Introduction to CSS Training

with examples and
hands-on exercises

WEBUCATOR

Copyright © 2022 by Webucator. All rights reserved.

No part of this manual may be reproduced or used in any manner without written permission of the copyright owner.

Version: 5.2.9

The Authors

Jared Dunn

In addition to supporting students in self-paced, online courses and writing and tech editing courseware, Jared studies computer science and education at Brown University. He also is active in community theater as an actor, director, and producer.

Nat Dunn

Nat Dunn is the founder of Webucator (www.webucator.com), a company that has provided training for tens of thousands of students from thousands of organizations. Nat started the company in 2003 to combine his passion for technical training with his business expertise, and to help companies benefit from both. His previous experience was in sales, business and technical training, and management. Nat has an MBA from Harvard Business School and a BA in International Relations from Pomona College.

Follow Nat on Twitter at @natdunn and Webucator at @webucator.

Brian Hoke (Editor)

Brian Hoke joined Webucator as CEO in January, 2022. Brian Hoke was formerly Principal of Bentley Hoke, a web consultancy formed in Syracuse, New York, in 2000. The firm served the professional services, education, government, nonprofit, and retail sectors with a variety of development, design, and marketing services. Previously, Brian served as Director of Technology, Chair of the Computer and Information Science Department, and Dean of Students at Manlius Pebble Hill School, an independent day school in DeWitt, NY. Before that, Brian taught at Insitut auf dem Rosenberg, an international boarding school in St. Gallen, Switzerland. Brian holds degrees from Hamilton and Dartmouth colleges.

Class Files

Download the class files used in this manual at https://static.webucator.com/media/public/materials/classfiles/CSS101-5.2.9.zip.

Errata

Corrections to errors in the manual can be found at https://www.webucator.com/books/errata/.

Table of Contents

LESSON 1. Crash Course in CSS....1
- Benefits of Cascading Style Sheets....2
- CSS Rules....2
- Selectors....3
- Combinators....7
- Precedence of Selectors....9
- How Browsers Style Pages....10
- CSS Resets....13
- CSS Normalizers....15
- External Stylesheets, Embedded Stylesheets, and Inline Styles....16
- **Exercise 1: Creating an External Stylesheet....21**
- **Exercise 2: Creating an Embedded Stylesheet....29**
- **Exercise 3: Adding Inline Styles....32**
- and 34
- **Exercise 4: Styling div and span....38**
- Media Types....40
- Units of Measurement....42
- Inheritance....47

LESSON 2. CSS Fonts....53
- font-family....53
- @font-face....57
- font-size....64
- font-style....69
- font-variant....69
- font-weight....70
- line-height....73
- font....78
- **Exercise 5: Styling Fonts....82**

LESSON 3. Color and Opacity....85
- About Color and Opacity....85
- Color and Opacity Values....85
- color....88
- opacity....91
- **Exercise 6: Adding Color and Opacity to Text....94**

LESSON 4. CSS Text....99
letter-spacing....99
text-align....101
text-decoration....103
text-indent....105
text-shadow....107
text-transform....109
white-space....111
word-break....114
word-spacing....116
Exercise 7: Text Properties....119
LESSON 5. Borders, Margins, and Padding....125
The CSS Basic Box Model....125
Padding....129
Margin....133
Border....136
box-sizing....153
box-shadow....157
Exercise 8: Borders, Margin, and Padding....161
LESSON 6. Backgrounds....167
background-color....167
background-image....170
background-repeat....172
overflow....177
background-attachment....181
background-position....185
Creating a “hint” Class....188
background-size....191
background-origin....193
background-clip....196
Modifying the “hint” Class....198
background....202
Exercise 9: Backgrounds....205
LESSON 7. Display and Visibility....211
display....211
visibility....214

LESSON 8. Pseudo-classes and Pseudo-elements....219
Pseudo-classes....219
Styling Links with Pseudo-classes....221
Styling Tables and Articles with Pseudo-classes....227
Pseudo-elements....237
Using Pseudo-elements....238
Exercise 10: Pseudo-classes and Pseudo-elements....245
LESSON 9. Styling Tables with CSS....249
A Review of HTML table Elements and Attributes....249
CSS Properties and Styling Tables....253
Exercise 11: Styling a Table Exercise....264
LESSON 10. Positioning....271
Normal Flow....271
position....274
z-index....279
float and clear....284
Exercise 12: Positioning....293
LESSON 11. Transforms and Transitions....301
transition....301
transform....308
Exercise 13: Transforms and Transitions....321
LESSON 12. Layouts....327
Introduction to Flexible Box Layout Module....327
Introduction to Grid Layout....335
Multi-column Layout....347
Exercise 14: Layouts....354
LESSON 13. CSS Lists as Hierarchical Navigation....359
Exercise 15: Basic Horizontal Navigation Bars....360
Exercise 16: Basic Vertical Navigation Bars....366
Dynamic Drop-down and Fly-out Navigation Bars....371
Exercise 17: Basic Drop-down Menu....372
Exercise 18: Basic Fly-out Menu....380
Exercise 19: CSS List Menus....388
LESSON 14. Media Queries....393
What Are Media Queries?....393
Font Awesome....394
Exercise 20: Breakpoints....398
Exercise 21: Targeting Print....406

LESSON 1
Crash Course in CSS

Topics Covered

☑ Benefits of Cascading Style Sheets.

☑ Redefining how elements are formatted.

☑ CSS selectors.

☑ CSS combinators.

☑ The CSS cascade.

☑ CSS resets and normalizers.

☑ External stylesheets, embedded stylesheets, and inline styles.

☑ The `div` and `span` elements.

Evaluation Copy

☑ Media types.

☑ The viewport `<meta>` tag.

☑ Units of measurement.

☑ How browsers style pages.

☑ Inheritance.

Introduction

Since HTML 4.0, most HTML formatting elements (e.g., `center` for centering content) and attributes (e.g., `bgcolor` for background color) have been deprecated, meaning that, although they may still be supported by browsers, the WHATWG (the maintainer of the HTML specification) recommends that they no longer be used. Web designers are to use CSS instead. In this lesson, you will get a high-level overview of CSS.

1.1. Benefits of Cascading Style Sheets

The major benefits of CSS are:

1. Cleaner Code
    - Easier to maintain.
    - Speedier download.
    - Better for search engine optimization.
2. Modular Code
    - Style rules can be applied to multiple pages.
    - Consistency of design.
    - Easier to maintain.
3. Design Power
    - Precise control of position, size, margins, etc.
4. Division of Labor
    - Developers develop / Designers design.
5. Better Accessibility[1]
    - No need to misuse tags (e.g., `<blockquote>` for formatting).
    - No need for invisible images for positioning.
    - Users' stylesheets override authors' styles.

Evaluation Copy

1.2. CSS Rules

CSS *rules* are statements that define the style of an element or group of elements. The syntax is as follows:

1. See `https://www.w3.org/TR/CSS-access`.

EVALUATION COPY: Not to be used in class.

```
selector {
  property: value;
  property: value;
  property: value;
}
```

Each *property: value* pair is a *declaration*. Multiple declarations are separated by semicolons. The *selector* defines which elements are affected by the rule. Take a look at the following rule:

```
p {
  color: darkgreen;
  font-family: Verdana;
  font-size: 1.5em;
}
```

This rule specifies that all paragraph text should be darkgreen and use a `1.5em` Verdana font. That's one and a half times the size that the font would normally be. You will learn about units of measurement soon.

❖ 1.2.1. CSS Comments

Evaluation Copy

Comments in CSS begin with "`/*`" and end with "`*/`". See the example below:

```
p {
  color: red; /* All paragraphs should be red */
}
```

1.3. Selectors

Selectors identify the element(s) affected by the CSS rule. There are several types of selectors:

1. Type
2. Class
3. ID
4. Attribute
5. Universal

In this section, we will give a high-level explanation of each of these types of selectors. It is a lot of information all at once. You should read through this section slowly and carefully. While it is important that you have an understanding of the different types of selectors, you do not need to commit the syntax for each type to memory. That will happen over time as you use them in practice.

❖ 1.3.1. Type Selectors

Type selectors specify elements by tag name and affect every instance of that element type. Looking again at the previous example:

```
p {
  color: darkgreen;
  font-family: Verdana;
  font-size: 1.5em;
}
```

Again, this rule specifies that the text of **every** `p` element should be darkgreen and use a `1.5em` Verdana font.

❖ 1.3.2. Class Selectors

Evaluation Copy

In HTML, all elements can take the `class` attribute, which is used to assign one or more class names to an element. The names given to classes are arbitrary, but should be descriptive of the purpose of the class. In CSS, class selectors begin with a dot. For example, the following rule specifies that any elements with the class "`warning`" should be bold and red:

```
.warning {
  color: red;
  font-weight: bold;
}
```

Following are a couple of examples of elements of the "`warning`" class:

```
<h1 class="warning">WARNING</h1>
<p class="warning">Don't go there!</p>
```

If the class selector is preceded by an element name, then that selector only applies to the specified type of element. To illustrate, the following two rules indicate that `h1` elements of the class "`warning`" will be underlined, while `p` elements of the class "`warning`" will not be:

EVALUATION COPY: Not to be used in class.

```
h1.warning {
  color: red;
  text-decoration: underline;
}

p.warning {
  color: red;
  font-weight: bold;
}
```

Because both rules indicate that the color should be red, this could be rewritten as follows:

```
.warning {
  color: red;
}

h1.warning {
  text-decoration: underline;
}

p.warning {
  font-weight: bold;
}
```

Evaluation Copy

Note that you can assign an element any number of classes simply by separating the class names with spaces like this:

```
<div class="class1 class2 class3">...
```

❖ 1.3.3. ID Selectors

As with the `class` attribute, in HTML, all elements can take the `id` attribute, which is used to uniquely identify an element on the page. In CSS, ID selectors begin with a number sign (#) and have arbitrary names. The following rule will give the element with the "main-text" `id` a margin of `1.2em` on the left and right:

```
#main-text {
  margin-left: 1.2em;
  margin-right: 1.2em;
}
```

```
<div id="main-text">
  <p>This is the main text of the page...</p>
</div>
```

❖ 1.3.4. Attribute Selectors

Attribute selectors specify elements that contain a specific attribute. They can also specify what the value of that attribute should or should not be.

The following selector affects all links with a `target` attribute:

```
a[target] {
  color: red;
}
```

Evaluation Copy

The `=` (equals) operator can be used to specify the attribute value. The following selector would only affect links whose `target` attribute is exactly "`_blank`":

```
a[target='_blank'] {
  color: red;
}
```

You can get much more specific about attribute values…

The `^=` (starts-with) operator can be used to specify the beginning text of the attribute value. The following selector will only affect links whose `href` attribute starts with "`mailto`":

```
a[href^='mailto'] {
  color: red;
}
```

The `$=` (ends-with) operator can be used to specify the ending text of the attribute value. The following selector will only affect links whose `class` attribute ends with "`link`":

EVALUATION COPY: Not to be used in class.

```
a[class$='link'] {
  color: red;
}
```

The `*=` (contains) operator can be used to specify text the attribute value must contain. The following selector will only affect links whose `class` attribute contains "`top`":

```
a[class*='top'] {
  color: red;
}
```

❖ 1.3.5. The Universal Selector

The universal selector is an asterisk (`*`). It matches every element:

```
* {
  color: red;
}
```

❖ 1.3.6. Grouping

Selectors can share the same declarations by separating them with commas. The following rule will underline all `em` elements, all elements of the class "`warning`" and the element with the `id` of "`important`":

```
em,
.warning,
#important {
  text-decoration: underline;
}
```

1.4. Combinators

Combinators allow for the selection of elements based on the relationships between selectors.

There are several types of combinators:

1. Descendant
2. Child
3. General sibling
4. Adjacent sibling

❖ 1.4.1. Descendant Combinators

Descendant combinators specify elements by ancestry. Each "generation" is separated by a space. For example, the following rule states that `strong` elements within `p` elements should have red text:

```
p strong {
  color: red;
}
```

With descendant selectors, generations can be skipped. In other words, the code above does not require that the `strong` element is a direct child of the `p` element.

Evaluation Copy

❖ 1.4.2. Child Combinators

Child combinators specify a direct parent-child relationship. They are indicated by placing a `>` sign between the two element names:

```
p > strong {
  color: red;
}
```

In this case only `strong` elements that are direct children of `p` elements are affected.

❖ 1.4.3. General Sibling Combinators

General sibling combinators specify a sibling relationship[2] between two elements where the second element specified comes after the first. They are indicated by placing a `~` sign between the two element names:

2. Sibling elements have the same parent element.

EVALUATION COPY: Not to be used in class.

```
em ~ strong {
  color: red;
}
```

In this case only `strong` elements that are siblings of and follow `em` elements are affected. For example, in the following code: both `strong` elements would be red:

```
<em>Hello!</em>
<strong>Hi there!</strong>
I'm not in a strong or em element.
<strong>Howdy!</strong>
```

❖ 1.4.4. Adjacent Sibling Combinators

Adjacent sibling combinators specify a sibling relationship between two elements where the second element specified comes *immediately* after the first. They are indicated by placing a `+` sign between the two element names:

Evaluation Copy

```
em + strong {
  color: red;
}
```

In this case only `strong` elements that are siblings of and immediately follow `em` elements are affected. In the HTML code just shown above, only the first `strong` element (with the text "Hi there!") would be red, because it immediately follows an `em` element.

1.5. Precedence of Selectors

In the event that rules conflict:

- The rule with the more specific selector takes precedence.
- If two selectors have the same specificity, the rule specified later in the document takes precedence.

!important

Adding `!important` to any CSS declaration will give that declaration the highest specificity, but using `!important` is bad practice, and should be avoided.[3]

Evaluation Copy

1.6. How Browsers Style Pages

Browsers have built-in styles to make web pages readable by default (i.e., without CSS). Without any styles, an HTML page would just be one big line of text. For example, look at the code below:

3. See `https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity#the_!important_exception`.

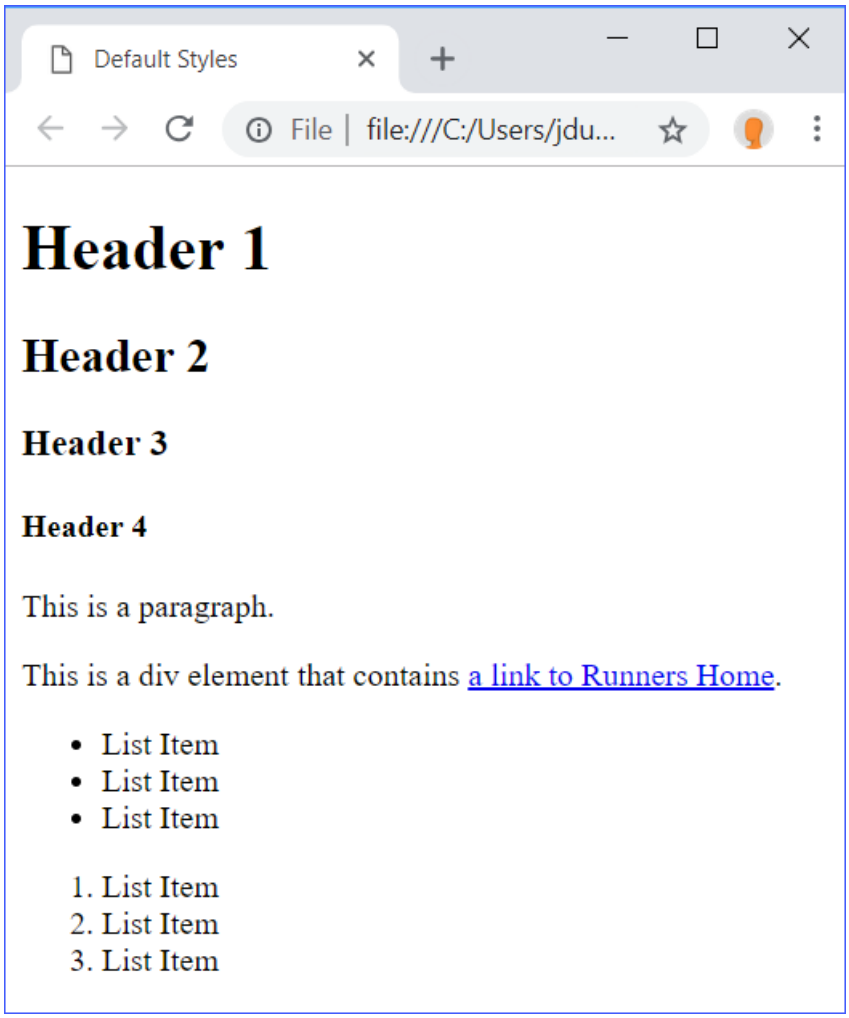
EVALUATION COPY: Not to be used in class.

