent element.
* Vertical margins collapse, so the space between vertically adjacent elements is equal to the larger margin.
* margin: 0 auto horizontally centers an element inside of its parent content area, if it has a width.
* The overflow property can be set to display, hide, or scroll, and dictates how HTML will render content that overflows its parent’s content area.
* The visibility property can hide or show elements.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* {  
  box-sizing: border-box;  
}

**Review: Changing the Box Model**

In this lesson, you learned about an important limitation of the default box model: box dimensions are affected by border thickness and padding.

Let’s review what you learned:

1. In the default box model, box dimensions are affected by border thickness and padding.
2. The box-sizing property controls the box model used by the browser.
3. The default value of the box-sizing property is content-box.
4. The value for the new box model is border-box.
5. The border-box model is not affected by border thickness or padding.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

HTML

<!DOCTYPE html>

<html>

<head>

  <title>Davie JR's Menu</title>

  <link href="https://fonts.googleapis.com/css?family=Roboto:100,500,700|Oswald:300,400,700" rel="stylesheet">

  <link rel="stylesheet" type="text/css" href="reset.css">

  <link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

  <!-- Navigation Section -->

  <nav>

    <img src="https://content.codecademy.com/courses/web-101/unit-6/htmlcss1-img\_burger-logo.svg" />

    <span><a href="#">MENU</a></span>

    <span><a href="#">NUTRITION</a></span>

    <span><a href="#">ORDER</a></span>

    <span><a href="#">LOCATIONS</a></span>

  </nav>

  <!-- Content Section -->

  <div class="content">

    <!-- Content Header -->

    <div class="header">

      <div class="testh1">

        <h1>BBQ BACON BURGER</h1>

      </div>

    </div>

    <!-- Content Body -->

    <div class="body">

      <p>

        Our BBQ Bacon Burger features our special house ground blend of wagyu and sirloin, spiced perfectly, and finished off with just a drop of white truffle oil. A butter grilled brioche bun layered with roasted red onion, perfectly crispy pork belly, and our hickory smoked BBQ sauce.

      </p>

      <!-- Order Button -->

      <a href="#" class="button">ORDER NOW</a>

      <!-- Nutrition Information -->

      <ul class="nutrition">

        <li>

          <span class="category">CALORIES</span>

          <span class="value">678</span>

        </li>

        <li>

          <span class="category">FAT</span>

          <span class="value">32</span>

        </li>

        <li>

          <span class="category">PROTEIN</span>

          <span class="value">8</span>

        </li>

        <li>

          <span class="category">CARBOHYDRATES</span>

          <span class="value">34</span>

        </li>

        <li>

          <span class="category">SODIUM</span>

          <span class="value">112</span>

        </li>

      </ul>

    </div>

  </div>

</body>

</html>

CSS

/\* Universal Styles \*/

body {

  background-image: url("https://content.codecademy.com/courses/web-101/unit-6/htmlcss1-img\_foodlogo.png");

  text-align: center;

  font-family: 'Roboto', sans-serif;

  font-size: 18px;

}

a {

  text-decoration: none;

}

/\* Navigation \*/

nav {

  text-align: center;

}

nav img {

  display: block;

  width: 180px;

  margin: auto;

}

nav span {

  display: block;

  font-size: 16px;

  font-weight: 100;

  letter-spacing: 2px;

  margin: 10px 0px;

}

nav a {

  color: #666666;

}

/\* Content Container \*/

.content {

  width: 100%;

  height: 500px;

  margin: 10px auto;

  overflow: scroll;

}

/\* Content Header \*/

.header {

  background-image: url("https://content.codecademy.com/courses/web-101/unit-6/htmlcss1-img\_burgerphoto.jpeg");

  background-position: center;

  background-size: cover;

  height: 320px;

}

.header h1 {

  background-color: #05A8AA;

  color: #FFF;

  font-family: 'Oswald', sans-serif;

  font-size: 40px;

  font-weight: 300;

  line-height: 40px;

  width: 65%;

  margin: 0 auto;

  padding: 20px;

}

/\* Content Body \*/

.content .body {

  width: 90%;

  margin: 0 auto;

}

.body p {

  color: #333333;

  font-weight: 100;

  line-height: 34px;

  width: 90%;

  margin-top: 18px;

}

/\* Content Button \*/

.button {

  border-radius: 4px;

  color: #05A8AA;

  display: block;

  font-weight: 700;

  width: 200px;

  padding: 20px;

  margin: 40px auto;

  border: 1px solid blue;

}

.button:hover {

  background-color: #05A8AA;

  border: 1px solid #05A8AA;

  color: #FFF;

}

/\* Content Nutrition \*/

ul.nutrition {

  padding: 20px;

}

ul.nutrition li {

  display: inline-block;

  background-color: #05A8AA;

  list-style: none;

  width: 200px;

  padding: 10px 20px;

  margin-bottom: 3px;

}

.nutrition .category {

  color: white;

  font-weight: 100;

  letter-spacing: 2px;

  display: block;

}

.nutrition .value {

  color: white;

  font-size: 26px;

  font-weight: 700;

  letter-spacing: 2px;

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**POSITION: RELATIVE**

One way to modify the default position of an element is by setting its position property to relative.

This value allows you to position an element *relative* to its default static position on the web page.

.green-box {  
  background-color: green;  
  position: relative;  
}

Although the code in the example above instructs the browser to expect a relative positioning of the .green-box element, it does not specify where the .green-box element should be positioned on the page. This is done by accompanying the position declaration with one or more of the following *offset properties* that will move the element away from its default static position:

* top - moves the element down from the top.
* bottom - moves the element up from the bottom.
* left - moves the element away from the left side (to the right).
* right - moves the element away from the right side (to the left).

You can specify values in pixels, ems, or percentages, among others, to dial in exactly how far you need the element to move. It’s also important to note that offset properties will not work if the element’s position property is the default static.

.green-box {  
  background-color: green;  
  position: relative;  
  top: 50px;  
  left: 120px;  
}

In the example above, the element of green-box class will be moved down 50 pixels, and to the right 120 pixels, from its default static position. The image below displays the new position of the box.

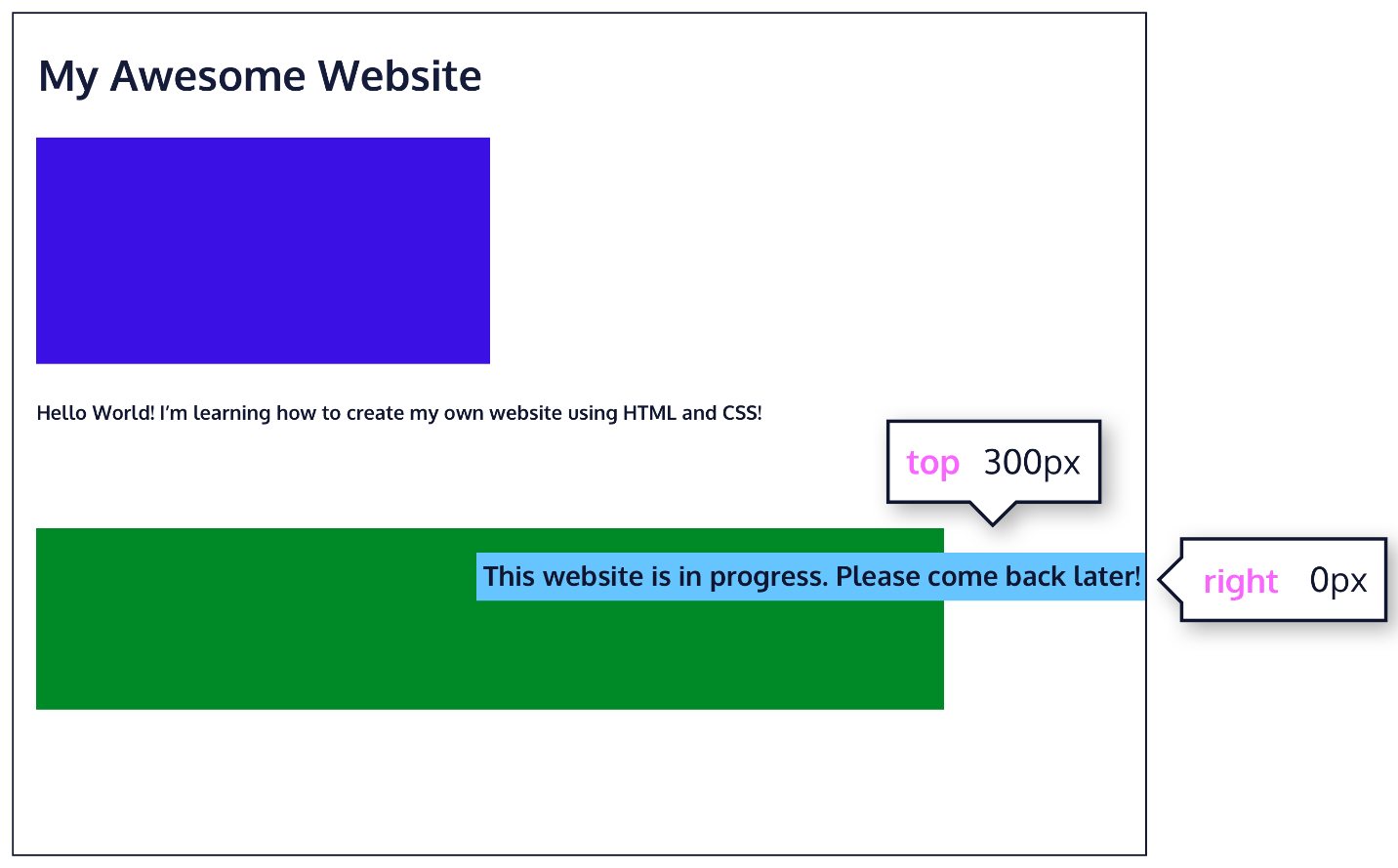


Offsetting the relative element will not affect the positioning of other elements.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**POSITION: ABSOLUTE**

Another way of modifying the position of an element is by setting its position to absolute.

When an element’s position is set to absolute, all other elements on the page will ignore the element and act like it is not present on the page. The element will be positioned relative to its closest positioned parent element, while offset properties can be used to determine the final position from there. Take a look at the image below:



The “This website is in progress. Please come back later!” text is displaced from its static position at the top left corner of its parent container. It has offset property declarations of top: 300px; and right: 0px;, positioning it 300 pixels down, and 0 pixels from the right side of the page.

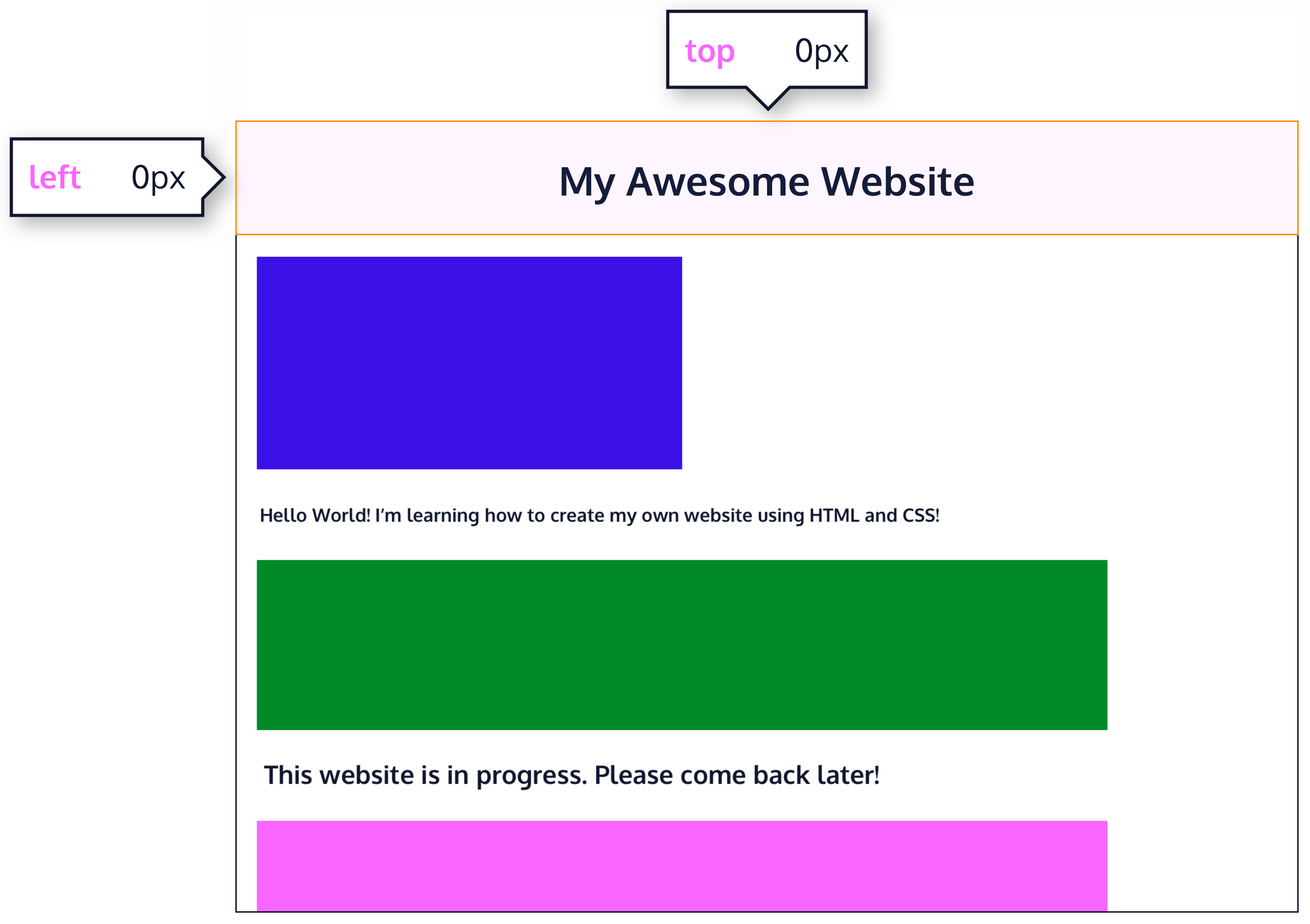
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**POSITION: FIXED**

When an element’s position is set to absolute, as in the last exercise, the element will scroll with the rest of the document when a user scrolls.

We can *fix* an element to a specific position on the page (regardless of user scrolling) by setting its position to fixed, and accompanying it with the familiar offset properties top, bottom, left, and right.

.title {  
  position: fixed;  
  top: 0px;  
  left: 0px;  
}

In the example above, the .title element will remain fixed to its position no matter where the user scrolls on the page, like in the image below:



This technique is often used for navigation bars on a web page.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Z-INDEX**

When boxes on a web page have a combination of different positions, the boxes (and therefore, their content) can overlap with each other, making the content difficult to read or consume.

.blue-box {  
  background-color: blue;  
}  
   
.green-box {  
  background-color: green;  
  position: relative;  
  top: -170px;  
  left: 170px;  
}

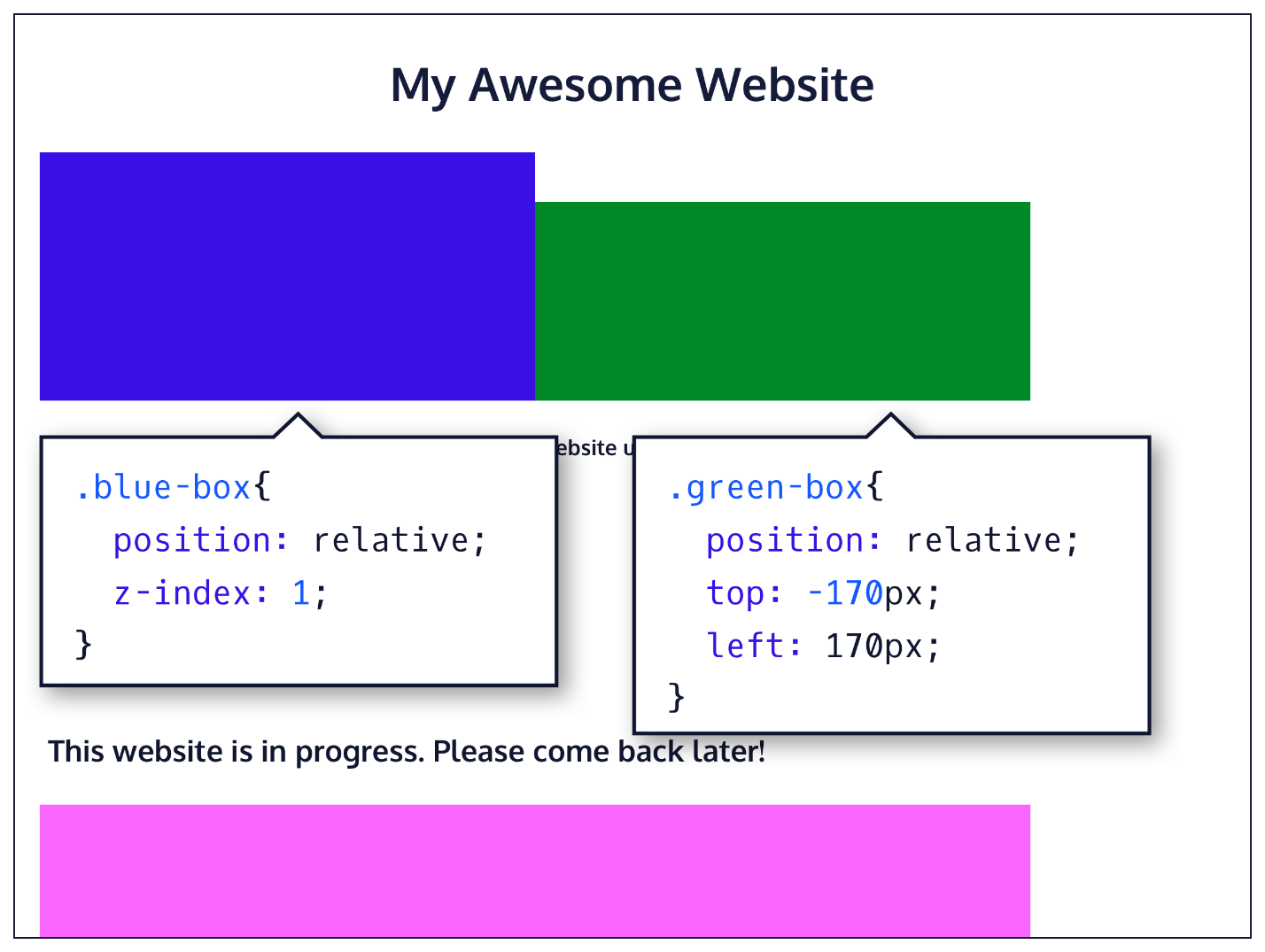
In the example above, the .green-box element overlaps on top of the .blue-box element.

The z-index property controls how far back or how far forward an element should appear on the web page when elements overlap. This can be thought of as the *depth* of elements, with deeper elements appearing behind shallower elements.

The z-index property accepts integer values. Depending on their values, the integers instruct the browser on the order in which elements should be layered on the web page.

.blue-box {  
  background-color: blue;  
  position: relative;  
  z-index: 1;  
}  
   
.green-box {  
  background-color: green;  
  position: relative;  
  top: -170px;  
  left: 170px;  
}

In the example above, we set the .blue-box position to relative and the z-index to 1. We changed position to relative, because the z-index property does *not* work on static elements. The z-index of 1 moves the .blue-box element forward, because the z-index value has not been explicitly specified for the .green-box element, which means it has a default z-index value of 0. Take a look the example image below:



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**INLINE DISPLAY**

Every HTML element has a default display value that dictates if it can share horizontal space with other elements. Some elements fill the entire browser from left to right regardless of the size of their content. Other elements only take up as much horizontal space as their content requires and can be directly next to other elements.

In this lesson, we’ll cover three values for the display property: inline, block, and inline-block.

The default display for some elements, such as <em>, <strong>, and <a>, is called inline. Inline elements have a box that wraps tightly around their content, only taking up the amount of space necessary to display their content and not requiring a new line after each element. The height and width of these elements cannot be specified in the CSS document. For example, the text of an anchor tag (<a>) will, by default, be displayed on the same line as the surrounding text, and it will only be as wide as necessary to contain its content. inline elements cannot be altered in size with the height or width CSS properties.

To learn more about <em>inline</em> elements, read <a href="#">MDN documentation</a>.

In the example above, the <em> element is inline, because it displays its content on the same line as the content surrounding it, including the anchor tag. This example will display:

To learn more about inline elements, read [MDN documentation](https://developer.mozilla.org/en-US/docs/Web/HTML/Inline_elements).

The CSS display property provides the ability to make any element an inline element. This includes elements that are not inline by default such as paragraphs, divs, and headings.

h1 {  
  display: inline;  
}

The CSS in the example above will change the display of all <h1> elements to inline. The browser will render <h1> elements on the same line as other inline elements immediately before or after them (if there are any).  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**DISPLAY: BLOCK**

Some elements are not displayed in the same line as the content around them. These are called block-level elements. These elements fill the entire width of the page by default, but their width property can also be set. Unless otherwise specified, they are the height necessary to accommodate their content.

Elements that are block-level by default include all levels of heading elements (<h1> through <h6>), <p>, <div> and <footer>. For a complete list of block level elements, visit [the MDN documentation](https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements).