Demo 1.1: CrashCourse/Demos/default-styles.html

```
1.   <!DOCTYPE html>
2.   <html lang="en">
3.   <head>
4.   <meta charset="UTF-8">
5.   <meta name="viewport" content="width=device-width,initial-scale=1">
6.   <title>Default Styles</title>
7.   </head>
8.   <body>
9.     <h1>Header 1</h1>
10.    <h2>Header 2</h2>
11.    <h3>Header 3</h3>
12.    <h4>Header 4</h4>
13.    <p>This is a paragraph.</p>
14.    <div>This is a div element that contains
15.      <a href="https://www.runners-home.com">a link to Runners Home</a>.
16.    </div>
17.    <ul>
18.      <li>List Item</li>
19.      <li>List Item</li>
20.      <li>List Item</li>
21.    </ul>
22.    <ol>
23.      <li>List Item</li>
24.      <li>List Item</li>
25.      <li>List Item</li>
26.    </ol>
27.  </body>
28.  </html>
```

Evaluation Copy

To make pages readable by default, the browsers add styles to render the page like this:

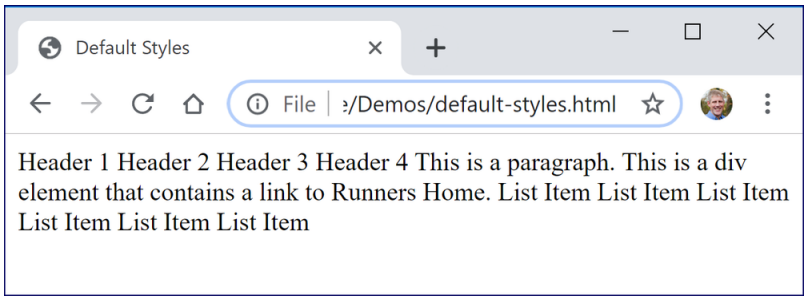


Notice the default styles:

1. The headings are bold, have varying font sizes, and have margin before and after them.
2. All of the elements, except the link, are on their own blocks as if they had `<br>` tags before and after them.
3. The unordered and ordered lists come with bullets and numbers, respectively.
4. The link is blue and underlined.

If browsers did not have default styles, they would display all the text in the body along a single line like this:

EVALUATION COPY: Not to be used in class.



As that's not very readable, it's a good thing that browsers do provide default styles. Unfortunately, not all browsers provide the same default styles for all elements, so it's common practice for web designers to remove or override the browsers' default styles with default styles of their own.

Evaluation Copy

1.7. CSS Resets

CSS resets are used to remove most default styles that browsers add to HTML elements, so that designers can decide on these styles themselves. For example, after applying a CSS reset, header tags will no longer be big and bold. By resetting the default styles, the designer gets rid of inconsistencies between browser default styles, opening the door for them to add styles that are rendered the same by all browsers.

A common CSS reset is the Meyer Reset[4], which you can see below:

4. https://meyerweb.com/eric/tools/css/reset/

Demo 1.2: CrashCourse/Demos/reset.css

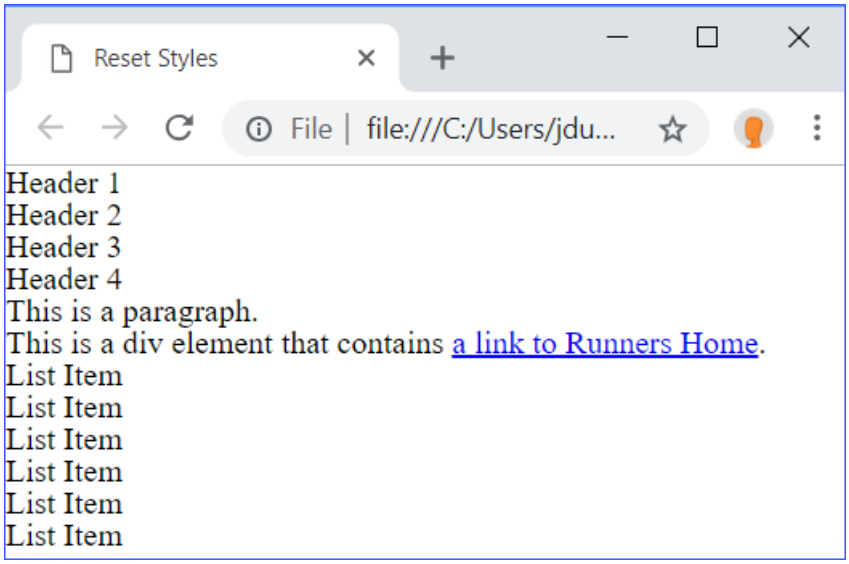
```
1.  /* http://meyerweb.com/eric/tools/css/reset/
2.     v2.0 | 20110126
3.     License: none (public domain)
4.  */
5.
6.  html, body, div, span, applet, object, iframe,
7.  h1, h2, h3, h4, h5, h6, p, blockquote, pre,
8.  a, abbr, acronym, address, big, cite, code,
9.  del, dfn, em, img, ins, kbd, q, s, samp,
10. small, strike, strong, sub, sup, tt, var,
11. b, u, i, center,
12. dl, dt, dd, ol, ul, li,
13. fieldset, form, label, legend,
14. table, caption, tbody, tfoot, thead, tr, th, td,
15. article, aside, canvas, details, embed,
16. figure, figcaption, footer, header, hgroup,
17. menu, nav, output, ruby, section, summary,
18. time, mark, audio, video {
19.   margin: 0;
20.   padding: 0;
21.   border: 0;
22.   font-size: 100%;
23.   font: inherit;
24.   vertical-align: baseline;
25. }
26. /* HTML5 display-role reset for older browsers */
27. article, aside, details, figcaption, figure,
28. footer, header, hgroup, menu, nav, section {
29.   display: block;
30. }
31. body {
32.   line-height: 1;
33. }
34. ol, ul {
35.   list-style: none;
36. }
37. blockquote, q {
38.   quotes: none;
39. }
40. blockquote:before, blockquote:after,
41. q:before, q:after {
42.   content: '';
43.   content: none;
44. }
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
45.  table {
46.    border-collapse: collapse;
47.    border-spacing: 0;
48.  }
```

Attaching this stylesheet to `default-styles.html` will render the page like this:



1.8. CSS Normalizers

CSS normalizers are slightly different from CSS resets. Instead of getting rid of all styles, they adjust styles so that they are consistent between browsers. In doing so, normalizers have to choose some styles over others. As such, normalizers are *opinionated*, meaning that the authors of these normalizers have made design choices based on their own preferences. For the rest of this course, we will be using a slightly modified version of a normalizer called normalize.css[5] in all of our files.

5. https://necolas.github.io/normalize.css/

Normalize

`normalize.css` is the normalizer used by Bootstrap[6], a popular open source toolkit for developing web pages, to improve cross-browser rendering.

1.9. External Stylesheets, Embedded Stylesheets, and Inline Styles

❖ 1.9.1. External Stylesheets

External stylesheets are created in separate documents with a `.css` extension. An external stylesheet is simply a listing of rules. It cannot contain HTML tags. Throughout this course, we will mainly be working with external stylesheets. `CrashCourse/Demos/styles.css` is an example of an external stylesheet.

Evaluation Copy

Demo 1.3: CrashCourse/Demos/styles.css

```
1.   .warning {
2.     color: red;
3.   }
4.
5.   h1.warning {
6.     text-decoration: underline;
7.   }
8.
9.   p.warning {
10.    font-weight: bold;
11.  }
```

The above CSS file can be included in any number of HTML pages using the `<link>` tag, which usually goes in the head of an HTML page:

6. https://getbootstrap.com

EVALUATION COPY: Not to be used in class.

```
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,initial-scale=1">
<link href="styles.css" rel="stylesheet">
<title>Page Title</title>
</head>
```

`<link>` Attributes

- `href` - points to the location of the external stylesheet.
- `rel` - must be set to "stylesheet" for linking stylesheets.

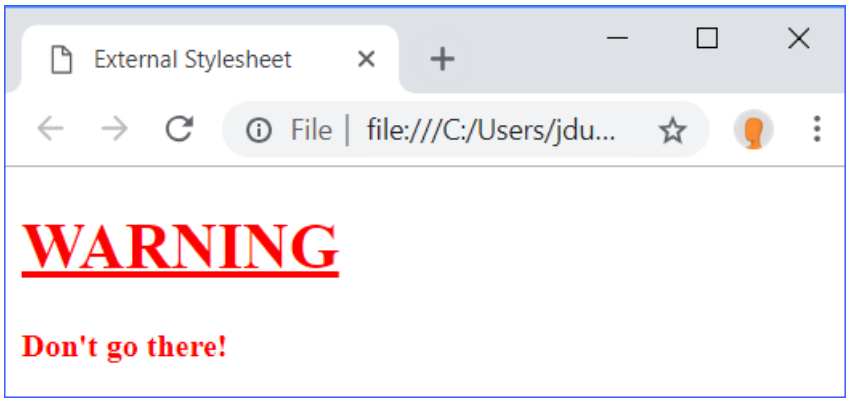
Notice the `<link>` tag in the code below that links to `styles.css`:

Evaluation Copy

Demo 1.4: CrashCourse/Demos/external-stylesheet.html

```
1.   <!DOCTYPE html>
2.   <html lang="en">
3.   <head>
4.   <meta charset="UTF-8">
5.   <meta name="viewport" content="width=device-width,initial-scale=1">
6.   <link href="../normalize.css" rel="stylesheet">
7.   <link href="styles.css" rel="stylesheet">
8.   <title>External Stylesheet</title>
9.   </head>
10.  <body class="webucator">
11.    <h1 class="warning">WARNING</h1>
12.    <p class="warning">Don't go there!</p>
13.  </body>
14.  </html>
```

This page will render as follows:



There is no limit to the number of external stylesheets a single HTML page can use. Notice in the preceding example that we linked to two external stylesheets:

Evaluation Copy

```
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,initial-scale=1">
<link href="../normalize.css" rel="stylesheet">
<link href="styles.css" rel="stylesheet">
<title>External Stylesheet</title>
</head>
```

Also, as you will see, external stylesheets can be combined with embedded stylesheets and inline styles.

❖ 1.9.2. Embedded Stylesheets

Embedded stylesheets appear in the `style` element, which usually goes in the head of an HTML page. The code below shows a page with an embedded stylesheet:

EVALUATION COPY: Not to be used in class.

Demo 1.5: CrashCourse/Demos/embedded-stylesheet.html

```
1.  <!DOCTYPE html>
2.  <html lang="en">
3.  <head>
4.  <meta charset="UTF-8">
5.  <meta name="viewport" content="width=device-width,initial-scale=1">
6.  <link href="../normalize.css" rel="stylesheet">
7.  <style>
8.    .warning {
9.      color: red;
10.   }
11.
12.   h1.warning {
13.     text-decoration: underline;
14.   }
15.
16.   p.warning {
17.     font-weight: bold;
18.   }
19. </style>
20. <title>Embedded Stylesheet</title>
21. </head>
22. <body class="webucator">
23.   <h1 class="warning">WARNING</h1>
24.   <p class="warning">Don't go there!</p>
25. </body>
26. </html>
```

Evaluation Copy

This page will render the same as the HTML page with the external stylesheet.

❖ 1.9.3. Inline Styles

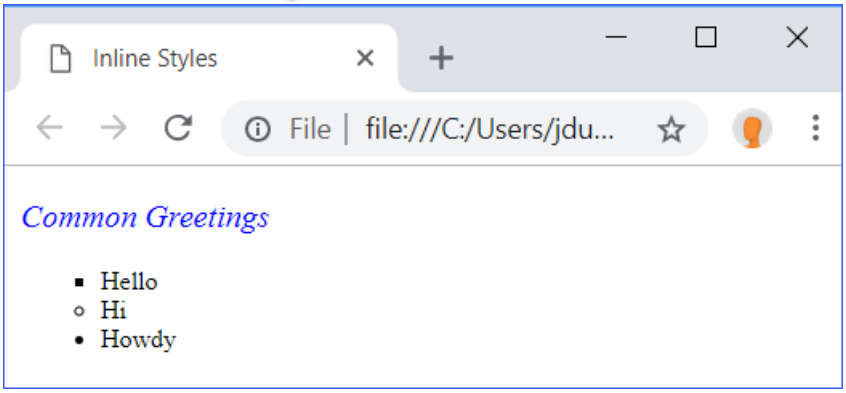
Inline styles are created by adding the `style` attribute to a tag. As with the `class` and `id` attributes, all elements can take the `style` attribute. The value of the `style` attribute is a list of one or more declarations separated by semicolons. The code sample below illustrates how inline styles are used:

Demo 1.6: CrashCourse/Demos/inline-styles.html

```
1.   <!DOCTYPE html>
2.   <html lang="en">
3.   <head>
4.   <meta charset="UTF-8">
5.   <meta name="viewport" content="width=device-width,initial-scale=1">
6.   <link href="../normalize.css" rel="stylesheet">
7.   <title>Inline Styles</title>
8.   </head>
9.   <body class="webucator">
10.    <p style="color: blue; font-style: italic; font-size: 1em;">
11.      Common Greetings
12.    </p>
13.    <ul style="font-size: 0.8em;">
14.      <li style="list-style-type: square;">Hello</li>
15.      <li style="list-style-type: circle;">Hi</li>
16.      <li style="list-style-type: disc;">Howdy</li>
17.    </ul>
18.  </body>
19.  </html>
```

Evaluation Copy

This page will render as follows:



Avoid Using Inline Styles

It is generally bad practice to use inline styles in production code; however, they can be useful for testing and debugging.

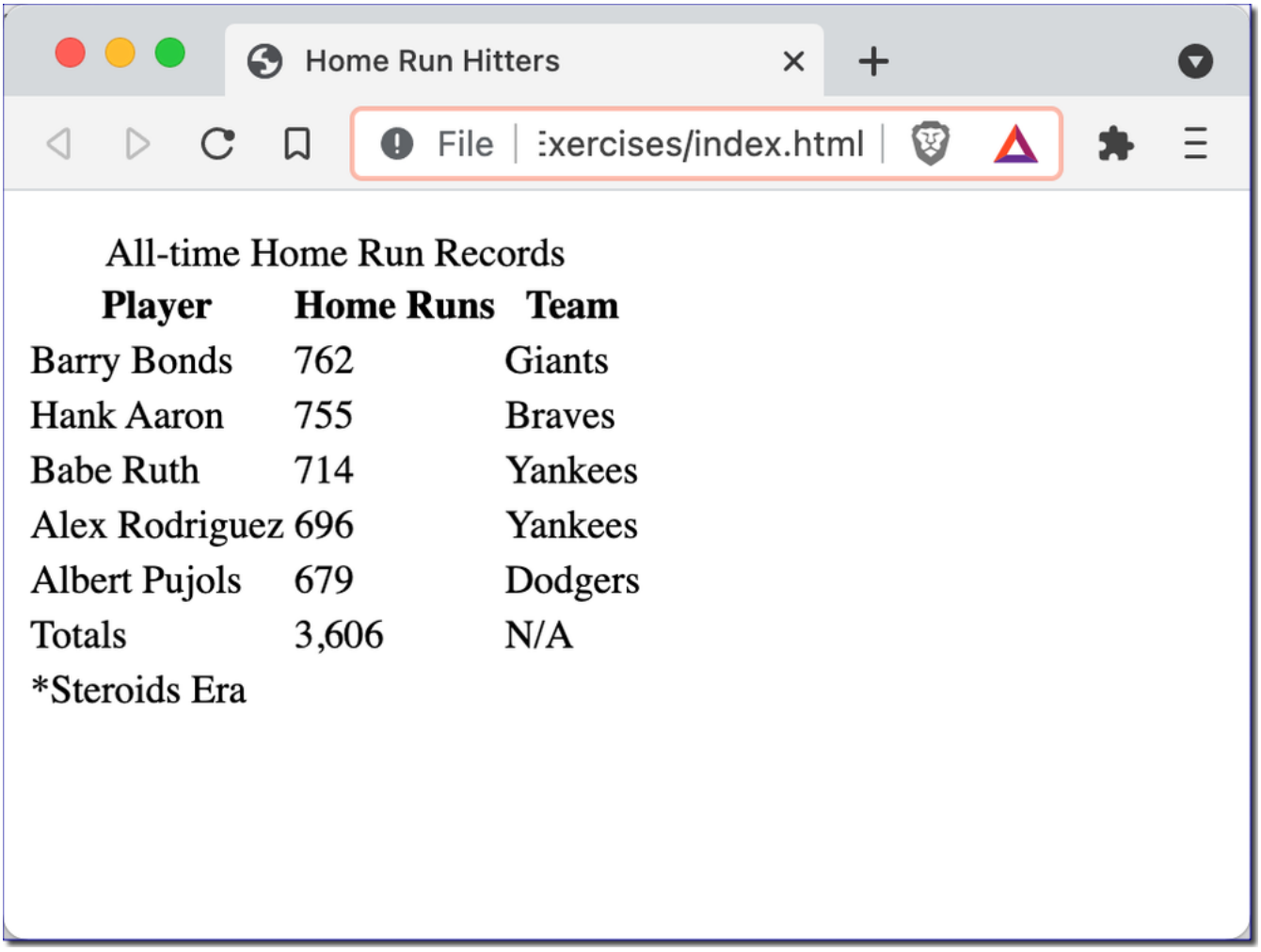
EVALUATION COPY: Not to be used in class.

Exercise 1: Creating an External Stylesheet

25 to 40 minutes

In this exercise, you will add several simple rules to an external stylesheet and link to it from `Crash Course/Exercises/index-external-styles.html`. Do not worry about learning the CSS properties and values at this point. This exercise is just to give you some practice creating a stylesheet.

1. Open `CrashCourse/Exercises/index.html` in a browser. It should look like this:



2. Open `CrashCourse/Exercises/index.html` in your editor and save it as `index-external-styles.html`.

3. Create a new file and save it as `styles.css` in the same directory.
4. Add a `<link>` tag with an `href` value of "`styles.css`" and a `rel` value of "stylesheet" in the head of `index-external-styles.html`.
5. In `styles.css`, add a rule for the `body` element that contains the following declarations:
    - `background-attachment: fixed;`
    - `background-image: url(images/baseball.png);`
    - `background-position: bottom;`
    - `background-repeat: repeat-x;`

    For a refresher on how to add rules, refer to the **CSS Rules** section of this lesson (see page 2).
6. Add a rule for `table` elements that contains the following declarations:
    - `background-color: white;`
    - `border-collapse: collapse;`
    - `margin: auto;`
    - `min-width: 800px;`
7. Add a rule for `caption` elements that contains the following declaration:
    - `font-style: italic;`
8. Add a rule for `thead` elements that contains the following declarations:
    - `background-color: darkblue;`
    - `color: white;`
9. Add a rule for `td` and `th` elements that contains the following declarations:
    - `border: 1px solid orange;`
    - `margin: 0;`
    - `padding: .3em;`

    For a refresher on adding a rule for multiple selectors, refer to **Grouping** within the **Selectors** section of this lesson (see page 7).
10. Add a rule for `td` elements that are in the `tfoot` that contains the following declarations:
    - `background-color: silver;`
    - `border-top: 2px solid black;`

Evaluation Copy

EVALUATION COPY: Not to be used in class.

- `font-weight: bold;`

For a refresher on adding a rule for an element that is contained within another element, refer to **Descendant Combinators** in the **Combinators** section of this lesson (see page 8).

11. Add a rule for elements with the "`steroids-era`" class that contains the following declaration:

    - `font-style: italic;`

    For a refresher on adding a rule for an element of a specific class, refer to the **Class Selectors** within the **Selectors** section of this lesson (see page 4).

12. Add a rule for the element with the "`key`" id that contains the following declarations:

    - `background-color: lightgray;`
    - `font-style: italic;`

    For a refresher on adding a rule for an element of a specific id, refer to the **ID Selectors** within the **Selectors** section of this lesson (see page 5).

13. Add the following two rules to the bottom of `styles.css`:

Evaluation Copy

```
tr td:nth-child(2) {
  text-align: center;
}

.steroids-era td:first-child::after {
  content: '*';
}
```

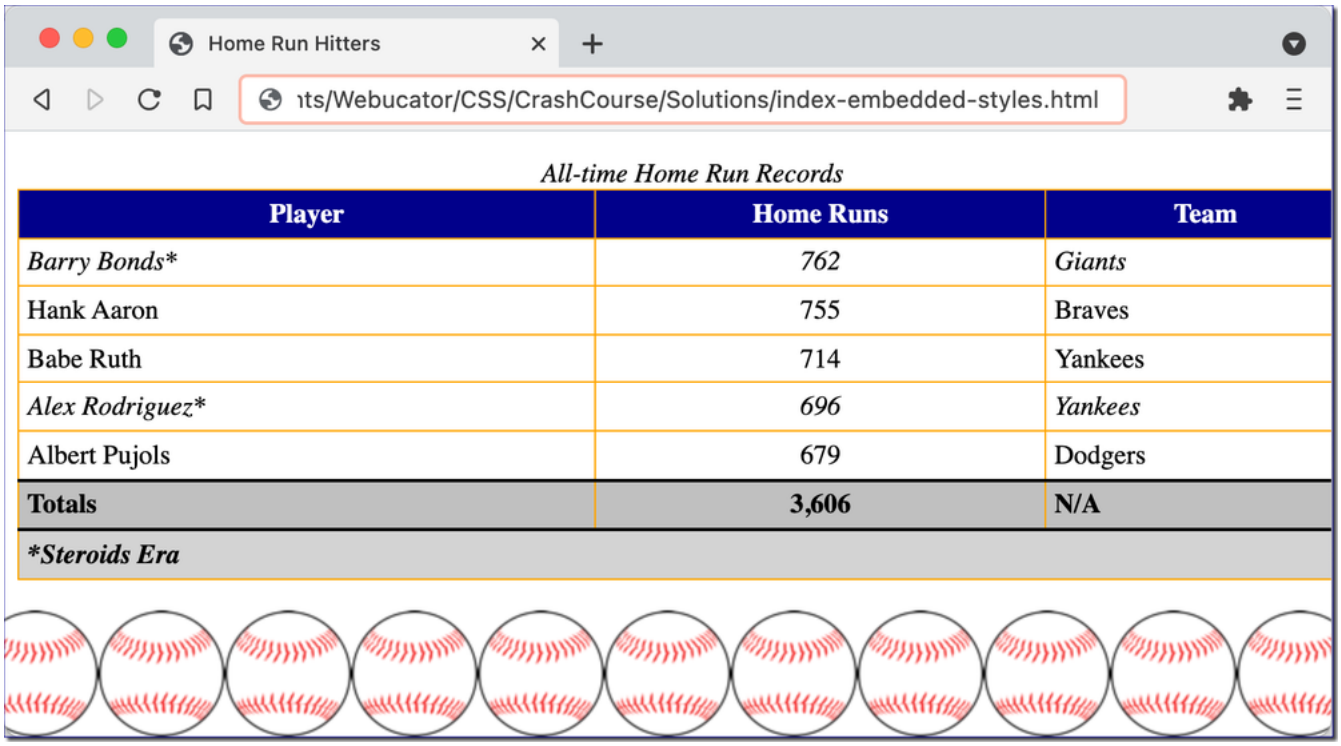
    Just add these last two rules exactly as written. Don't worry about understanding the selectors yet. We will cover them in later lessons. For now, we just want you to see what CSS can do.

14. In `index-external-styles.html`:

    A. Assign a `class` of "`steroids-era`" to the table rows containing Barry Bonds and Alex Rodriguez.

    B. Assign an `id` of "`key`" to the table data cell in the `tfoot` that contains the text "*Steroids Era."

15. Open `index-external-styles.html` in a browser. It should look like this:



All-time Home Run Records

Player	Home Runs	Team
*Barry Bonds**	*762*	*Giants*
Hank Aaron	755	Braves
Babe Ruth	714	Yankees
*Alex Rodriguez**	*696*	*Yankees*
Albert Pujols	679	Dodgers
Totals	**3,606**	**N/A**
***Steroids Era**		

EVALUATION COPY: Not to be used in class.

Solution: CrashCourse/Solutions/index-external-styles.html

```
      -------Lines 1 through 6 Omitted-------
7.    <link href="styles.css" rel="stylesheet">
8.    <title>Home Run Hitters</title>
9.    </head>
10.   <body class="webucator">
11.     <table>
12.       <caption>All-time Home Run Records</caption>
13.       <thead>
14.         <tr>
15.           <th>Player</th>
16.           <th>Home Runs</th>
17.           <th>Team</th>
18.         </tr>
19.       </thead>
20.       <tbody>
21.         <tr class="steroids-era">
22.           <td>Barry Bonds</td>
23.           <td>762</td>
24.           <td>Giants</td>
25.         </tr>
      -------Lines 26 through 35 Omitted-------
36.         <tr class="steroids-era">
37.           <td>Alex Rodriguez</td>
38.           <td>696</td>
39.           <td>Yankees</td>
40.         </tr>
      -------Lines 41 through 52 Omitted-------
53.         <tr>
54.           <td colspan="3" id="key">*Steroids Era</td>
55.         </tr>
56.       </tfoot>
57.     </table>
58.   </body>
59.   </html>
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

Solution: CrashCourse/Solutions/styles.css

```
1.  body {
2.    background-attachment: fixed;
3.    background-image: url(images/baseball.png);
4.    background-position: bottom;
5.    background-repeat: repeat-x;
6.  }
7.
8.  table {
9.    background-color: white;
10.   border-collapse: collapse;
11.   margin: auto;
12.   min-width: 800px;
13. }
14.
15. caption {
16.   font-style: italic;
17. }
18.
19. thead {
20.   background-color: darkblue;
21.   color: white;
22. }
23.
24. td,
25. th {
26.   border: 1px solid orange;
27.   margin: 0;
28.   padding: .3em;
29. }
30.
31. tfoot td {
32.   background-color: silver;
33.   border-top: 2px solid black;
34.   font-weight: bold;
35. }
36.
37. .steroids-era {
38.   font-style: italic;
39. }
40.
41. #key {
42.   background-color: lightgray;
43.   font-style: italic;
44. }
```

Evaluation Copy

```
45.
46.   tr td:nth-child(2) {
47.     text-align: center;
48.   }
49.
50.   .steroids-era td:first-child::after {
51.     content: '*';
52.   }
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

Exercise 2: Creating an Embedded Stylesheet

5 to 10 minutes

In this exercise, you will replace your external stylesheet with an embedded stylesheet.

Evaluation Copy

1. Open `CrashCourse/Exercises/index-external-styles.html` in your editor and save it as `index-embedded-styles.html`.
2. Remove the `<link>` tag, and add a `style` block.
3. Copy and paste all the CSS rules from `styles.css` to the `style` block in `index-embedded-styles.html`.
4. Open `index-embedded-styles.html` in a browser. It should look the same as `index-external-styles.html` did (see page 23).

Solution: CrashCourse/Solutions/index-embedded-styles.html

```
     -------Lines 1 through 6 Omitted-------
7.   <title>Home Run Hitters</title>
8.   <style>
9.     body {
10.      background-attachment: fixed;
11.      background-image: url(images/baseball.png);
12.      background-position: bottom;
13.      background-repeat: repeat-x;
14.    }
15.
16.    table {
17.      background-color: white;
18.      border-collapse: collapse;
19.      margin: auto;
20.      min-width: 800px;
21.    }
22.
23.    caption {
24.      font-style: italic;
25.    }
26.
27.    thead {
28.      background-color: darkblue;
29.      color: white;
30.    }
31.
32.    td,
33.    th {
34.      border: 1px solid orange;
35.      margin: 0;
36.      padding: .3em;
37.    }
38.
39.    tfoot td {
40.      background-color: silver;
41.      border-top: 2px solid black;
42.      font-weight: bold;
43.    }
44.
45.    .steroids-era {
46.      font-style: italic;
47.    }
48.
49.    #key {
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
50.       background-color: lightgray;
51.       font-style: italic;
52.     }
53.
54.     tr td:nth-child(2) {
55.       text-align: center;
56.     }
57.
58.     .steroids-era td:first-child::after {
59.       content: '*';
60.     }
61.   </style>
62.   </head>
      -------Lines 63 through 112 Omitted-------
```

Evaluation Copy

Exercise 3: Adding Inline Styles

10 to 20 minutes

In this exercise, you will add some inline styles.

1. Open `CrashCourse/Exercises/index-external-styles.html` and save it as `index-inlines-styles.html`.
2. Add inline styles to the table data cells containing the word "Giants". The style should contain the following declarations:
   - `color: darkred;`
   - `font-style: italic;`
3. Add an inline style to the table data cell containing the word "Braves". The style should contain the following declarations:
   - `color: red;`
   - `font-style: italic;`

Evaluation Copy

4. Add an inline style to the table data cells containing the word "Yankees". The style should contain the following declarations:
   - `color: navy;`
   - `font-style: italic;`
5. Add an inline style to the table data cells containing the word "Dodgers". The style should contain the following declarations:
   - `color: dodgerblue;`
   - `font-style: italic;`
6. Open `index-inline-styles.html` in a browser. The last column should now look like this:

EVALUATION COPY: Not to be used in class.

Team
Giants
Braves
Yankees
Yankees
Dodgers
N/A

Solution: CrashCourse/Solutions/index-inline-styles.html

```
      -------Lines 1 through 19 Omitted-------
20.       <tbody>
21.         <tr class="steroids-era">
22.           <td>Barry Bonds</td>
23.           <td>762</td>
24.           <td style="color: darkred; font-style: italic;">Giants</td>
25.         </tr>
26.         <tr>
27.           <td>Hank Aaron</td>
28.           <td>755</td>
29.           <td style="color: red; font-style: italic;">Braves</td>
30.         </tr>
31.         <tr>
32.           <td>Babe Ruth</td>
33.           <td>714</td>
34.           <td style="color: navy; font-style: italic;">Yankees</td>
35.         </tr>
36.         <tr class="steroids-era">
37.           <td>Alex Rodriguez</td>
38.           <td>696</td>
39.           <td style="color: navy; font-style: italic;">Yankees</td>
40.         </tr>
41.         <tr>
42.           <td>Albert Pujols</td>
43.           <td>679</td>
44.           <td style="color: dodgerblue; font-style: italic;">Dodgers</td>
45.         </tr>
46.       </tbody>
      -------Lines 47 through 59 Omitted-------
```

Evaluation Copy

1.10. and

The and tags are used in conjunction with Cascading Style Sheets. By themselves, they do very little. In fact, the tag without CSS has no visual effect on its contents. The only visual effect of the tag is to block off its contents, similar to putting a <br> tag before and after a section on the page.

Like all tags, the and tags can take the class, id, and style attributes. It is through these attributes that styles are applied to the elements. Unlike p (paragraph) elements, main (main

EVALUATION COPY: Not to be used in class.

content) elements, `header` (header content) elements, etc., `div` elements and `span` elements do not inherently represent anything. `div` elements are generic content containers for *flow* content and `span` elements are generic containers of *phrasing* content.