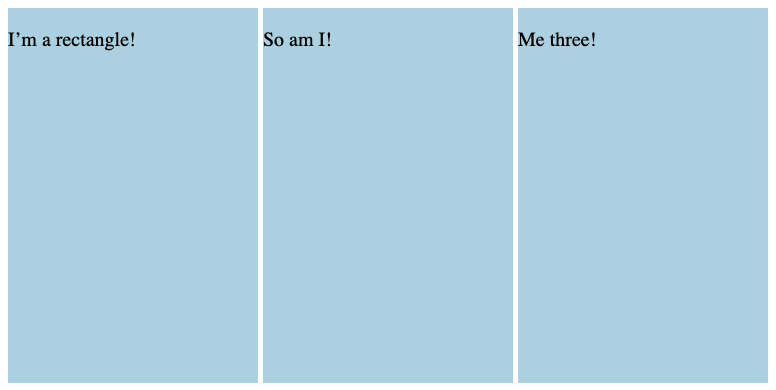
strong {  
  display: block;  
}

In the example above, all <strong> elements will be displayed on their own line, with no content directly on either side of them even though their contents may not fill the width of most computer screens.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**DISPLAY: INLINE-BLOCK**

The third value for the display property is inline-block. Inline-block display combines features of both inline and block elements. Inline-block elements can appear next to each other and we can specify their dimensions using the width and height properties. Images are the best example of default inline-block elements.

For example, the <div>s below will be displayed on the same line and with the specified dimensions:



Let’s take a look at the code:

<div class="rectangle">  
  <p>I’m a rectangle!</p>  
</div>  
<div class="rectangle">  
  <p>So am I!</p>  
</div>  
<div class="rectangle">  
  <p>Me three!</p>  
</div>

.rectangle {  
  display: inline-block;  
  width: 200px;  
  height: 300px;  
}

There are three rectangular divs that each contain a paragraph of text. The .rectangle <div>s will all appear inline (provided there is enough space from left to right) with a width of 200 pixels and height of 300 pixels, even though the text inside of them may not require 200 pixels by 300 pixels of space.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**CLEAR**

The float property can also be used to float multiple elements at once. However, when multiple floated elements have different heights, it can affect their layout on the page. Specifically, elements can “bump” into each other and not allow other elements to properly move to the left or right.

The clear property specifies how elements should behave when they bump into each other on the page. It can take on one of the following values:

* left—the left side of the element will not touch any other element within the same containing element.
* right—the right side of the element will not touch any other element within the same containing element.
* both—neither side of the element will touch any other element within the same containing element.
* none—the element can touch either side.

div {  
  width: 200px;  
  float: left;  
}  
   
div.special {  
  clear: left;  
}

In the example above, all <div>s on the page are floated to the left side. The element with class special did not move all the way to the left because a taller <div> blocked its positioning. By setting its clear property to left, the special <div> will be moved all the way to the left side of the page.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**REVIEW: LAYOUT**

Great job! In this lesson, you learned how to control the positioning of elements on a web page.

Let’s review what you’ve learned so far:

* The position property allows you to specify the position of an element.
* When set to relative, an element’s position is relative to its default position on the page.
* When set to absolute, an element’s position is relative to its closest positioned parent element. It can be pinned to any part of the web page, but the element will still move with the rest of the document when the page is scrolled.
* When set to fixed, an element’s position can be pinned to any part of the web page. The element will remain in view no matter what.
* When set to sticky, an element can stick to a defined offset position when the user scrolls its parent container.
* The z-index of an element specifies how far back or how far forward an element appears on the page when it overlaps other elements.
* The display property allows you to control how an element flows vertically and horizontally in a document.
* inline elements take up as little space as possible, and they cannot have manually adjusted width or height.
* block elements take up the width of their container and can have manually adjusted heights.
* inline-block elements can have set width and height, but they can also appear next to each other and do not take up their entire container width.
* The float property can move elements as far left or as far right as possible on a web page.
* You can clear an element’s left or right side (or both) using the clear property.

When combined with an understanding of the box model, positioning can create visually appealing web pages. So far, we’ve focused on adding content in the form of text to a web page. In the next unit, you’ll learn how to add and manipulate images to a web page.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

HTML

<!DOCTYPE html>

<html>

<head>

  <title>Please Participate in Our Survey!</title>

  <link href="https://fonts.googleapis.com/css?family=Oswald:300,700|Varela+Round" rel="stylesheet">

  <link rel="stylesheet" href="style.css">

</head>

<body>

  <header>

    <ul>

      <li>Question 1</li>

      <li>Question 2</li>

      <li>Question 3</li>

      <li>Question 4</li>

      <li>Question 5</li>

      <li>Question 6</li>

    </ul>

  </header>

  <div class="welcome">

    <h1><strong>Welcome</strong> to our survey!</h1>

    <p>We're looking forward to getting your answers so we can make sure our products and services are the best they can be!</p>

  </div>

  <div class="question">

    <h4>Question 1</h4>

    <h2>I like participating in physical activity such as running, swimming, or biking.</h2>

    <div class="answer">

      <h3>Disagree</h3>

    </div>

    <div class="answer">

      <h3>Neutral</h3>

    </div>

    <div class="answer">

      <h3>Agree</h3>

    </div>

  </div>

  <div class="question">

    <h4>Question 2</h4>

    <h2>I try to keep up to date with the latest fashion in active wear.</h2>

    <div class="answer">

      <h3>Disagree</h3>

    </div>

    <div class="answer">

      <h3>Neutral</h3>

    </div>

    <div class="answer">

      <h3>Agree</h3>

    </div>

  </div>

  <div class="question">

    <h4>Question 3</h4>

    <h2>I purchase clothing online regularly.</h2>

    <div class="answer">

      <h3>Disagree</h3>

    </div>

    <div class="answer">

      <h3>Neutral</h3>

    </div>

    <div class="answer">

      <h3>Agree</h3>

    </div>

  </div>

  <div class="question">

    <h4>Question 4</h4>

    <h2>I try to buy goods that are designed and/or manufactured in my home country.</h2>

    <div class="answer">

      <h3>Disagree</h3>

    </div>

    <div class="answer">

      <h3>Neutral</h3>

    </div>

    <div class="answer">

      <h3>Agree</h3>

    </div>

  </div>

  <div class="question">

    <h4>Question 5</h4>

    <h2>I look to famous athletes when trying to choose what to wear when training.</h2>

    <div class="answer">

      <h3>Disagree</h3>

    </div>

    <div class="answer">

      <h3>Neutral</h3>

    </div>

    <div class="answer">

      <h3>Agree</h3>

    </div>

  </div>

  <footer>

    <h3>Thanks for taking our survey!</h3>

  </footer>

</body>

</html>

CSS

body {

  background-color: #FFF;

  margin: 0 auto;

}

header {

  background-color: #466995;

  border-bottom: 1px solid #466995;

  position: fixed;

  width: 100%;

  z-index: 10;

}

ul {

  margin: 30px auto;

  padding: 0 20px;

  text-align: center;

}

li {

  color: #FFF;

  font-family: 'Oswald', sans-serif;

  font-size: 16px;

  font-weight: 300;

  text-transform: uppercase;

  display: inline-block;

  width: 80px;

}

li:hover {

  color: #DBE9EE;

}

h1 {

  color: #466995;

  font-family: 'Oswald', sans-serif;

  font-size: 32px;

  font-weight: 300;

  text-transform: uppercase;

}

h2 {

  color: #333;

  font-family: 'Varela Round', sans-serif;

  font-size: 26px;

  font-weight: 100;

  margin: 0 auto 20px auto;

}

h3 {

  color: #466995;

  font-family: 'Oswald', sans-serif;

  font-size: 18px;

  text-align: center;

  font-weight: 700;

  text-transform: uppercase;

  padding: 30px;

}

h4 {

  color: #466995;

  font-family: 'Oswald', sans-serif;

  font-size: 18px;

  font-weight: 300;

  letter-spacing: 2px;

  text-align: center;

  text-transform: uppercase

}

p {

  color: #333;

  font-family: 'Varela Round', sans-serif;

  font-size: 18px;

}

footer {

  background-color: #DBE9EE;

  text-align: center;

  height: 100px;

}

.welcome {

  background-color: #DBE9EE;

  box-sizing: border-box;

  padding: 40px;

  text-align: center;

  width: 100%;

  position: relative;

  top: 50px;

}

.question {

  text-align: center;

  position: relative;

  top: 40px;

}

.answer {

  border: 1px solid #466995;

  margin: 20px;

  display: inline-block;

}

.answer:hover {

  background: #C0D6DF;

  color: #FFF;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

HTML

<!DOCTYPE html>

<html>

  <head>

    <link href='https://fonts.googleapis.com/css?family=Raleway:400, 600' rel='stylesheet' type='text/css'>

    <link href='style.css' rel='stylesheet' type='text/css'/>

  </head>

  <body>

    <header>

        <nav>

          <ul>

            <li> About </li> <li> Work </li> <li> Team </li> <li> Contact </li>

          </ul>

        </nav>

    </header>

    <main>

      <div class="jumbotron">

        <div class="container">

          <h1>We are Broadway</h1>

          <a href="#" class="btn-main"> Get Started </a>

        </div>

      </div>

    </main>

    <section class="supporting">

      <div class="container">

        <div class="col">

          <img src="https://content.codecademy.com/projects/broadway/design.svg">

          <h2>Design</h2>

          <p>Make your projects look great and interact beautifully.</p>

          <a href="#"> Learn More</a><br>

        </div>

        <div class="col">

          <img src="https://content.codecademy.com/projects/broadway/develop.svg">

          <h2>Develop</h2>

          <p>Use modern tools to turn your design into a web site</p>

          <a href="#"> Learn More</a><br>

        </div>

        <div class="col test">

          <img src="https://content.codecademy.com/projects/broadway/deploy.svg">

          <h2>Deploy</h2>

          <p>Use modern tools to turn your design into a web site</p>

          <a href="#"> Learn More</a><br>

        </div>

      </div>

    </section>

    <footer>

      <div class="container">

        <p>&copy; Broadway 2017</p>

      </div>

    </footer>

  </body>

</html>

CSS

html, body {

  margin: 0;

  padding: 0;

}

header {

  background-color: #333333;

  position: fixed;

  width: 100%;

  z-index: 2;

}

nav {

  padding: 20px 0;

  text-align: center;

}

nav li {

  color: #fff;

  font-family: 'Raleway', sans-serif;

  font-weight: 600;

  font-size: 12px;

  display: inline-block;

  width: 80px;

}

main {

  text-align: center;

  position: relative;

  top: 80px;

}

main h1 {

  color: #333;

  font-family: 'Raleway', sans-serif;

  font-weight: 600;

  font-size: 70px;

  margin-top: 0px;

  padding-top: 80px;

  margin-bottom: 80px;

  text-transform: uppercase;