Flow and Phrasing Content

Phrasing content is content that *can fit* inside a sentence: images, emphasized text, links (usually), etc. Flow content is content that *cannot fit* inside a sentence: headings, tables, sections, etc.

Evaluation Copy

For more information on content categories, see `https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Content_categories`.

The following demo shows how you can apply styles do `div` and `span` elements:

Demo 1.7: CrashCourse/Demos/div-and-span.html

```
1.  <!DOCTYPE html>
2.  <html lang="en">
3.  <head>
4.  <meta charset="UTF-8">
5.  <meta name="viewport" content="width=device-width,initial-scale=1">
6.  <link href="../normalize.css" rel="stylesheet">
7.  <style>
8.    div {
9.      border-color: blue;
10.     border-style: groove;
11.     border-width: 2em;
12.     font-family: Verdana;
13.     font-size: 1em;
14.     padding: .5em;
15.   }
16.
17.   span {
18.     color: red;
19.     font-style: italic;
20.   }
21. </style>
22. <title>Div and Span</title>
23. </head>
24. <body class="webucator">
25. <div>
26.   <p>
27.     All of this text is in a div, and
28.     <span>this text is also in a span</span>.
29.   </p>
30. </div>
31. </body>
32. </html>
```

Evaluation Copy

This page will render as follows:

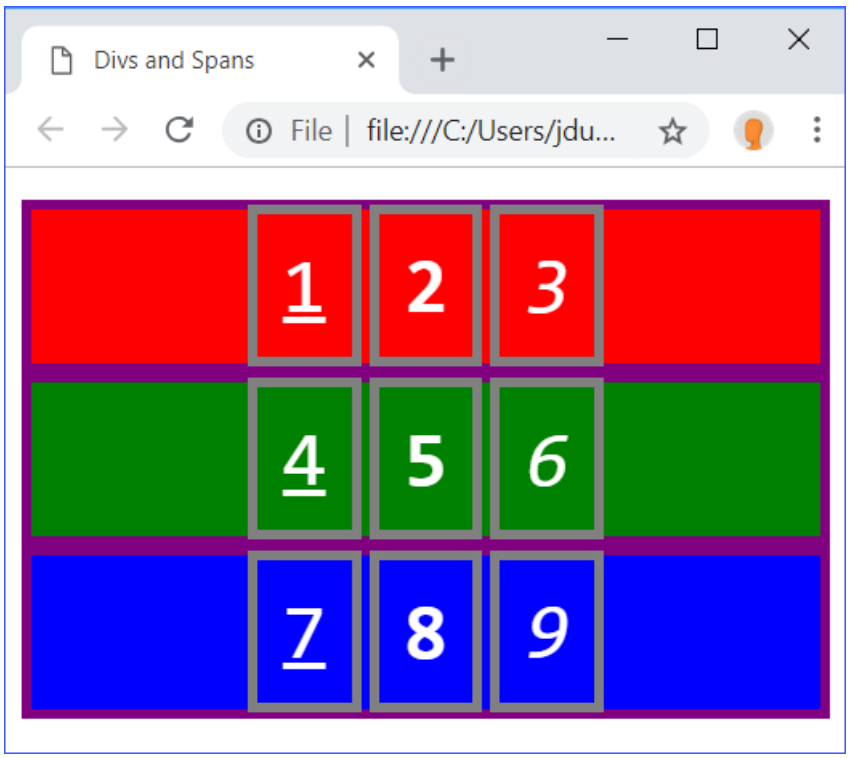
EVALUATION COPY: Not to be used in class.

Div and Span
File
se/Demos/div-and-span.html
All of this text is in a div, and this text is also in a span.

Exercise 4: Styling div and span

10 to 20 minutes

In this exercise, you will add `class` and `id` attributes to `<div>` and `<span>` tags on an already existing HTML page. Open `CrashCourse/Exercises/divs-and-spans.html` and `CrashCourse/Exercises/styles-divs-spans.css` in your editor. You will need to modify the HTML page based on the code in the CSS page. You will not need to modify the CSS page. Your goal is to make the page render as follows:



There are no step-by-step instructions. Review the rules in the external stylesheet (`styles-divs-spans.css`) and apply classes and ids as appropriate.

EVALUATION COPY: Not to be used in class.

Solution: CrashCourse/Solutions/divs-and-spans.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,initial-scale=1">
<link href="../normalize.css" rel="stylesheet">
<link href="styles-divs-spans.css" rel="stylesheet">
<title>Divs and Spans</title>
</head>
<body class="webucator">
<div id="top-div">
  <span class="left-span">1</span>
  <span class="mid-span">2</span>
  <span class="right-span">3</span>
</div>
<div id="mid-div">
  <span class="left-span">4</span>
  <span class="mid-span">5</span>
  <span class="right-span">6</span>
</div>
<div id="bottom-div">
  <span class="left-span">7</span>
  <span class="mid-span">8</span>
  <span class="right-span">9</span>
</div>
</body>
</html>
```

Evaluation Copy

1.11. Media Types

Styles can be defined for different media. For example, you may want to style a page one way for viewing with a browser and a different way for viewing in print. Some possible media types are:

Media List

- all
- print
- screen
- speech

EVALUATION COPY: Not to be used in class.

To define the media type for an entire external or embedded stylesheet, the `media` attribute is added to the `<link>` or `<style>` tag, and assigned the value of a media type. If the `media` attribute is not included, the media type defaults to `all`.

```
<link href="stylesheet.css" rel="stylesheet" media="screen">

<style media="all">
  /* rules */
</style>
```

It is also possible to target multiple media types within one stylesheet using `@media`.

```
@media screen {
  /* rules */
}

@media print {
  /* rules */
}
```

Evaluation Copy

`@media` is a powerful CSS tool. As you will see later, it can be used to target not just the type of media but also aspects of the user's device: screen width, screen height, orientation, etc. This is done with media queries. Media queries allow us to craft pages that are responsive, presenting different layouts for desktop computers, tablet devices, and smartphones.

The viewport meta tag

You may have noticed that all of our files have a line of code in the head that looks like this:

```
<meta name="viewport" content="width=device-width,initial-scale=1">
```

This is called the *viewport* `meta` tag and it is used to help pages adjust correctly for all devices. You should use it on all your web pages. For detailed information, see `https://developer.mozilla.org/en-US/docs/Web/HTML/Viewport_meta_tag`.

1.12. Units of Measurement

CSS allows for the use of many different units when specifying property values such as font size, border size, margins, etc. However, standards have been established for what units to use for different media types. In these lessons, except where specifically stated, you will be designing for screen.

Note that units of measurement and sizing will be covered throughout these lessons. In this section, we provide a general overview of the different units of measurement.

❖ 1.12.1. Absolute vs. Relative Units

Absolute units have a basis in the physical world; however, their actual size on the screen depends on the resolution and size of the user's device. When designing for screen, it is standard to use *pixels* for absolute units.

Relative units are relative to the size of other things (e.g., window size, font size, container element size, etc.). When designing for screen, it is standard to use *ems*, *rems*, and *percentages* for relative units.

❖ 1.12.2. Pixels

Evaluation Copy

Pixels are the only recommended absolute unit to be used when designing for screen. A common practice is to use pixels to set *root* font sizes for different screen and window sizes. Root font sizes are font sizes set on the `html` or `body` element. This allows for designs using mostly ems and rems to adjust between different screen and window sizes because ems and rems are relative to font size.

❖ 1.12.3. Ems and Rems

You should use ems and rems for almost everything when building responsive designs for screens. The exception to this rule is height and width, which are more often defined in pixels (for absolute values) and percentages or viewport units (for relative values). An *em* is the size of the font size for the current element or the parent element. For example, if the font-size of the `html` element is set to `14px`, then for all elements within that `html` element, `1em` is `14px`.

Rems are similar to ems, except for one key difference. The 'r' in *rem* stands for 'root' because `1rem` is always equal to the root font size (the font size set for the `html` element) regardless of what element it is in. This means `1rem` is consistent throughout the page, while `1em` is not. Both ems and rems can be used to accomplish the same things.

The code below illustrates the difference between ems and rems:

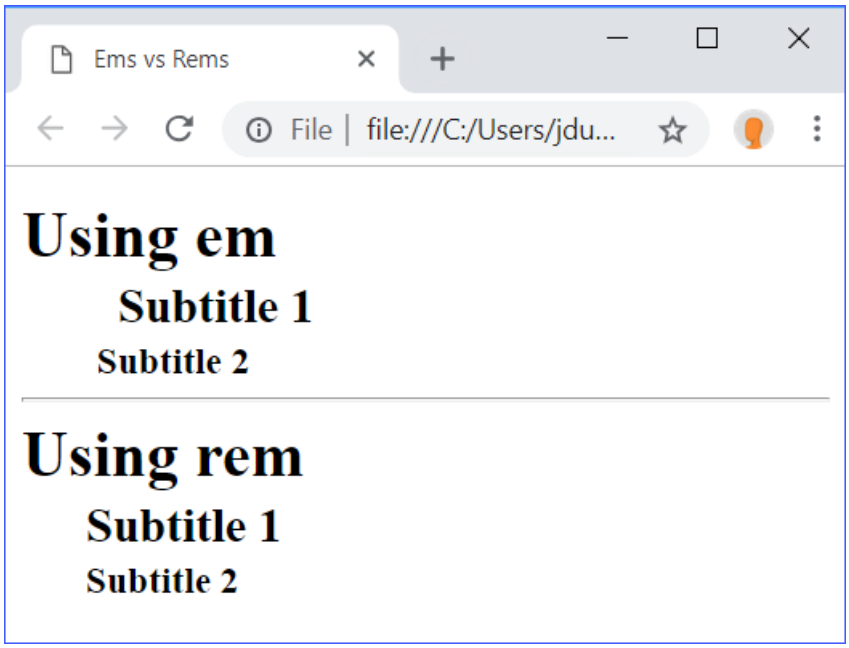
EVALUATION COPY: Not to be used in class.

Demo 1.8: CrashCourse/Demos/ems-vs-rems.html

```
1.   <!DOCTYPE HTML>
2.   <html lang="en">
3.   <head>
4.   <meta charset="UTF-8">
5.   <meta name="viewport" content="width=device-width,initial-scale=1">
6.   <link href="../normalize.css" rel="stylesheet">
7.   <style>
8.     /* Tighten vertical spacing of headings. */
9.     h1,
10.    h2,
11.    h3 {
12.      margin-bottom: .2rem;
13.      margin-top: .2rem;
14.    }
15.  </style>
16.  <title>Ems vs Rems</title>
17.  </head>
18.  <body class="webucator">
19.    <h1>Using em</h1>
20.    <h2 style="margin-left: 2em;">Subtitle 1</h2>
21.    <h3 style="margin-left: 2em;">Subtitle 2</h3>
22.
23.    <hr>
24.
25.    <h1>Using rem</h1>
26.    <h2 style="margin-left: 2rem;">Subtitle 1</h2>
27.    <h3 style="margin-left: 2rem;">Subtitle 2</h3>
28.  </body>
29.  </html>
```

Evaluation Copy

The code above will render the following:



Notice that "Subtitle 1" and "Subtitle 2" do not line up in the **Using em** section, but do line up in the **Using rem** section. This is because the ems are relative to the font sizes of their `h2` and `h3` elements, while the rems are not:

Evaluation Copy

- `2em` = twice the size of the *inherited* font.
- `2rem` = twice the size of the *base* font.

❖ 1.12.4. Percentages

Percentages are most often used to create flexible layouts that change to fit the size of the browser window. When specifying a value such as `width` or `height` in percentages, the percentage is relative to the size of the containing element. For `font-size`, percentages work the same as ems. `100%` is equal to the inherited font size, so `100%` is essentially `1em` when specifying font size. It is most common to use percentages to specify values like `width` and `height` to create "fluid" boxes that grow and shrink to fit within the browser window.

❖ 1.12.5. Other Units

Inches (`in`), centimeters (`cm`), millimeters (`mm`), points (`pt`), and picas (`pc`) are all absolute units that are most often used when designing for print.

EVALUATION COPY: Not to be used in class.

You might also run into viewport units, which are relative units based on the size of the viewport (the browser window). `vw` and `vh` specify sizes relative to the user's viewport width and height, respectively.

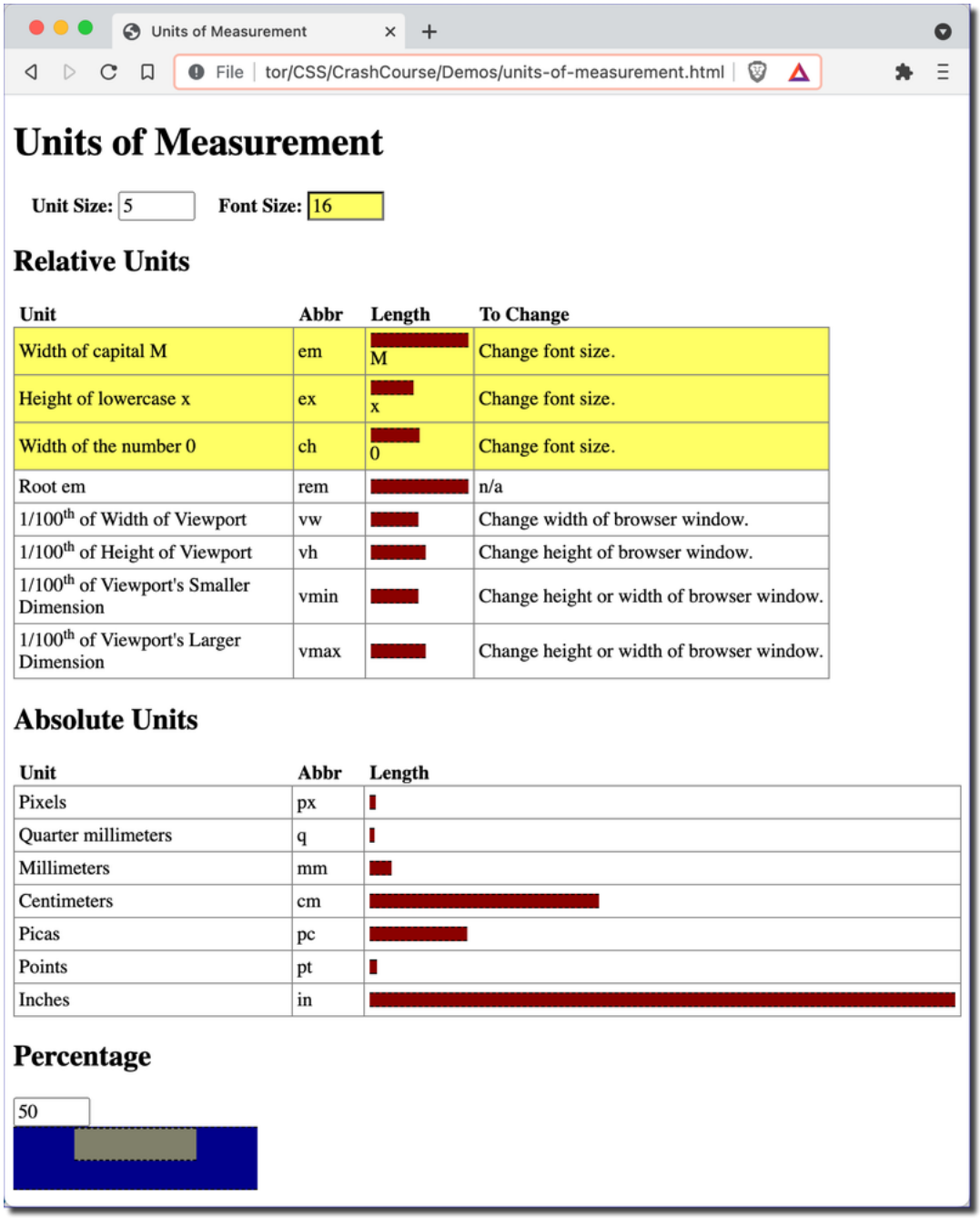
1. `vw` is 1/100th of the viewport's width.
2. `vh` is 1/100th of the viewport's height.
3. `vmax` specifies size relative to whichever dimension is larger: width or height.
4. `vmin` specifies size relative to whichever dimension is smaller: width or height.

Evaluation Copy

Viewport units are used for designing for screen, but they are not as commonly used as other screen units, in part because the specification is not clear on how they should behave on mobile[7]. There are newer units called *dynamic viewport units* coming, but major browsers do not yet support them[8].

If you would like to play with the different CSS units to see how they compare on a screen, open `CrashCourse/Demos/units-of-measurement.html` (shown below) in your browser:

7. https://www.bram.us/2021/07/08/the-large-small-and-dynamic-viewports/
8. https://caniuse.com/?search=viewport%20units



EVALUATION COPY: Not to be used in class.

CSS Values and Units

For even more detailed information on CSS values and units see `https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Values_and_units`.

1.13. Inheritance

By default, all CSS properties are either **inherited** or **non-inherited** properties. This difference determines what happens when a property for an element goes unspecified.

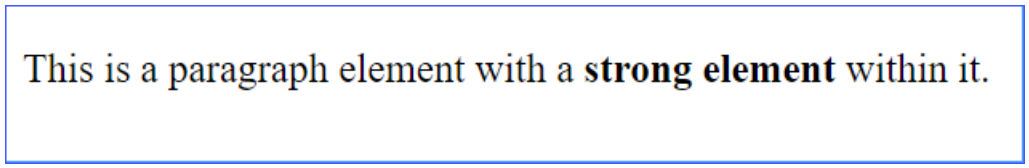
If an inherited property for an element goes unspecified, then that element will inherit the value from its parent element. A common inherited property is `font-size`:

Demo 1.9: CrashCourse/Demos/inherited.html

Evaluation Copy

```
1.   <!DOCTYPE HTML>
2.   <html>
3.   <head>
4.   <meta charset="UTF-8">
5.   <meta name="viewport" content="width=device-width,initial-scale=1">
6.   <link href="../normalize.css" rel="stylesheet">
7.   <style>
8.     p {
9.       font-size: 20px;
10.    }
11.  </style>
12.  <title>Inherited Properties</title>
13.  </head>
14.  <body class="webucator">
15.      <p>This is a paragraph element with a
16.          <strong>strong element</strong> within it.</p>
17.  </body>
18.  </html>
```

The above code will render the following:



Notice that the `strong` element has the same font size as the `p` element, even though `font-size` was not specified for it.

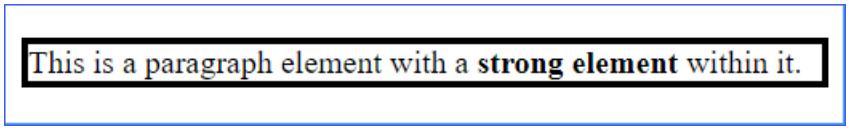
If a non-inherited property for an element goes unspecified, then that element will get the initial (default) value of that property. A common non-inherited property is `border`:

Demo 1.10: CrashCourse/Demos/non-inherited.html

```
1.   <!DOCTYPE HTML>
2.   <html>
3.   <head>
4.   <meta charset="UTF-8">
5.   <meta name="viewport" content="width=device-width,initial-scale=1">
6.   <link href="../normalize.css" rel="stylesheet">
7.   <style>
8.     p {
9.       border: 0.2rem solid black;
10.    }
11.  </style>
12.  <title>Non-Inherited Properties</title>
13.  </head>
14.  <body class="webucator">
15.      <p>This is a paragraph element with a
16.        <strong>strong element</strong> within it.</p>
17.  </body>
18.  </html>
```

Evaluation Copy

The above code will render the following:



Notice that the `strong` element does not also have its own border. This is because the initial value for `border` is `none` and `border` is **not** an inherited property.

EVALUATION COPY: Not to be used in class.

❖ 1.13.1. The inherit Value

You can force a property to inherit the value of its parent by setting its value to `inherit`.

The `inherit` property is most often used to override other rules, as illustrated in the following demo:

Demo 1.11: CrashCourse/Demos/inherit-styles.css

```
1.   h2 {
2.     color: blue;
3.   }
4.
5.   article h2 {
6.     color: inherit;
7.   }
8.
9.   #article-red {
10.    color: red;
11.  }
12.
13.  #article-green {
14.    color: green;
15.  }
16.
17.  #article-purple {
18.    color: purple;
19.  }
```

Evaluation Copy

Demo 1.12: CrashCourse/Demos/inherit.html

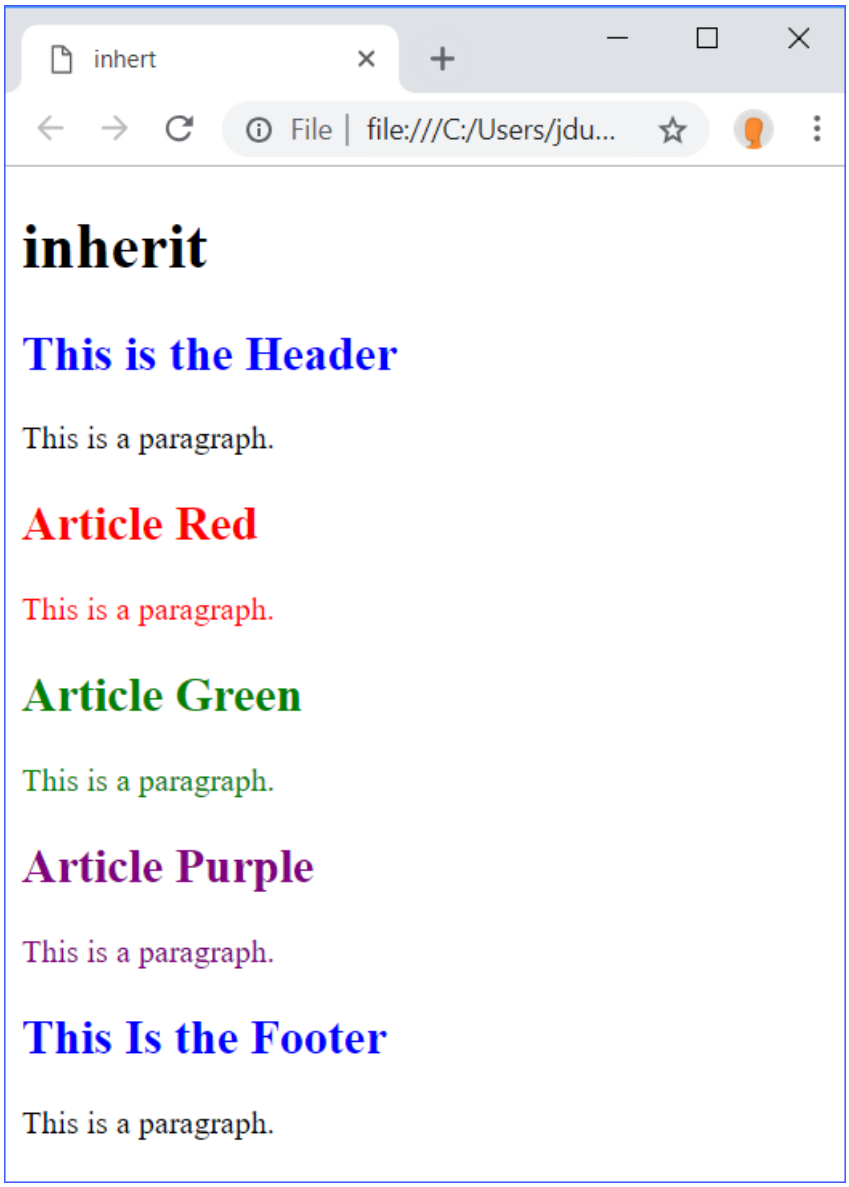
```
      -------Lines 1 through 9 Omitted-------
10.   <body class="webucator">
11.     <h1>inherit</h1>
12.     <header>
13.       <h2>This is the Header</h2>
14.       <p>This is a paragraph.</p>
15.     </header>
16.     <article id="article-red">
17.       <h2>Article Red</h2>
18.       <p>This is a paragraph.</p>
19.     </article>
20.     <article id="article-green">
21.       <h2>Article Green</h2>
22.       <p>This is a paragraph.</p>
23.     </article>
24.     <article id="article-purple">
25.       <h2>Article Purple</h2>
26.       <p>This is a paragraph.</p>
27.     </article>
28.     <footer>
29.       <h2>This Is the Footer</h2>
30.       <p>This is a paragraph.</p>
31.     </footer>
32.   </body>
33.   </html>
```

Evaluation Copy

The above code will render the following:[9]

9. If you are reading this in black and white, be sure to open the page in your browser.

EVALUATION COPY: Not to be used in class.



Notice that this rule says that all `h2` elements should be blue:

```
h2 {
  color: blue;
}
```

This rule would normally trump inheritance. In other words, by default the `h2` elements would inherit the color of their parent element, but the rule above overrides that, specifying that all `h2` elements should be blue.

But in the result, the `h2` elements in the `article` elements are **not** blue. Why?

Because this more specific rule says that `h2` elements within `article` elements should inherit the `color` property from their parent elements:

```
article h2 {
    color: inherit;
  }
```

Conclusion

Cascading Style Sheets provide a far better way of formatting HTML pages than the traditional use of HTML tags. In this lesson, you have developed a foundation for creating and applying CSS rules.

EVALUATION COPY: Not to be used in class.

LESSON 2
CSS Fonts

Topics Covered

☑ `font-family`

☑ `@font-face`.

☑ `font-size`

☑ `font-style`

☑ `font-variant`

☑ `font-weight`

☑ `line-height`

☑ Shorthand properties.

☑ `font`

Evaluation Copy

Introduction

In this lesson, you will learn to use CSS to modify font properties. You will also learn to work with CSS shorthand properties.

2.1. font-family

The `font-family` property is used in CSS to specify the font applied to an element. You can specify by font name or font category.

❖ 2.1.1. Specifying by Font Name

When you specify a font by font name, the browser will look for the named font on the end user's computer. If it finds it, the text will be displayed in that font. For example, the following rule would make the font of all `<p>` tags Arial:

```
p {
  font-family: Arial;
}
```

If the Arial font were not found on the end user's computer, the browser would display a default font. If you are concerned that the font name you want to use might not be found on a user's computer, you can provide a list of options.

```
p {
  font-family: Arial, Helvetica;
}
```

In this case, the browser will first look for Arial. If it doesn't find Arial, it will then look for Helvetica.

❖ 2.1.2. Specifying Font by Category

When you specify a font by category, the browser will use the font the user's computer specifies for that category. For example, for monospace, the computer might specify Courier. Some common font family categories are listed below:

Evaluation Copy

- `cursive`
- `fantasy`
- `monospace`
- `sans-serif`
- `serif`

To be extra safe, designers often specify a couple specific options followed by a font family category, like so:

```
p {
  font-family: Arial, Helvetica, sans-serif;
}
```

This way, if neither Arial nor Helvetica is found, the browser at least knows to use some sans-serif font.

The following code samples illustrate how `font-family` works:

EVALUATION COPY: Not to be used in class.

Demo 2.1: Fonts/Demos/font-family-styles.css

```
1.   #cursive-div {
2.     font-family: 'Cool Linked Font', cursive;
3.   }
4.
5.   #fantasy-div {
6.     font-family: 'Cool Linked Font', fantasy;
7.   }
8.
9.   #monospace-div {
10.    font-family: 'Courier New', monospace;
11.  }
12.
13.  #sans-serif-div {
14.    font-family: Tahoma, Verdana, Arial, sans-serif;
15.  }
16.
17.  #serif-div {
18.    font-family: Times, 'Times New Roman', Georgia, serif;
19.  }
```

Evaluation Copy

Notice that `#cursive-div` and `#fantasy-div` both have a 'Cool Linked Font' listed before their generic family names. Cursive and fantasy fonts are not as popular or as widely supported as the other three, so it is common to load your own fonts for these two font families.

Also notice that fonts that have multi-word names (e.g., Times New Roman) should be contained in either single or double quotes.

Here is an HTML page that uses the stylesheet shown above:

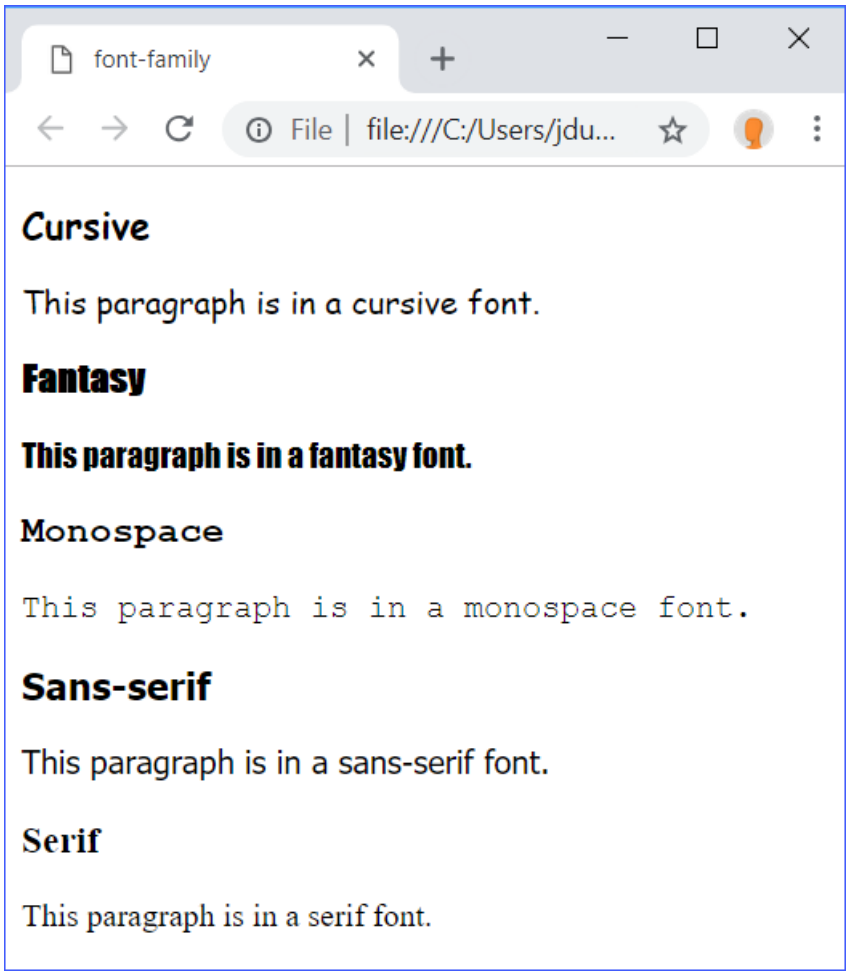
Demo 2.2: Fonts/Demos/font-family.html

```
      -------Lines 1 through 9 Omitted-------
10.   <body class="webucator">
11.     <div id="cursive-div">
12.       <h3>Cursive</h3>
13.       <p>This paragraph is in a cursive font.</p>
14.     </div>
15.     <div id="fantasy-div">
16.       <h3>Fantasy</h3>
17.       <p>This paragraph is in a fantasy font.</p>
18.     </div>
19.     <div id="monospace-div">
20.       <h3>Monospace</h3>
21.       <p>This paragraph is in a monospace font.</p>
22.     </div>
23.     <div id="sans-serif-div">
24.       <h3>Sans-serif</h3>
25.       <p>This paragraph is in a sans-serif font.</p>
26.     </div>
27.     <div id="serif-div">
28.       <h3>Serif</h3>
29.       <p>This paragraph is in a serif font.</p>
30.     </div>
31.   </body>
32.   </html>
```

Evaluation Copy

The code above will render as follows:

EVALUATION COPY: Not to be used in class.



The web-safe fonts are those fonts that are most commonly installed on computers and are therefore safe to use in your CSS. You can see the full list of web-safe fonts on cssfontstack.com[10].

2.2. @font-face

The CSS `@font-face` rule enables the loading of a font file, and thus offers us the ability to use any font, not just those our users happen to have loaded on their computer or device. The `@font-face` rule is defined in the W3C's CSS Fonts Module Level 3[11] specification. As the W3C specification states:

10. https://www.cssfontstack.com/
11. https://www.w3.org/TR/css-fonts-3/

> The `@font-face` rule allows for linking to fonts that are automatically fetched and activated when needed. This allows authors to select a font that closely matches the design goals for a given page rather than limiting the font choice to a set of fonts available on a given platform. A set of font descriptors define the location of a font resource, either locally or externally, along with the style characteristics of an individual face. Multiple `@font-face` rules can be used to construct font families with a variety of faces. Using CSS font matching rules, a user agent can selectively load only those faces that are needed for a given piece of text.

In the CSS, you define your own `font-family` and associate it with a font file:[12]