}

footer {

  background-color: #333;

  color: #fff;

  padding: 30px 0;

  position: fixed;

  width: 100%;

  bottom: 0px;

  z-index: 1;

  margin-top: 10px;

}

footer p {

  font-family: 'Raleway', sans-serif;

  text-transform: uppercase;

  font-size: 11px;

}

.container {

  max-width: 940px;

  margin: 0 auto;

  padding: 0 10px;

  text-align: center;

}

.jumbotron {

  height: 800px;

  background-image: url("https://content.codecademy.com/projects/broadway/bg.jpg");

  -webkit-background-size: cover;

  -moz-background-size: cover;

  -o-background-size: cover;

  background-size: cover;

}

.btn-main {

  background-color: #333;

  color: #fff;

  font-family: 'Raleway', sans-serif;

  font-weight: 600;

  font-size: 18px;

  letter-spacing: 1.3px;

  padding: 16px 40px;

  text-decoration: none;

  text-transform: uppercase;

}

.btn-default {

  font-family: 'Raleway', sans-serif;

  font-weight: 600;

  font-size: 10px;

  letter-spacing: 1.3px;

  padding: 10px 20px;

  text-decoration: none;

  text-transform: uppercase;

  margin-bottom: 20px;

}

.supporting {

  padding-top: 80px;

  padding-bottom: 100px;

}

.supporting .col {

  font-family: 'Raleway', sans-serif;

  text-align: center;

  display: inline-block;

  height: 200px;

  width: 200px;

  margin-top: 50px;

  margin-bottom: 50px;

}

.supporting img {

  height: 32px;

}

.supporting h2 {

  font-weight: 600;

  font-size: 23px;

  text-transform: uppercase;

}

.supporting p {

  font-weight: 400;

  font-size: 14px;

  line-height: 20px;

  padding: 0 20px;

  margin-bottom: 20px;

}

.supporting a {

  background-color: white;

  color: #333333;

  font-family: 'Raleway', sans-serif;

  font-weight: 600;

  font-size: 12px;

  letter-spacing: 1.3px;

  text-decoration: none;

  text-transform: uppercase;

  padding: 10px;

  margin-bottom: 10px;

  border: 2px solid #333333;

}

@media (max-width: 500px) {

  main h1 {

    font-size: 50px;

    padding: 0 40px;

  }

  .supporting .col {

    width: 100%;

  }

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**cd**  
Use cd .. to move up in the directory structure if you need to navigate back to **blog**.

For example, to move from **/home/ccuser/workspace/blog/2015/jan/memory** to **/home/ccuser/workspace/blog**, you need to move up 3 levels in the directory structure.

$ cd ../../..

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**pwd**

The next command we’re going to look at is pwd, which stands for “print working directory.” It outputs the name of the directory you are currently in, called the *working directory*.

Here the working directory is **blog/**. In Codecademy courses, your working directory is usually inside the **/home/ccuser/workspace/** directory.

Together with ls, the pwd command is useful to show where you are in the filesystem.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**ls**

The first command we’re going to look at is ls. A *command* is a directive to the computer to perform a specific task. When you type ls, the command line looks at the directory you are in, and then “lists” all the files and directories inside of it. Be sure to type the letter l as in “List” and not the number 1.

In the terminal, the first thing you see is $. This is called a *shell prompt*. It appears when the terminal is ready to accept a command.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**MKDIR**  
We’ve made our new **media** directory, but what if we wanted to create another directory inside that one? We could cd into **media** and then use mkdir, or we could make the new directory from our current position by using a / to combine arguments as we did in the last lesson.

Use:

$ mkdir media/tv

to create a new directory named **tv** inside **media**.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**touch**

Now we know how to create directories through the command line, but how do we create new files?

We can do this using the command touch:

$ touch keyboard.txt

The touch command creates a new file inside the working directory. It takes in a filename as an argument and then creates an empty file with that name in the current working directory.

Here we used touch to create a new file named **keyboard.txt**.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**Review**

Congratulations! You’ve learned five commands commonly used to navigate the filesystem from the command line. What can we generalize so far?

* The *command line* is a text interface for the computer’s operating system. To access the command line, we use the terminal.
* A *filesystem* organizes a computer’s files and directories into a tree structure. It starts with the *root directory*. Each parent directory can contain more child directories and files.
* From the command line, you can navigate through files and folders on your computer:
  + pwd outputs the name of the current working directory.
  + ls lists all files and directories in the working directory.
  + cd switches you into the directory you specify.
  + mkdir creates a new directory in the working directory.
  + touch creates a new file inside the working directory.
* You can use helper commands to make navigation easier:
  + clear clears the terminal
  + tab autocompletes the name of a file or directory
  + ↑ and ↓ allow you to cycle through previous commands

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**OPACITY AND ALPHA**

All of the colors we’ve seen so far have been opaque, or non-transparent. When we overlap two opaque elements, nothing from the bottom element shows through the top element. In this exercise, we’ll change the *opacity*, or the amount of transparency, of some colors so that some or all of the bottom elements are visible through a covering element.

To use opacity in the HSL color scheme, use hsla instead of hsl, and four values instead of three. For example:

color: hsla(34, 100%, 50%, 0.1);

The first three values work the same as hsl. The fourth value (which we have not seen before) is the *alpha*. This last value is sometimes called opacity.

Alpha is a decimal number from zero to one. If alpha is zero, the color will be completely transparent. If alpha is one, the color will be opaque. The value for half-transparent would be 0.5.

You can think of the alpha value as, “the amount of the background to mix with the foreground”. When a color’s alpha is below one, any color behind it will be blended in. The blending happens for each pixel; no blurring occurs.

The RGB color scheme has a similar syntax for opacity, rgba. Again, the first three values work the same as rgb and the last value is the alpha. Here’s an example:

color: rgba(234, 45, 98, 0.33);

A little unconventional, but still worth mentioning is how hex colors can also have an alpha value. By adding a two-digit hexadecimal value to the end of the six-digit representation (#52BC8280), or a one-digit hexadecimal value to the end of the three-digit representation (#F003), you can change the opacity of a hexadecimal color. Hex opacity ranges from 00 (transparent) to FF (opaque).

Alpha can only be used with HSL, RGB, and hex colors; we cannot add the alpha value to name colors like green.

There is, however, a named color keyword for zero opacity, transparent. It’s equivalent to rgba(0, 0, 0, 0), and it’s used like any other color keyword:

color: transparent;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**REVIEW**

We’ve completed our extensive tour of the colors in CSS! Let’s review the key information we’ve learned.

There are four ways to represent color in CSS:

* Named colors—there are more than 140 named colors, which you can review [here](https://developer.mozilla.org/en-US/docs/Web/CSS/color_value" \t "_blank).
* Hexadecimal or hex colors
  + Hexadecimal is a number system that has sixteen digits, 0 to 9 followed by “A” to “F”.
  + Hex values always begin with # and specify values of red, blue, and green using hexadecimal numbers such as #23F41A.
  + Six-digit hex values with duplicate values for each RGB value can be shorted to three digits.
* RGB
  + RGB colors use the rgb() syntax with one value for red, one value for blue and one value for green.
  + RGB values range from 0 to 255 and look like this: rgb(7, 210, 50).
* HSL
  + HSL stands for hue (the color itself), saturation (the intensity of the color), and lightness (how light or dark a color is).
  + Hue ranges from 0 to 360 and saturation and lightness are both represented as percentages like this: hsl(200, 20%, 50%).
* You can add opacity to color in RGB and HSL by adding a fourth value, a, which is represented as a percentage.

Great job! Feel empowered to add a bit of color to each of your projects!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

HTML

<!DOCTYPE html>

<html>

<head>

  <title>House Store</title>

  <link rel="stylesheet" type="text/css" href="resources/css/reset.css">

  <link rel="stylesheet" type="text/css" href="resources/css/style.css">

  <link href="https://fonts.googleapis.com/css?family=Creepster" rel="stylesheet">

  <link href="https://fonts.googleapis.com/css?family=Open+Sans:400,700" rel="stylesheet">

</head>

<body>

  <!-- Header -->

  <header>

    <div class="image-container logo-small">

      <img src="https://content.codecademy.com/courses/freelance-1/unit-6/logo-sm.png">

    </div>

    <div class="image-container logo-big">

      <img src="https://content.codecademy.com/courses/freelance-1/unit-6/logo-lg.png">

    </div>

    <nav>

      <ul>

        <li><a href="#">Tools</a></li>

        <li><a href="#">Lumber</a></li>

        <li><a href="#">Paint</a></li>

        <li><a href="#">Garden</a></li>

        <li><a href="#">Contact</a></li>

      </ul>

    </nav>

  </header>

  <!-- Banner -->

  <div id="banner">

    <h1>PAINTING IS SCARY</h1>

    <p>But it doesn't have to be! Let our handy Virtual Paint Department help guide you through the process of choosing a color. Know the hue? You know what to do. Click the order button and we'll get the paint to you.</p>

  </div>

  <!-- Color Guide -->

  <div id="color-guide">

    <div class="introduction">

      <h2>Color Guide</h2>

      <p>Here at HouseStore, we take color seriously. In each of the following sections, well explore our base colors in swatches that incrementally change three values: <strong>Hue</strong>, <strong>Saturation</strong> and <strong>Lightness</strong>. You'll be able to pick from a wide variety of colors that all work well with each other because they <strong>stem from the same value.</strong></p>

    </div>

    <!-- Red Swatches -->

    <div class="color reds">

      <div class="information">

        <h3>Reds</h3>

        <p>Firetrucks, lipstick, fresh berries &mdash; red is a color with power, emotion, intensity. For this reason we recommend using reds as accent colors. Red is particularly suited to kitchens, as it is said to evoke hunger!</p><p>Our base red is <span class="base-color">HSL (350, 100, 50)</span></p>

        <div class="image-container">

          <img src="  https://content.codecademy.com/courses/freelance-1/unit-6/reds.png">

        </div>

      </div>

      <div class="swatches">

        <h4>Lightness</h4>

        <div class="swatch lightness">

          <div class="color-1"></div>

          <div class="color-2"></div>

          <div class="color-3"></div>

          <div class="color-4"></div>

          <div class="color-5"></div>

        </div>

        <h4>Saturation</h4>

        <div class="swatch saturation">

          <div class="color-1"></div>

          <div class="color-2"></div>

          <div class="color-3"></div>

          <div class="color-4"></div>

          <div class="color-5"></div>

        </div>

        <h4>Hue</h4>

        <div class="swatch hue">

          <div class="color-1"></div>

          <div class="color-2"></div>

          <div class="color-3"></div>

          <div class="color-4"></div>

          <div class="color-5"></div>

        </div>

      </div>

    </div>

    <!-- Green Swatches -->

    <div class="color greens">

      <div class="information">

        <h3>Greens</h3>

        <p>They say that geniuses choose green &mdash; we think any of these verdant colors will look smart! Deck your halls like a lush outdoor space, adorn your rooms with emerald, or just make your houseguests green with envy!</p>