```
@font-face {
  font-family: Gentium;
  src: url('Gentium.ttf');
}
```

Evaluation Copy

A subsequent CSS rule could style a given element with the newly defined `font-family`:

```
div.newsitem {
  font-family: Gentium;
}
```

Browsers will download the specified font file and will use it to render the designated content.

Different browsers support different font file formats. To address these differences, the specification for the `@font-face` rule allows for a series of font files, with format hints to aid browsers in selecting the appropriate file for the font:

```
@font-face {
  font-family: bodytext;
  src: url(ideal-sans-serif.woff) format("woff"),
       url(basic-sans-serif.ttf) format("opentype");
}
```

12. We will cover how to get font files soon.

EVALUATION COPY: Not to be used in class.

Here's a list of the formats supported by the W3C specification:[13]

1. **Embedded OpenType**
    - **Format string:** `embedded-opentype`
    - **Extension(s):** `.eot`
    - Only supported in Internet Explorer.
2. **OpenType**
    - **Format string:** `opentype`
    - **Extension(s):** `.ttf`, `.otf`
    - Well supported, but in Internet Explorer it only works with fonts set to installable.
3. **SVG Font**
    - **Format string:** `svg`
    - **Extension(s):** `.svg`, `.svgz`
    - Not well supported.
4. **TrueType**
    - **Format string:** `truetype`
    - **Extension(s):** `.ttf`
    - Well supported, but in Internet Explorer it only works with fonts set to installable.
5. **WOFF (Web Open Font Format)**
    - **Format string:** `woff`
    - **Extension(s):** `.woff`
    - Well supported.
6. **WOFF 2.0**
    - **Format string:** `woff2`
    - **Extension(s):** `.woff2`
    - Supported well in most recent browsers, but not supported in Internet Explorer or older browsers.

Evaluation Copy

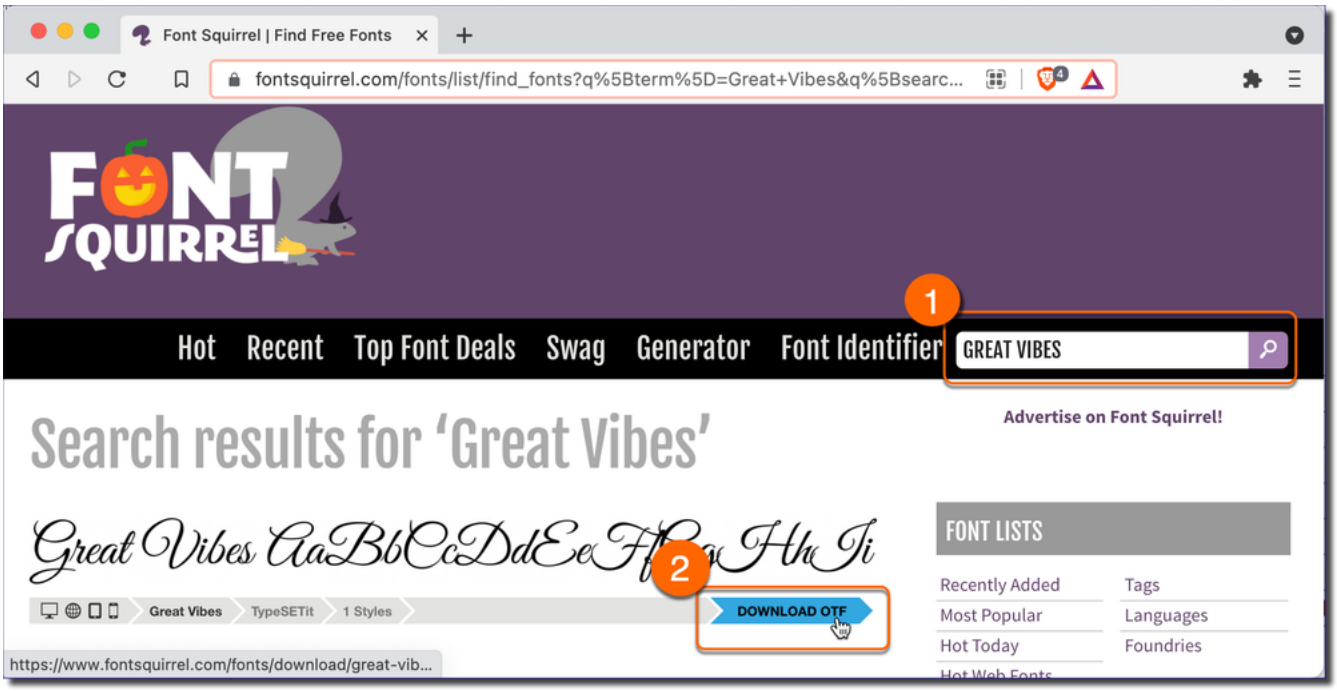
13. https://www.w3.org/TR/2018/REC-css-fonts-3-20180920/#fontformats

In the `@font-face` rule, you can specify additional properties, such as `font-style` and `font-weight`, which we will cover soon.

❖ 2.2.1. Getting Fonts

Let's look at a typical example, in which you will download the font "Great Vibes" from Font Squirrel[14], a popular resource for free fonts and then convert it into several different file font formats:

1. Search `https://www.fontsquirrel.com` for "Great Vibes":

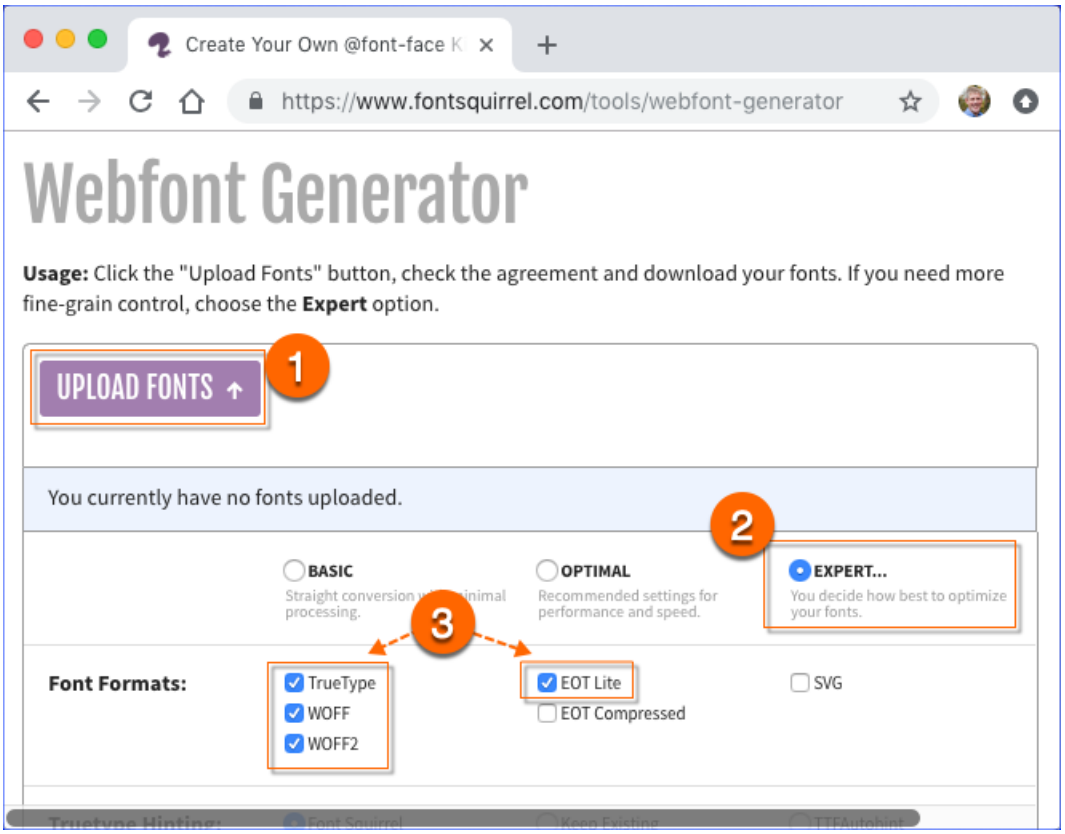


   Click the **DOWNLOAD OTF** button to download the zip file containing the font. You can save it wherever you like.

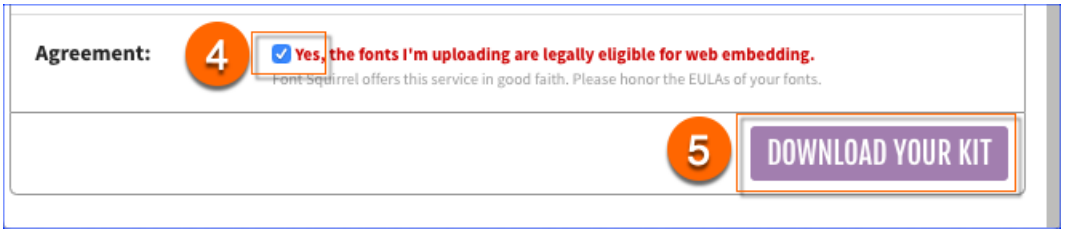
2. Unzip the file. You will see a `GreatVibes-Regular.otf` file.
3. To convert the `GreatVibes-Regular.otf` to other file types, use the Font Squirrel Webfont Generator.[15]
4. Upload the OpenType file you just downloaded and choose four font formats to convert it to. Select the "Expert" option as the other options do not allow for the specific selection of font formats:

14. https://www.fontsquirrel.com/
15. https://www.fontsquirrel.com/tools/webfont-generator

EVALUATION COPY: Not to be used in class.



5. At the bottom of the generator, check the agreement box and click the **DOWNLOAD YOUR KIT** button:



   You can save it wherever you like.

6. Unzip the file. Among its contents, you will see:

   - `greatvibes-regular-demo.html` – a demo HTML file using the new font.

`stylesheet.css` – A stylesheet demonstrating how to create the `@font-face` rule:

```
@font-face {
    font-family: 'great_vibesregular';
    src: url('greatvibes-regular-webfont.eot');
    src: url('greatvibes-regular-webfont.eot?#iefix') format('embedded-
opentype'),
         url('greatvibes-regular-webfont.woff2') format('woff2'),
         url('greatvibes-regular-webfont.woff') format('woff');
    font-weight: normal;
    font-style: normal;
}
```

- Three font files:
  - `greatvibes-regular-webfont.eot`
  - `greatvibes-regular-webfont.woff`
  - `greatvibes-regular-webfont.woff2`

Evaluation Copy

7. Open `greatvibes-regular-demo.html` to see the demo, which illustrates the use of "Great Vibes" in different sizes, shows a sample page that uses only the "Great Vibes" font, and provides charts showing special characters and glyphs.

Internet Explorer Bug

In the generated stylesheet, notice the `?#iefix` appended to the `.eot` file. This tricks versions of Internet Explorer to get around a known bug for not loading multiple versions of the font file.

We made a few edits to the generated stylesheet and attached it to a simple HTML page as you can see in the samples below:

EVALUATION COPY: Not to be used in class.

Demo 2.3: Fonts/Demos/font-face-styles.css

```
1.   @font-face {
2.     font-family: 'GreatVibes';
3.     src: url('fonts/greatvibes-regular-webfont.eot');
4.     src: url('fonts/greatvibes-regular-webfont.eot?#iefix')
5.           format('embedded-opentype'),
6.          url('fonts/greatvibes-regular-webfont.woff2')
7.           format('woff2'),
8.          url('fonts/greatvibes-regular-webfont.woff')
9.           format('woff'),
10.         url('fonts/greatvibes-regular-webfont.ttf')
11.          format('truetype');
12.    font-weight: normal;
13.    font-style: normal;
14.  }
15.
16.  p {
17.    font-family: GreatVibes;
18.  }
```

Evaluation Copy

We changed the font name from what the generated stylesheet suggested, and we adjusted the paths for the urls. In the `Fonts/Demos` folder, we moved all of our "Great Vibes" font files into a folder called `fonts`, and we deleted the unnecessary generated files (e.g., the demo file, the original stylesheet, etc.).

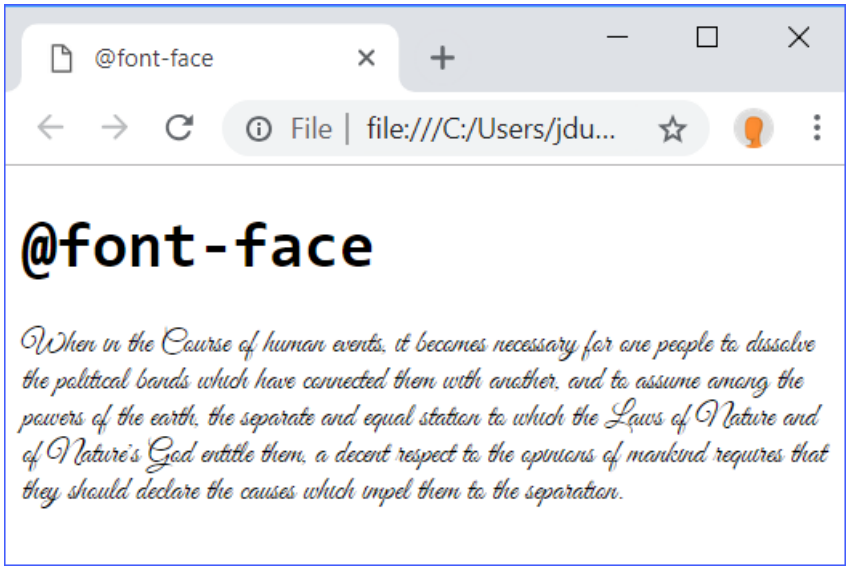
Finally, after the `@font-face` rule, we used the "Great Vibes" font to style the first paragraph of the U.S. Declaration of Independence.

Here's the HTML:

Demo 2.4: Fonts/Demos/font-face.html

```
      -------Lines 1 through 9 Omitted-------
10.   <body class="webucator">
11.     <h1><code>@font-face</code></h1>
12.     <p>When in the Course of human events, it becomes necessary
13.       for one people to dissolve the political bands which have
14.       connected them with another, and to assume among the powers
15.       of the earth, the separate and equal station to which the
16.       Laws of Nature and of Nature's God entitle them,  a decent
17.       respect to the opinions of mankind requires that they should
18.       declare the causes which impel them to the separation.</p>
19.   </body>
20.   </html>
```

And here is the resulting page:



2.3. font-size

`font-size` is an unexpectedly complicated CSS property. It is difficult to quickly grasp the different `font-size` units, how they relate to each other, and how they change depending on the font family. It will require some practice and experimentation to get used to.

EVALUATION COPY: Not to be used in class.

❖ 2.3.1. Relative font-size Terms

In addition to all the units of measurement we discussed in the Crash Course lesson (see page 42), font size can be defined using the following terms:

- `xx-large`
- `x-large`
- `large`
- `medium`
- `small`
- `x-small`
- `xx-small`
- `smaller`
- `larger`

Evaluation Copy

The terms `xx-small` to `xx-large` are absolute-size keywords relative to the user's default font size (`medium`). The terms `smaller` and `larger` are relative-size keywords, meaning that they change the font size of an element relative to its parent element's font size. The following examples illustrate these terms:

Demo 2.5: Fonts/Demos/font-size-styles.css

```
1.   .biggest {
2.     font-size: xx-large;
3.   }
4.
5.   .second-biggest {
6.     font-size: x-large;
7.   }
8.
9.   .third-biggest {
10.    font-size: large;
11.  }
12.
13.  .medium {
14.    font-size: medium;
15.  }
16.
17.  .third-smallest {
18.    font-size: small;
19.  }
20.
21.  .second-smallest {
22.    font-size: x-small;
23.  }
24.
25.  .smallest {
26.    font-size: xx-small;
27.  }
28.
29.  .larger {
30.    font-size: larger;
31.  }
32.
33.  .smaller {
34.    font-size: smaller;
35.  }
```

Evaluation Copy

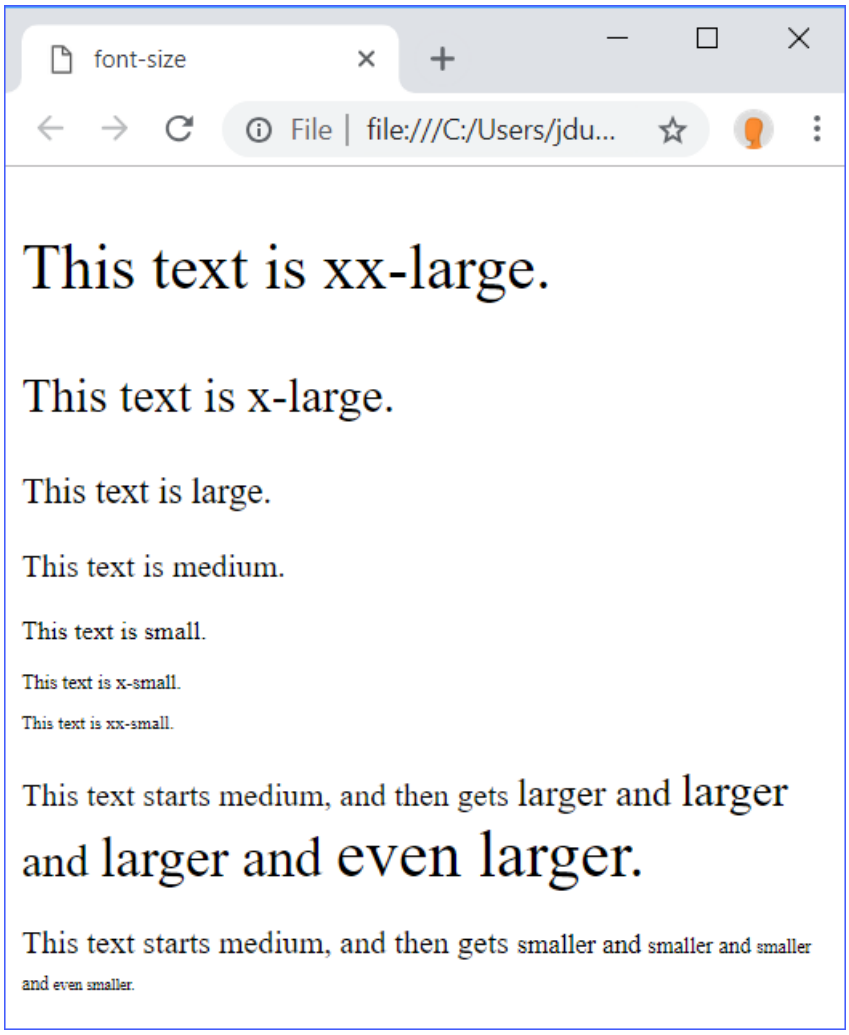
EVALUATION COPY: Not to be used in class.

Demo 2.6: Fonts/Demos/font-size.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.    <p class="biggest">This text is xx-large.</p>
12.    <p class="second-biggest">This text is x-large.</p>
13.    <p class="third-biggest">This text is large.</p>
14.    <p class="medium">This text is medium.</p>
15.    <p class="third-smallest">This text is small.</p>
16.    <p class="second-smallest">This text is x-small.</p>
17.    <p class="smallest">This text is xx-small.</p>
18.    <div id="compare-div" class="medium">
19.      <p>This text starts medium, and then gets
20.        <span class="larger">larger and
21.          <span class="larger">larger and
22.            <span class="larger">larger and
23.              <span class="larger">even larger.
24.              </span>
25.            </span>
26.          </span>
27.        </span>
28.      </p>
29.      <p>This text starts medium, and then gets
30.        <span class="smaller">smaller and
31.          <span class="smaller">smaller and
32.            <span class="smaller">smaller and
33.              <span class="smaller">even smaller.
34.              </span>
35.            </span>
36.          </span>
37.        </span>
38.      </p>
39.    </div>
40.  </body>
41.  </html>
```

Evaluation Copy

The code above will output the following:



Visualizing Ems and Rems

For a visual representation of fonts using `ems` and `rems`, open `CrashCourse/Demos/units-of-measurement.html` in your browser.

❖ 2.3.2. Best Practices

Most experts agree that font size should be defined in relative units (e.g., `em`, `rem`, or `%`) or in terms (e.g., `large`, `small`, etc.). This is because absolute font sizes can make pages inaccessible to people who have difficulty seeing. In most browsers, a user can change both the "zoom" of the page and, separately, the default font size. While "zooming" will increase the size of all elements (font included)

EVALUATION COPY: Not to be used in class.

on the page, changing the default browser font size won't have any effect on fonts whose size in CSS is specified using absolute, rather than relative, units.

2.4. font-style

Currently, the only use for `font-style` is to italicize (and unitalicize) text. The values are listed below:

- `normal`
- `italic`
- `oblique`

However, `italic` and `oblique` are displayed in the same way. Since `italic` has better support, you should use it. For more on `font-style`, see `https://developer.mozilla.org/en-US/docs/Web/CSS/font-style`.

Evaluation Copy

2.5. font-variant

The most common and well-supported use of `font-variant` is to turn lowercase letters into small caps. The values are listed below:

- `normal`
- `small-caps`

The screenshot below shows an `h1` tag with small-caps:[16]

16. The file used for this screenshot is `Fonts/Demos/font-variant.html`.

`font-variant` is actually a shorthand property (more on shorthand properties soon) for several longhand properties, but those are rarely used and poorly supported, so you shouldn't worry about them. For more details, see `https://developer.mozilla.org/en-US/docs/Web/CSS/font-variant`.

Evaluation Copy

2.6. font-weight

The weight of a font determines how thick (or bold) it is. Possible values are:

- Any number between `1` and `1000`
- `bold`
- `bolder`
- `lighter`
- `normal`

The numeric values are used for fonts that can have many different degrees of boldness. Most fonts, however, are either `bold` or `normal` (not bold). `bold` is the same as `700` and `normal` is the same as `400`. `bolder` and `lighter` work similarly to `larger` and `smaller` for `font-size`.

Here is a stylesheet with rules setting `font-weight` for elements on the page:

EVALUATION COPY: Not to be used in class.

Demo 2.7: Fonts/Demos/font-weight-styles.css

```
1.   html {
2.     font-family:'Segoe UI', 'Open Sans',
3.     'Helvetica Neue', sans-serif;
4.   }
5.   #div1 {
6.     font-weight: normal;
7.   }
8.   #div2 {
9.     font-weight: bold;
10.  }
11.  #div3 {
12.    font-weight: 400;
13.  }
14.  #div4 {
15.    font-weight: 1000;
16.  }
17.  .bolder {
18.    font-weight: bolder;
19.  }
20.  .lighter {
21.    font-weight: lighter;
22.  }
23.  #li1 {
24.    font-weight: 100;
25.  }
26.  #li2 {
27.    font-weight: 400;
28.  }
29.  #li3 {
30.    font-weight: 600;
31.  }
32.  #li4 {
33.    font-weight: 700;
34.  }
35.  #li5 {
36.    font-weight: 900;
37.  }
```

Evaluation Copy

We used the Segoe UI font because it has precise control over its weight. If we were to use a font that didn't have this control, like Times, then each element would only render either `bold` or `normal`.

And here is an HTML page that uses this stylesheet:

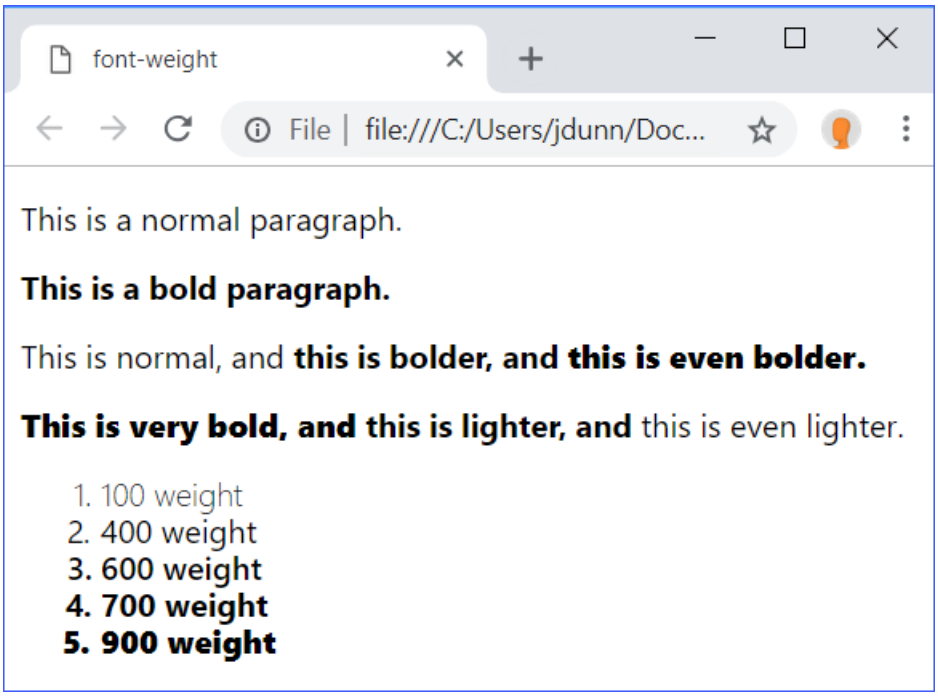
Demo 2.8: Fonts/Demos/font-weight.html

```
      -------Lines 1 through 9 Omitted-------
10.   <body class="webucator">
11.     <div id="div1">
12.       <p>This is a normal paragraph.</p>
13.     </div>
14.     <div id="div2">
15.       <p>This is a bold paragraph.</p>
16.     </div>
17.     <div id="div3">
18.       <p>This is normal, and
19.         <span class="bolder">this is bolder, and
20.           <span class="bolder">this is even bolder.
21.           </span>
22.         </span>
23.       </p>
24.     </div>
25.     <div id="div4">
26.       <p>This is very bold, and
27.         <span class="lighter">this is lighter, and
28.           <span class="lighter">this is even lighter.
29.           </span>
30.         </span>
31.       </p>
32.     </div>
33.     <ol>
34.       <li id="li1">100 weight</li>
35.       <li id="li2">400 weight</li>
36.       <li id="li3">600 weight</li>
37.       <li id="li4">700 weight</li>
38.       <li id="li5">900 weight</li>
39.     </ol>
40.   </body>
41.   </html>
```

Evaluation Copy

The code above will render the following:

EVALUATION COPY: Not to be used in class.



font-weight Values

In early versions of `font-weight`, only intervals of `100` (`100` to `900`) were accepted, so if you use a more exact number value, make sure both the font you're using and your target browsers support that value. For more on `font-weight`, see `https://developer.mozilla.org/en-US/docs/Web/CSS/font-weight`.

Evaluation Copy

2.7. line-height

`line-height` determines the amount of vertical space used for lines, most commonly in text.

Although there are several options for `line-height` values, the best option is to use a unit-less number. This number is relative to the `font-size` of the text. For example, if you set `line-height` to `1`, there will be no space between the text on a line and the text on lines before and after that line.

The default `line-height` value is approximately `1.2`. You can set that specifically using:

```
line-height: 1.2;
```

Or you can set it using:

```
line-height: normal;
```

The latter will set the `line-height` to the exact default value, which is dependent on the browser and the font.

To add more space between lines, set `line-height` to a value higher than `1.2` and to make lines tighter, set it to a value less than `1.2`.

Evaluation Copy

Other Values of line-height

While `line-height` can also take a length with units (e.g., `em` or `px`), we recommend you stick with a unit-less number.

The following samples shows some different line heights:

EVALUATION COPY: Not to be used in class.

Demo 2.9: Fonts/Demos/line-height-styles.css

```
1.   html {
2.     font-size: 16px;
3.   }
4.
5.   #normal {
6.     line-height: normal;
7.   }
8.
9.   #p1 {
10.    line-height: 1;
11.  }
12.
13.  #p2 {
14.    line-height: 1.1;
15.  }
16.
17.  #p3 {
18.    line-height: 1.2;
19.  }
20.
21.  #p4 {
22.    line-height: 1.3;
23.  }
24.
25.  #p5 {
26.    line-height: 1.5;
27.  }
28.
29.  #p6 {
30.    line-height: 2;
31.  }
```