        <p>Our base green is <span class="base-color">HSL (130, 100, 50)</span></p>

        <div class="image-container">

          <img src="  https://content.codecademy.com/courses/freelance-1/unit-6/greens.png">

        </div>

      </div>

      <div class="swatches">

        <h4>Lightness</h4>

        <div class="swatch lightness">

          <div class="color-1"></div>

          <div class="color-2"></div>

          <div class="color-3"></div>

          <div class="color-4"></div>

          <div class="color-5"></div>

        </div>

        <h4>Saturation</h4>

        <div class="swatch saturation">

          <div class="color-1"></div>

          <div class="color-2"></div>

          <div class="color-3"></div>

          <div class="color-4"></div>

          <div class="color-5"></div>

        </div>

        <h4>Hue</h4>

        <div class="swatch hue">

          <div class="color-1"></div>

          <div class="color-2"></div>

          <div class="color-3"></div>

          <div class="color-4"></div>

          <div class="color-5"></div>

        </div>

      </div>

    </div>

    <!-- Blue Swatches -->

    <div class="color blues">

      <div class="information">

        <h3>Blues</h3>

        <p>Blues dont give us the blues &mdash; in fact, blue is one of the most popular colors in home interiors, and for good reason! Energizing like a summer sky, calming like a lakeshore, light as mist or deep as indigo, blue can do it all.</p>

        <p>Our base blue is <span class="base-color">HSL (220, 100, 50)</span></p>

        <div class="image-container">

          <img src="  https://content.codecademy.com/courses/freelance-1/unit-6/blues.png">

        </div>

      </div>

      <div class="swatches">

        <h4>Lightness</h4>

        <div class="swatch lightness">

          <div class="color-1"></div>

          <div class="color-2"></div>

          <div class="color-3"></div>

          <div class="color-4"></div>

          <div class="color-5"></div>

        </div>

        <h4>Saturation</h4>

        <div class="swatch saturation">

          <div class="color-1"></div>

          <div class="color-2"></div>

          <div class="color-3"></div>

          <div class="color-4"></div>

          <div class="color-5"></div>

        </div>

        <h4>Hue</h4>

        <div class="swatch hue">

          <div class="color-1"></div>

          <div class="color-2"></div>

          <div class="color-3"></div>

          <div class="color-4"></div>

          <div class="color-5"></div>

        </div>

      </div>

    </div>

  </div>

  <!-- Footer -->

  <footer>

    <strong>All set? Click below to order.</strong>

    <p>Most colors can be delivered to your door within 48 hours. We'll reach out if your color needs some extra attention, which could delay shipment.</p>

    <a href="#" class="button">Order My Paint</a>

  </footer>

</body>

</html>

CSS

/\* Universal Styles \*/

html {

  font-size: 16px;

}

body {

  font-family: "Open Sans", sans-serif;

}

a {

  text-decoration: none;

  color: inherit;

}

strong {

  font-weight: bold;

}

.image-container {

  overflow: hidden;

}

.image-container img {

  display: block;

  max-width: 100%;

}

@media only screen and (max-width: 990px) {

  html {

    font-size: 14px;

  }

}

/\* Header \*/

header {

  display: flex;

  align-items: center;

  padding: .5rem 3.75rem;

  background-color: #ff8000;

}

header .logo-small {

  display: none;

}

header .logo-small,

header .logo-big {

  flex-grow: 1;

}

nav li {

  display: inline;

  padding-right: 2rem;

}

nav li:last-child {

  padding-right: 0;

}

@media only screen and (max-width: 990px) {

  header .logo-big {

    display: none;

  }

  header .logo-small {

    display: block;

  }

}

@media only screen and (max-width: 540px) {

  header {

    flex-direction: column;

    padding-left: 0;

    padding-right: 0;

  }

  header .logo-small {

    margin-bottom: 1rem;

  }

}

/\* Banner \*/

#banner {

  position: relative;

  height: 43.75rem;

  padding: 0 25%;

  background: url(" https://content.codecademy.com/courses/freelance-1/unit-6/banner.png") center center no-repeat;

  background-size: cover;

  text-align: center;

}

#banner:before { /\* Orange Overlay \*/

  position: absolute;

  content: "";

  top: 0;

  left: 0;

  width: 100%;

  height: 100%;

  background-color:rgba(255,128,0,0.75);

}

#banner \* {

  position: relative; /\* Makes elements display above overlay. \*/

}

h1 {

  padding-top: 10.4375rem;

  padding-bottom: 1.25rem;

  font-family: "Creepster", cursive;

  font-size: 8rem;

  color: rgba(0,0,0,0.7);

}

#banner p {

  color: white;

  line-height: 1.5;

  font-size: 1.375rem;

}

@media only screen and (max-width: 820px) {

  h1 {

    padding-top: 7rem;

    font-size: 6rem;

  }

}

@media only screen and (max-width: 590px) {

  h1 {

    font-size: 4rem;

  }

}

/\* Color Guide \*/

#color-guide {

  padding: 3.875rem 15% 13.5rem 15%;

}

#color-guide .introduction {

  padding: 0 10%;

  margin-bottom: .75rem;

  text-align: center;

  font-size: 1.375rem;

  line-height: 1.4;

}

#color-guide h2 {

  margin-bottom: 2.4375rem;

  line-height: 1;

  font-size: 2rem;

  color: #ff8000;

}

#color-guide .color {

  display: flex;

  justify-content: space-between;

  padding-top: 5.25rem;

}

.color .information {

  display: flex;

  flex-direction: column;

  justify-content: space-between;

  width: 35%;

}

.color .information h3 {

  padding-bottom: .5rem;

  font-size: 1.375rem;

  font-weight: bold;

}

.color .information p {

  font-size: .875rem;

  line-height: 1.4;

}

.color .swatches {

  width: 60%;

}

.color .swatches h4 {

  margin-bottom: 1.25rem;

  font-size: 1.125rem;

  font-weight: bold;

  color:#9b9b9b;

}

.color .swatch {

  display: flex;

  height: 6.6875rem;

  border: 10px solid #e6e6e6;

  margin-bottom: 1.25rem;

}

.color .swatch:last-child {

  margin-bottom: 0;

}

.color .swatch > div {

  flex-grow: 1;

  border-right: 10px solid #e6e6e6;

}

.color .swatch > div:last-child {

  border-right: 0;

}

@media only screen and (max-width: 820px) {

  #color-guide .color {

    flex-direction: column;

    align-items: center;

  }

  .color .information {

    width: 100%;

    margin-bottom: 3rem;

    align-items: center;

  }

  .color .information h3 {

    padding-bottom: 1.5rem;

  }

  .color .information p {

    margin-bottom: 2rem;

  }

  .color .swatches {

    width: 100%;

  }

}

/\* Red Swatches \*/

.reds .base-color {

  color: hsl(350, 100%, 50%);

}

/\* Red lightness decreases by 15 \*/

.reds .lightness .color-1 {

  background-color:hsl(350, 100%, 80%);

}

.reds .lightness .color-2 {

  background-color: hsl(350, 100%, 65%);

}

.reds .lightness .color-3 {

  background-color: hsl(350, 100%, 50%);

}

.reds .lightness .color-4 {

  background-color: hsl(350, 100%, 35%);

}

.reds .lightness .color-5 {

  background-color: hsl(350, 100%, 20%);

}

/\* Red saturation decreases by 15 \*/

.reds .saturation .color-1 {

  background-color:hsl(350, 100%, 80%);

}

.reds .saturation .color-2 {

  background-color: hsl(350, 85%, 50%);

}

.reds .saturation .color-3 {

  background-color: hsl(350, 70%, 50%);

}

.reds .saturation .color-4 {

  background-color: hsl(350, 55%, 50%);

}

.reds .saturation .color-5 {

  background-color: hsl(350, 40%, 50%);

}

/\* Red hue increases by 15 \*/

.reds .hue .color-1 {

  background-color:hsl(350, 100%, 80%);

}

.reds .hue .color-2 {

  background-color: hsl(335, 100%, 50%);

}

.reds .hue .color-3 {

  background-color: hsl(350, 100%, 50%);

}

.reds .hue .color-4 {

  background-color: hsl(5, 100%, 50%);

}

.reds .hue .color-5 {

  background-color: hsl(20, 100%, 50%);

}

/\* Green Swatches \*/

.greens .base-color {

  color: #00ff2a;

}

/\* Green lightness decreases by 20 \*/

.greens .lightness .color-1 {

  background-color:hsl(350, 100%, 80%);

}

.greens .lightness .color-2 {

  background-color: hsl(103, 100%, 70%);

}

.greens .lightness .color-3 {

  background-color: hsl(103, 100%, 50%);

}

.greens .lightness .color-4 {

  background-color: hsl(103, 100%, 30%);

}

.greens .lightness .color-5 {

  background-color: hsl(104, 100%, 10%);

}

/\* Green saturation decreases by 20 \*/

.greens .saturation .color-1 {

  background-color:hsl(350, 100%, 80%);

}

.greens .saturation .color-2 {

  background-color: hsl(130, 80%, 50%);

}

.greens .saturation .color-3 {

  background-color: hsl(130, 60%, 50%);

}

.greens .saturation .color-4 {

  background-color: hsl(130, 40%, 50%);

}

.greens .saturation .color-5 {

  background-color: hsl(131, 20%, 50%);

}

/\* Green hue increases by 22 \*/

.greens .hue .color-1 {

  background-color:hsl(350, 100%, 80%);

}

.greens .hue .color-2 {

  background-color: hsl(108, 100%, 50%);

}

.greens .hue .color-3 {

  background-color: hsl(130, 100%, 50%);

}

.greens .hue .color-4 {

  background-color: hsl(152, 100%, 50%);

}

.greens .hue .color-5 {

  background-color: hsl(174, 100%, 50%);

}

/\* Blue Swatches \*/

.blues .base-color {

  color: #0055ff;

}

/\* Blue lightness decreases by 20 \*/

.blues .lightness .color-1 {

 background-color:hsl(350, 100%, 80%);

}

.blues .lightness .color-2 {

  background-color: hsl(220, 100%, 70%);

}

.blues .lightness .color-3 {

  background-color: hsl(220, 100%, 50%);

}

.blues .lightness .color-4 {

  background-color: hsl(220, 100%, 30%);

}

.blues .lightness .color-5 {

  background-color: hsl(220, 100%, 10%);

}

/\* Blue saturation decreases by 20 \*/

.blues .saturation .color-1 {

  background-color:hsl(350, 100%, 80%);

}

.blues .saturation .color-2 {

  background-color: hsl(220, 80%, 50%);

}

.blues .saturation .color-3 {

  background-color: hsl(220, 60%, 50%);

}

.blues .saturation .color-4 {

  background-color: hsl(220, 40%, 50%);

}

.blues .saturation .color-5 {

  background-color: hsl(220, 20%, 50%);

}

/\* Blue hue decreases by 20 \*/

.blues .hue .color-1 {

  background-color:hsl(350, 100%, 80%);

}

.blues .hue .color-2 {

  background-color: hsl(240, 100%, 50%);

}

.blues .hue .color-3 {

  background-color: hsl(220, 100%, 50%);

}

.blues .hue .color-4 {

  background-color: hsl(200, 100%, 50%);

}

.blues .hue .color-5 {

  background-color: hsl(180, 100%, 50%);

}

/\* Footer \*/

footer {

  position: relative;

  height: 30rem;

  padding: 7.8125rem 30% 0 30%;

  background: url("https://content.codecademy.com/courses/freelance-1/unit-6/footer.png") center center no-repeat;

  background-size: cover;

  text-align: center;

  font-size: 1.125rem;

  line-height: 1.4;

  color: white;

}

footer:before { /\* Overlay \*/

  position: absolute;

  content: "";

  top: 0;

  left: 0;

  width: 100%;

  height: 100%;

  background-color:rgba(255, 128, 0, 0.75);

}

footer \* {

  position: relative; /\* Makes elements display above overlay. \*/

}

footer strong {

  display: block;

  margin-bottom: 1.25rem;

  font-size: 2.25rem;

}

footer p {

  margin-bottom: 4.3125rem;

}

footer .button {

  padding: 1.25rem 3.75rem;

  border-radius: 4px;

  background-color: #ff8000;

}

@media only screen and (max-width: 560px) {

  footer {

    padding: 4.8125rem 15% 0 15%;

  }

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**FONT FAMILY**

You may remember from the [Visual Rules](https://www.codecademy.com/content-items/1368d1ea90382cbe44b60eeac19e9573/exercises/font-family" \t "_blank) lesson that the font of an element can be changed using the font-family property.

h1 {  
  font-family: Arial;  
}

In the example above, the font family for all <h1> heading elements have been set to Arial.

Let’s talk about some things to keep in mind when setting font-family values.

##### Multi-Word Values

When specifying a typeface with multiple words, like Times New Roman, it is recommended to use quotation marks (' ') to group the words together, like so:

h1 {  
  font-family: 'Times New Roman';  
}

##### Web Safe Fonts

There is a selection of fonts that will appear the same across all browsers and operating systems. These fonts are referred to as web safe fonts. You can check out a complete list of web safe fonts [here](https://www.cssfontstack.com/" \t "_blank).

##### Fallback Fonts and Font Stacks

Web safe fonts are good fallback fonts that can be used if your preferred font is not available.

h1 {  
  font-family: Caslon, Georgia, 'Times New Roman';  
}

In the example above, Georgia and Times New Roman are fallback fonts to Caslon. When you specify a group of fonts, you have what is known as a font stack. A font stack usually contains a list of similar-looking fonts. Here, the browser will first try to use the Caslon font. If that’s not available, it will try to use a similar font, Georgia. And if Georgia is not available, it will try to use Times New Roman.

##### Serif and Sans-Serif

You may be wondering what features make a font similar to another font. The fonts Caslon, Georgia, and Times New Roman are Serif fonts. Serif fonts have extra details on the ends of each letter, as opposed to Sans-Serif fonts, which do not have the extra details.

serif and sans-serif are also keyword values that can be added as a final fallback font if nothing else in the font stack is available.

h1 {  
  font-family: Caslon, Georgia, 'Times New Roman', serif;  
}

In this final example, the font stack has 4 fonts. If the first 3 fonts aren’t available, the browser will use whatever serif font is available on the system.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**TEXT LAYOUT**

You’ve learned how text can be defined by font family, weight, style, and transformations. Now you’ll learn about some ways text can be displayed or laid out within the element’s container.

##### **Letter Spacing**

The letter-spacing property is used to set the horizontal spacing between the individual characters in an element. It’s not common to set the spacing between letters, but it can sometimes help the readability of certain fonts or styles. The letter-spacing property takes length values in units, such as 2px or 0.5em.

p {  
  letter-spacing: 2px;  
}

In the example above, each character in the paragraph element will be separated by 2 pixels.

##### **Word Spacing**

You can set the space between words with the word-spacing property. It’s also not common to increase the spacing between words, but it may help enhance the readability of bolded or enlarged text. The word-spacing property also takes length values in units, such as 3px or 0.2em.

h1 {  
  word-spacing: 0.3em;  
}

In the example above, the word spacing is set to 0.3em. For word spacing, using em values are recommended because the spacing can be set based on the size of the font.

##### **Line Height**

We can use the line-height property to set how tall we want each line containing our text to be. Line height values can be a unitless number, such as 1.2, or a length value, such as 12px, 5% or 2em.

p {  
  line-height: 1.4;  
}

In the example above, the height between lines is set to 1.4. Generally, the unitless value is preferred since it is responsive based on the current font size. In other words, if the line-height is specified by a unitless number, changing the font size will automatically readjust the line height.

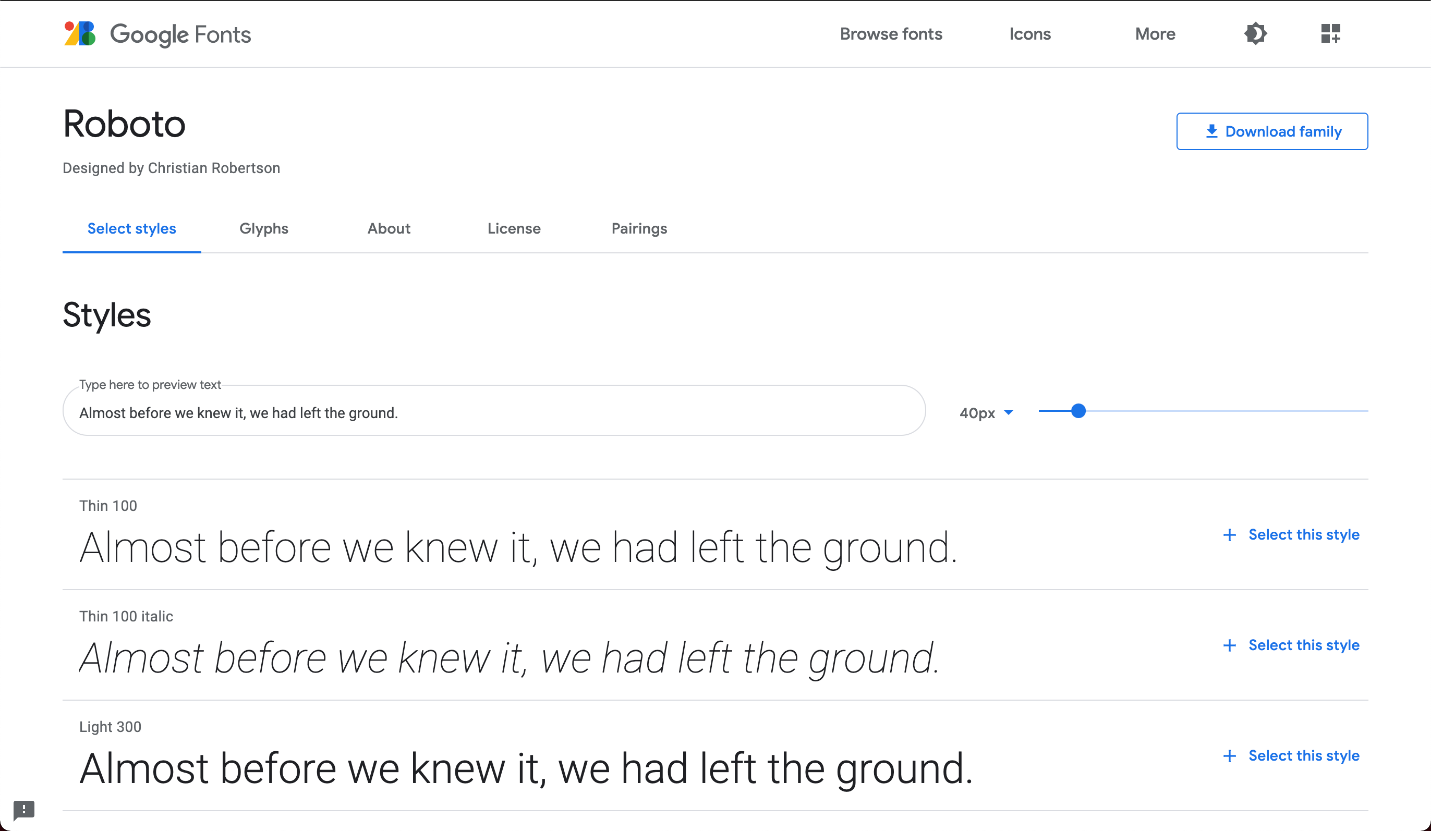
##### **Text Alignment**

The text-align property, which you may already be familiar with from the [CSS Visual Rules lesson](https://www.codecademy.com/courses/learn-css/lessons/css-visual-rules/exercises/text-align" \t "_blank), aligns text to its parent element.

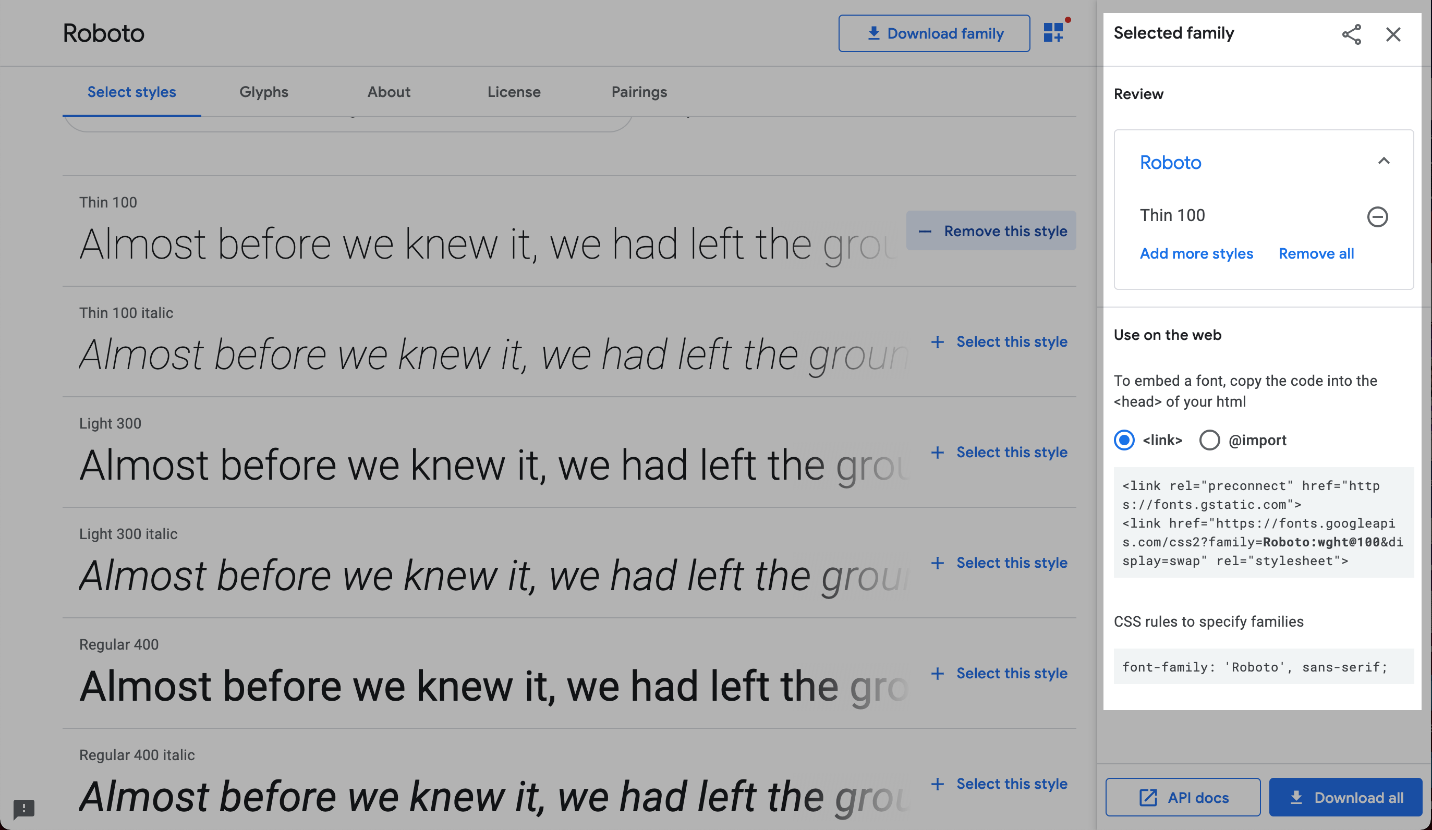
h1 {  
  text-align: right;  
}

In the example above, the <h1> element is aligned to the right side, instead of the default left.  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**WEB FONTS USING <LINK>**

Online font services, like [Google Fonts](https://fonts.google.com/" \t "_blank), make it easy to find and link to fonts from your site. You can browse and select fonts that match the style of your website.



When you select a font in Google Fonts, you’ll be shown all of the different styles available for that particular font. You can then select the styles you want to use on your site.



When you’re done selecting a font and its styles, you can review your selected font family, and a <link> element will be automatically generated for you to use on your site!

<head>  
  <!-- Add the link element for Google Fonts along with other metadata -->  
  <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@100&display=swap" rel="stylesheet">  
</head>

The generated <link> element needs to be added to the <head> element in your HTML document for it to be ready to be used in your CSS.

p {  
  font-family: 'Roboto', sans-serif;  
}

You can then create font-family declarations in your CSS, just like how you learned to do with other fonts!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

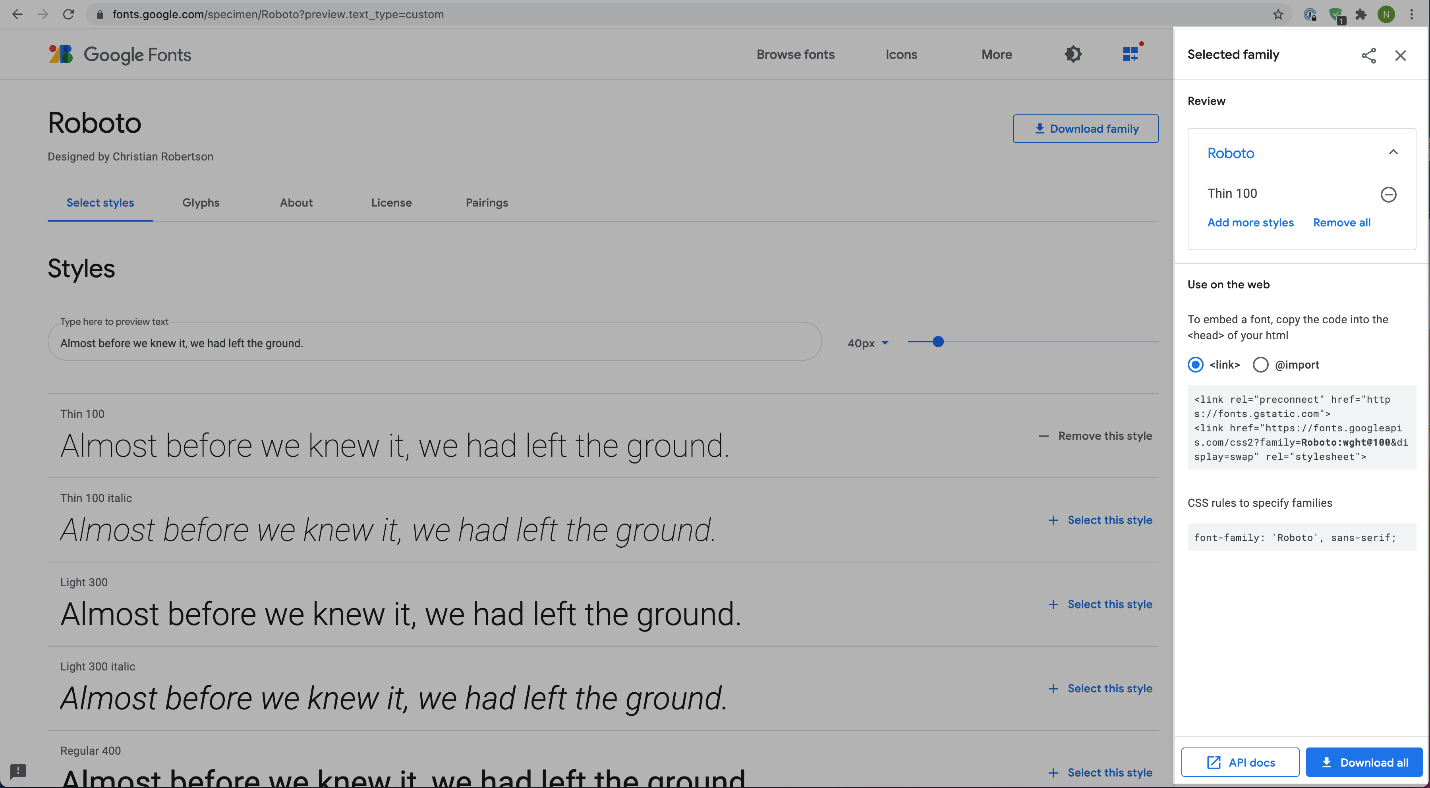
**WEB FONTS USING @FONT-FACE**

Fonts can also be added using a [@font-face ruleset](https://developer.mozilla.org/en-US/docs/Web/CSS/@font-face" \t "_blank) in your CSS stylesheet instead of using a <link> element in your HTML document. As mentioned earlier, fonts can be downloaded just like any other file on the web. They come in a few different file formats, such as:

* OTF (OpenType Font)
* TTF (TrueType Font)
* WOFF (Web Open Font Format)
* WOFF2 (Web Open Font Format 2)

The different formats are a progression of standards for how fonts will work with different browsers, with WOFF2 being the most progressive. It’s a good idea to include TTF, WOFF, and WOFF2 formats with your @font-face rule to ensure compatibility on all browsers.

Let’s take a look at how to use @font-face using the same Roboto font as before:



Within the “Selected Families” section, you can use the “Download” button to download the font files to your computer. The files will be downloaded as a single format, in this case, TTF. You can use a tool such as [Google Web Fonts Helper](https://google-webfonts-helper.herokuapp.com/fonts" \t "_blank) to generate additional file types for WOFF and WOFF2.

When you have the files you need, move them to a folder inside your website’s working directory, and you’re ready to use them in a @font-face ruleset!

@font-face {  
  font-family: 'MyParagraphFont';  
  src: url('fonts/Roboto.woff2') format('woff2'),  
       url('fonts/Roboto.woff') format('woff'),  
       url('fonts/Roboto.ttf') format('truetype');  
}

Let’s take a look at the example above, line by line:

* The @font-face at-rule is used as the selector. It’s recommended to define the @font-face ruleset at the top of your CSS stylesheet.
* Inside the declaration block, the font-family property is used to set a custom name for the downloaded font. The name can be anything you choose, but it must be surrounded by quotation marks. In the example, the font is named 'MyParagraphFont', as this font will be used for all paragraphs.
* The src property contains three values, each specifying the relative path to the font file and its format. In this example, the font files are stored inside a folder named fonts within the working directory.
* Note that the ordering for the different formats is important because our browser will start from the top of the list and search until it finds a font format that it supports. Read more on format prioritization on [CSS-Tricks](https://css-tricks.com/snippets/css/using-font-face-in-css/).

Once the @font-face at-rule is defined, you can use the font in your stylesheet!

p {  
  font-family: 'MyParagraphFont', sans-serif;  
}

Like using any other fonts, you can use the font-family property to set the font on any HTML element. The downloaded font can be referenced with the name you provided as the font-family property’s value in the @font-face ruleset—in this case, 'MyParagraphFont'.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

HTML

<!DOCTYPE html>

<html>

<head>

  <title>Morocco</title>

  <link rel="stylesheet" type="text/css" href="styles/reset.css">

  <link rel="stylesheet" type="text/css" href="styles/style.css">

  <link href="https://fonts.googleapis.com/css2?family=Abril+Fatface&display=swap" rel="stylesheet">

  <link href="https://fonts.googleapis.com/css2?family=Work+Sans&display=swap" rel="stylesheet">

  <link href="https://fonts.googleapis.com/css2?family=Merriweather&display=swap" rel="stylesheet">

</head>

<body>

  <!-- Header -->

  <nav class="header">

    <span class="logo">AOFIE CONLEAVY</span>

    <ul>

      <li><a href="#">TRAVELS</a></li>

      <li><a href="#">FICTION</a></li>

      <li><a href="#">CONTACT</a></li>

    </ul>

  </nav>

  <!-- Banner -->

  <div class="banner">

    <h2>DEC 20XX</h2>

    <h1>Morocco</h1>

  </div>

  <!-- Journal -->

  <div class="journal">

    <div class="first photo">

      <div class="image-container">

        <img src="https://content.codecademy.com/courses/freelance-1/unit-6/project-morocco/photo1.png">

      </div>

      <span class="caption">A convoy of camels criss-crossing the crests of the Sahara</span>

    </div>

    <p>

      <span class="first-letter">I</span> am in the Great Sahara Desert for the third sundown in a row. Fouad and I pass back and forth a sun-bleached map of the stars. It’s more of a tug-of-war to be honest. With what remains of the daylight, we commit to memory the positions of spots on the page. We hope to find some correlation in its negative, which slides overhead as the sun rapidly dies.

    </p>

    <p>

      "Navigating by night is always easier." Fouad flicks these English words as ash from a cigarette. "You’ll see."

    </p>

    <p>

      "We won't see anything. That's my point!"

    </p>

    <p>

      How did we get here?

    </p>

    <p>

      It started eight days ago when I arrived in Malilla on the boat from Malaga. The sun hit me like a judgement as I stepped onto the gangplank. A bit about Morocco:

    </p>

    <div class="photo">

      <div class="image-container">

        <img src="https://content.codecademy.com/courses/freelance-1/unit-6/project-morocco/photo2.png">

      </div>

      <span class="caption">The long and winding road to Merrakec</span>

    </div>

    <p>

      Morocco has a population of over 33.8 million and an area of 446,550 km2 (172,410 sq mi). Its capital is Rabat, and the largest city is Casablanca. Other major cities include Marrakesh, Tangier, Tetouan, Sale, Fes, Agadir, Meknes, Oujda, Kenitra, and Nador. A historically prominent regional power, Morocco has a history of independence not shared by its neighbours.

    </p>

    <span class="quote">"Navigating by night is always easier."</span>

    <p>

      Since the foundation of the first Moroccan state by Idris I in 789, the country has been ruled by a series of independent dynasties, reaching its zenith under the Almoravid and Almohad dynasty, spanning parts of Iberia and Northwestern Africa. Marinid and Saadi dynasties continued the struggle against foreign domination, and Morocco remained the only North-African country to avoid Ottoman occupation.

    </p>

    <p>

      The Alaouite dynasty, the current ruling dynasty, seized power in 1666. In 1912 Morocco was be divided into a French and Spanish protectorates, with an international zone in Tangier, and regained its independence in 1956. Moroccan culture is a blend of Arab, indigenous Berber, Sub-Saharan African, and European influences.

    </p>

    <div class="photo">

      <div class="image-container">

        <img src="https://content.codecademy.com/courses/freelance-1/unit-6/project-morocco/photo3.png">

      </div>

      <span class="caption">A stall at the Jemaa El Fnaa street market</span>

    </div>

    <p>

      Morocco claims the non-self-governing territory of Western Sahara as its Southern Provinces. Morocco annexed the territory in 1975, leading to a guerrilla war with indigenous forces until a cease-fire in 1991. Peace processes have thus far failed to break the political deadlock.

    </p>

    <p>

      Morocco is a constitutional monarchy with an elected parliament. The King of Morocco holds vast executive and legislative powers, especially over the military, foreign policy and religious affairs. Executive power is exercised by the government, while legislative power is vested in both the government and the two chambers of parliament, the Assembly of Representatives and the Assembly of Councillors. The king can issue decrees called dahirs which have the force of law. He can also dissolve the parliament after consulting the Prime Minister and the president of the Constitutional court.

    </p>

    <span class="quote">“Navigating by night is always easier.”</span>

    <p>

      Morocco's predominant religion is Islam, and the official languages are Arabic and Tamazight. Moroccan dialect, referred to as Darija, and French are also widely spoken. Morocco is an influential member of the Arab League and a part of the Union for the Mediterranean. It has the fifth largest economy of Africa.

    </p>

    <div class="photo">

      <div class="image-container">

        <img src="https://content.codecademy.com/courses/freelance-1/unit-6/project-morocco/photo4.png">

      </div>

      <span class="caption">The desert at night is strange and beautiful</span>

    </div>

  </div>

  <!-- Footer -->

  <footer>

    <div class="image-container">

      <img src="https://content.codecademy.com/courses/freelance-1/unit-6/project-morocco/author.png">

    </div>

    <div class="content">

      <p>

        <span class="author">Aoife Donleavy</span> has been writing on her travels for over two decades.

        After graduating from the <em>Idaho Writers' Workshop</em>, she piloted a biplane on a two-year voyage from <em>Anchorage</em>, <em>Alaska</em> to <em>Isafjorour</em>, <em>Iceland</em> stopping along the way for adventures throughout Europe.

      </p>

      <p>

        Since then, she has camped on all seven continents, and has been recognized worldwide for her spare, ageless prose. Aoife's new novel, <em>Tide Blade</em>, is currently available from <em>Walrus Publishing</em>.

      </p>

    </div>

  </footer>

</body>

</html>

CSS

html {

  font-size: 18px;

}

@media only screen and (max-width: 1000px) {

  html {

    font-size: 16px;

  }

}

@media only screen and (max-width: 680px) {

  html {

    font-size: 14px;

  }

}

/\* Header \*/

.header {

  display: flex;

  justify-content: space-around;

  align-items: center;

  height: 4.44rem;

  padding: 0 12%;

  background-color: white;

  box-shadow: 0 2px 6px 0 rgba(0, 0, 0, 0.05);

  font-family: Verdana, sans-serif;

  font-size: .77rem;

  font-weight: bold;

}

.header .logo {

  flex-grow: 1;

  color: #ffb78c;

}

.header li {

  display: inline;

  padding-right: 2.22rem;

}

.header li a {

  text-decoration: none;

  color: #4a4a4a;

}

@media only screen and (max-width: 550px) {

  .header {

    flex-direction: column;

  }

  .header .logo {

    flex-grow: 0;

  }

}

/\* Banner \*/

.banner {

  display: flex;

  flex-direction: column;

  justify-content: center;

  align-items: center;

  height: 50rem;

  background: url("https://content.codecademy.com/courses/freelance-1/unit-6/project-morocco/banner.jpg") center center no-repeat;

  background-size: cover;

  color: #ffb78c;

}

.banner h2 {

  padding: .55rem 0;

  border-top: 4px solid #ffb78c;

  border-bottom: 4px solid #ffb78c;

  font-size: 1.44rem;

  letter-spacing: 2px;

  font-weight: 500;

  font-family: "Work Sans", Arial, sans-serif;

}

.banner h1 {

  padding-top: 1.11rem;

  font-size: 11rem;

  font-weight: 900;

  font-family: "Abril Fatface", sans-serif;

}

@media only screen and (max-width: 750px) {

  .banner {

    height: 40rem;

  }

  .banner h1 {

    font-size: 8rem;

  }

}

@media only screen and (max-width: 530px) {

  .banner {

    height: 30rem;

  }

  .banner h1 {

    font-size: 6rem;

  }

}

/\* Journal \*/

.journal {

  padding: 0 25% 4rem 25%;

  background-color: rgb(254, 231, 218);

  color: #4a4a4a;

  font-family: "Work Sans", serif;

}

.photo {

  width: 75%;

  padding: 1.11rem;

  border-radius: 5px;

  margin: 0 auto 4.44rem auto;

  background-color: white;

  font-family: "Merriweather", serif;

}

.photo .image-container {

  overflow: hidden;

  margin-bottom: 1.11rem;

}

.photo .image-container img {

  width: 100%;

}

.photo .caption {

  font-style: italic;

}

.photo.first {

  position: relative;

  top: -2.77rem;

  margin-bottom: 1.67rem;

}

.journal p {

  padding-bottom: 2.77rem;

  font-size: 1.5rem;

  line-height: 1.4;

}

.journal .first-letter {

  float: left;

  padding-right: 1.4rem;

  font-family: "Abril Fatface", serif;

  font-size: 7.44rem;

  color: #10b0d8;

  line-height: 0.87;

}

.quote {

  display: block;

  padding: 4.44rem 0;

  margin-bottom: 3.33rem;

  border-top: 4px solid black;

  border-bottom: 4px solid black;

  text-align: center;

  font-size: 3.55rem;

  font-weight: 800;

  line-height: 1.2;

}

@media only screen and (max-width: 680px) {

  .journal {

    padding: 0 10% 4rem 10%;

  }

}

/\* Footer \*/

footer {

  display: flex;

  align-items: center;

  padding: 0 1%;

  background-color: #212121;

  line-height: 1.5;

}

footer .image-container {

  width: 400px;

}

footer .content {

  flex-grow: 1;

  font-style: italic;

  color: #9b9b9b;

}

footer p {

  padding-bottom: 1.66rem;

}

footer p:last-child {

  padding-bottom: 0;

}

footer .author {

  color: #ffb78c;

}

footer em {

  color: #cfcfcf;

}

@media only screen and (max-width: 750px) {

  footer {

    flex-direction: column;

    padding: 0 10% 2rem 10%;

  }

  footer .image-container {

    height: 300px;

    margin-bottom: 2rem;

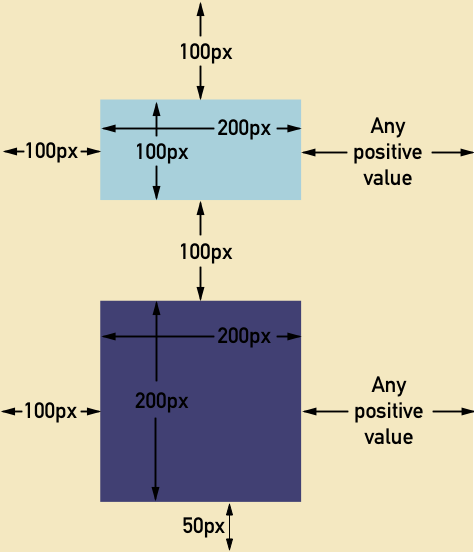
    overflow: hidden;

  }

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**Padding vs. Margin**

A padding vs. margin challenge! Take the initial teal and purple boxes that have a height and width of 100px and manipulate the padding and margin properties to get the image below:



HTML

<!DOCTYPE html>

<html>

<head>

  <title>Specificity</title>

  <link href="style.css" type="text/css" rel="stylesheet">

</head>

<body>

  <div id="container">

    <div id="teal" class="block">

      <h1> YOO</h1>

    </div>

    <div id="purple" class="block">

      <h2> YOO</h2>

    </div>

  </div>

</body>

</html>

CSS

\*{

  box-sizing: border-box;

}

.block {

  height: 100px;

  width: 100px;

  display: block;

}

#container{

  background-color: #F4E8C1;

  position: absolute;

  width: 100%;

}

#teal {

  background-color: #A8D0DB;

  width: 200px;

  margin-left: 100px;

  margin-bottom: 100px;

  margin-top: 100px;

  padding-right: 100px;

  overflow: hidden;

}

#purple {

  background-color: #414073;

  height: 200px;

  width: 200px;

  // padding: 50px;

  padding-right: 100px;

  padding-bottom: 100px;

  margin-left: 100px;

  margin-bottom: 50px;

}

h1{

  padding: 100px;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

HTML

CSS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

HTML

CSS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*