Evaluation Copy

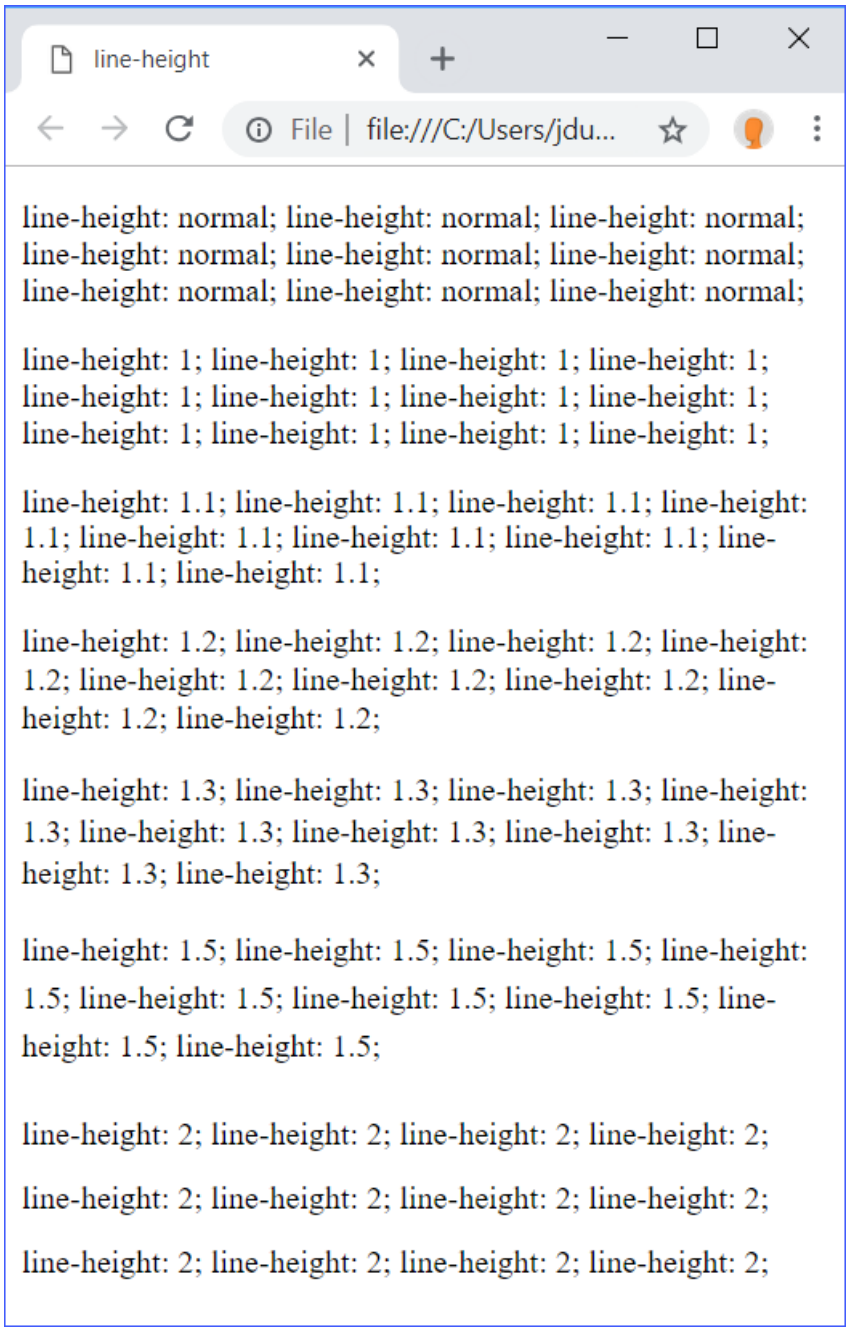
Demo 2.10: Fonts/Demos/line-height.html

```
      -------Lines 1 through 9 Omitted-------
10.   <body class="webucator">
11.     <p id="normal">
12.       line-height: normal; line-height: normal; line-height: normal;
13.       line-height: normal; line-height: normal; line-height: normal;
14.       line-height: normal; line-height: normal; line-height: normal;
15.     </p>
16.     <p id="p1">
17.       line-height: 1; line-height: 1; line-height: 1; line-height: 1;
18.       line-height: 1; line-height: 1; line-height: 1; line-height: 1;
19.       line-height: 1; line-height: 1; line-height: 1; line-height: 1;
20.     </p>
21.     <p id="p2">
22.       line-height: 1.1; line-height: 1.1; line-height: 1.1;
23.       line-height: 1.1; line-height: 1.1; line-height: 1.1;
24.       line-height: 1.1; line-height: 1.1; line-height: 1.1;
25.     </p>
26.     <p id="p3">
27.       line-height: 1.2; line-height: 1.2; line-height: 1.2;
28.       line-height: 1.2; line-height: 1.2; line-height: 1.2;
29.       line-height: 1.2; line-height: 1.2; line-height: 1.2;
30.     </p>
31.     <p id="p4">
32.       line-height: 1.3; line-height: 1.3; line-height: 1.3;
33.       line-height: 1.3; line-height: 1.3; line-height: 1.3;
34.       line-height: 1.3; line-height: 1.3; line-height: 1.3;
35.     </p>
36.     <p id="p5">
37.       line-height: 1.5; line-height: 1.5; line-height: 1.5;
38.       line-height: 1.5; line-height: 1.5; line-height: 1.5;
39.       line-height: 1.5; line-height: 1.5; line-height: 1.5;
40.     </p>
41.     <p id="p6">
42.       line-height: 2; line-height: 2; line-height: 2; line-height: 2;
43.       line-height: 2; line-height: 2; line-height: 2; line-height: 2;
44.       line-height: 2; line-height: 2; line-height: 2; line-height: 2;
45.     </p>
46.   </body>
47.   </html>
```

Evaluation Copy

The code above will render the following:

EVALUATION COPY: Not to be used in class.



For more on `line-height`, see `https://developer.mozilla.org/en-US/docs/Web/CSS/line-height`.

2.8. font

`font` is a shorthand property that encompasses the properties we have discussed in this lesson.

Shorthand Properties

Shorthand properties set multiple CSS properties at once, allowing for more concise, more readable stylesheets that save time and space. Using shorthand properties instead of longhand properties also has the benefit of reducing file size, which increases download speed.

An important thing to note about shorthand properties is that any unspecified values will be set to their default values.

The syntax for `font` is shown below. The highlighted properties are required:

```
font: font-style
      font-variant
      font-weight
      font-stretch
      font-size/line-height
      font-family;
```

Evaluation Copy

Some notes:

1. `font-stretch`, while well supported by browsers, only works with a small number of fonts.[17]
2. `font-style`, `font-variant`, `font-weight`, and `font-stretch` are optional and may be written in any order before `font-size`.
3. `line-height` is also optional, but it must be written immediately after `font-size` separated by a forward slash (`/`).
4. If one of the optional values is not included, it will be set to the default value, overriding any rules with lower precedence.

The code below illustrates the use of `font`:

17. See `https://developer.mozilla.org/en-US/docs/Web/CSS/font-stretch`.

EVALUATION COPY: Not to be used in class.

Demo 2.11: Fonts/Demos/font-styles.css

```
#div1 h1 {
  font: italic small-caps bold 2em Georgia, Times,
  'Times New Roman', sans-serif;
}

#div1 p {
  font: 1em/1.2 Arial, Helvetica, sans-serif;
}

#div2 h1 {
  font-style: italic; /* gets overridden */
  font-variant: small-caps; /* gets overridden */
  font: 2.5em 'Gill Sans', serif;
}

#div2 p {
  font: 0.9em/1.5 cursive;
}
```

Evaluation Copy

Notice that the `font-style` and `font-variant` declarations written in the `#div2 h1` rule will not be applied because they were later overridden by defaults in the `font` declaration, even though those defaults are not explicitly stated.

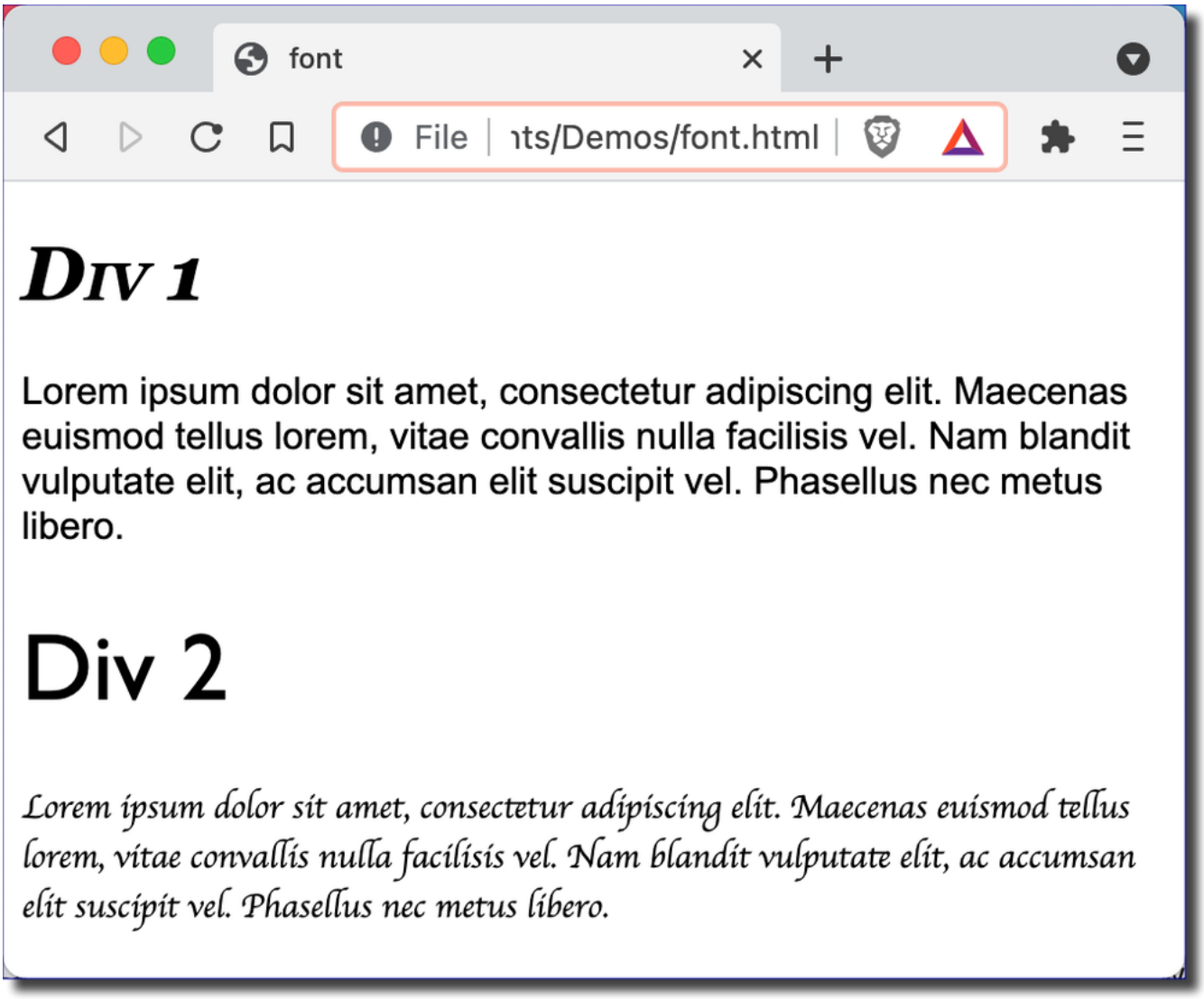
Demo 2.12: Fonts/Demos/font.html

```
      -------Lines 1 through 9 Omitted-------
10.   <body class="webucator">
11.     <div id="div1">
12.       <h1>Div 1</h1>
13.       <p>
14.         Lorem ipsum dolor sit amet, consectetur adipiscing elit.
15.         Maecenas euismod tellus lorem, vitae convallis nulla
16.         facilisis vel. Nam blandit vulputate elit, ac accumsan
17.         elit suscipit vel. Phasellus nec metus libero.
18.       </p>
19.     </div>
20.     <div id="div2">
21.       <h1>Div 2</h1>
22.       <p>
23.         Lorem ipsum dolor sit amet, consectetur adipiscing elit.
24.         Maecenas euismod tellus lorem, vitae convallis nulla
25.         facilisis vel. Nam blandit vulputate elit, ac accumsan
26.         elit suscipit vel. Phasellus nec metus libero.
27.       </p>
28.     </div>
29.   </body>
30.   </html>
```

Evaluation Copy

The code above will render the following:

EVALUATION COPY: Not to be used in class.



For more on `font`, see `https://developer.mozilla.org/en-US/docs/Web/CSS/font`.

Exercise 5: Styling Fonts

25 to 40 minutes

In this exercise, you will modify an HTML page by applying the font properties you just learned.

1. Open `Fonts/Exercises/index.html` in your editor. You will see the home page for a website called Runners Home.
2. Create an external stylesheet called `styles.css` and link to it from `index.html`.
3. Using your new knowledge of font properties, add some styles to this page. The object of this exercise is to practice using the font properties covered in this lesson. You are also welcome to download a font and use `@font-face`.
4. When you are done, open `index.html` in your browser to see the results. You are welcome to go back to the code and continue to work.

Evaluation Copy

You can design it however you like, or you can try to make it look something like this:

EVALUATION COPY: Not to be used in class.

Runners Home

- Home
- Resources
  - Calculator
  - Running Log
  - Running Terms
  - Links
- Running Advice
- Races
- Register

Hello, Stranger!

Welcome to Runners Home

Best Running Tips

There are good ways to train and bad ways to train. To great the most out of your runs... Read more...

Best Health Tips

In addition to keeping up with your physical training, you must be sure to take care of your general health... Read more...

More advice articles...

Purpose

Runners Home is dedicated to providing you with:

1. the most up-to-date information on running races.
2. the best resources for runners.

Newsletter

Email: Sign Up

Be the first to hear about our great offers.
Sign up for our newsletter today!

Disclosure: This is not a real website.

© 2022 Runners Home. All rights reserved.

info@runners-home.com

Solution: Fonts/Solutions/styles.css

```
1.  html {
2.    font-size: 16px;
3.  }
4.
5.  body {
6.    font: 1rem/1.3 Verdana, Geneva, Tahoma, sans-serif;
7.  }
8.
9.  #welcome {
10.   font-size: larger;
11. }
12.
13. article h2 {
14.   font-variant: small-caps;
15. }
16.
17. article p {
18.   line-height: 2;
19. }
20.
21. aside label {
22.   font-weight: bold;
23. }
24.
25. footer {
26.   font-size: smaller;
27.   font-style: italic;
28. }
```

Evaluation Copy

Conclusion

In this lesson, you have learned to use CSS font properties.

EVALUATION COPY: Not to be used in class.

LESSON 3
Color and Opacity

Topics Covered

☑ `color`

☑ `opacity`

Introduction

In this lesson, you will learn to add color and opacity to your HTML pages.

3.1. About Color and Opacity

Evaluation Copy

CSS makes it really easy to add color to your HTML pages. Almost every element can have color added to it, and CSS provides multiple methods to do so. For a comprehensive list of things in CSS that can have color, see Mozilla's list of Things that can have color[18].

The opacity of an element is its level of opaqueness: an element with zero opacity is fully transparent. Everything that can be colored, can also have its opacity changed, and opacity can additionally affect images, which colors do not affect. Opacity and color go hand in hand, and there are several ways to apply color and opacity at the same time.

3.2. Color and Opacity Values

There are two main color models used to create colors in CSS. The first is the RGB color model, which combines **r**ed, **g**reen, and **b**lue in different amounts to create a wide range of colors. The second is the HSL color model, which creates colors based on **h**ue, **l**ightness, and **s**aturation. RGB is more popular and more commonly used.

18. https://developer.mozilla.org/en-US/docs/Web/HTML/Applying_color#Things_that_can_have_color

Picking RGB and HSL Colors

Guessing RGB and HSL colors by their notations, which are explained below, is extremely difficult, and thankfully, unnecessary. There are plenty of excellent tools for selecting RGB and HSL colors from color pallettes, such as Mozilla's Color picker tool[19].

Opacity is most commonly expressed as a percentage from `0%` (invisible) to `100%` (opaque), and/or a number from `0.0` (invisible) to `1.0` (opaque).

❖ 3.2.1. Color Keywords

CSS recognizes over a hundred color keywords as acceptable color values. Examples include standard colors you would expect, such as `blue`, `green`, `red`, `black`, `silver`, and `white`; and some more fun color names, such as `blanchedalmond`, `firebrick`, `floralwhite`, and `lawngreen`. For a full list of accepted color keywords see Mozilla's List of color keywords[20]

❖ 3.2.2. RGB Hexadecimal Notation

Evaluation Copy

RGB hexadecimal notation is the most commonly used type of color value. Hexadecimal digits are used to represent the amounts of each color component (red, green, and blue) ranging from 0 (`00`) to 255 (`ff`). The syntax for hexadecimal notation is below:

```
#rrggbb /* longhand */
#rgb /* shorthand */
```

Hexadecimal Numbers

The numbering system we are all used to is base 10. To make that work, we have ten single-digit characters: `0`, `1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, and `9`.

The hexidecimal system uses base 16. So, there are sixteen single-digit characters: `0`, `1`, `2`, `3`, `4`, `5`, `6`,`7`, `8`, `9`, `a`, `b`, `c`, `d`, `e`, and `f`.

In base 10, `90` is `10` greater than `80`. In base 16, `90` is `16` greater than `80`, and `a0` is `16` greater than `90`. You count it like this:

19. https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Colors/Color_picker_tool
20. https://developer.mozilla.org/en-US/docs/Web/CSS/color_value#Color_keywords

EVALUATION COPY: Not to be used in class.

```
 0   1   2   3   4   5   6   7   8   9   a   b   c   d   e   f
10  11  12  13  14  15  16  17  18  19  1a  1b  1c  1d  1e  1f
20  21  22  23  24  25  26  27  28  29  2a  2b  2c  2d  2e  2f
30  31  32  33  34  35  36  37  38  39  3a  3b  3c  3d  3e  3f
40  41  42  43  44  45  46  47  48  49  4a  4b  4c  4d  4e  4f
50  51  52  53  54  55  56  57  58  59  5a  5b  5c  5d  5e  5f
60  61  62  63  64  65  66  67  68  69  6a  6b  6c  6d  6e  6f
70  71  72  73  74  75  76  77  78  79  7a  7b  7c  7d  7e  7f
80  81  82  83  84  85  86  87  88  89  8a  8b  8c  8d  8e  8f
90  91  92  93  94  95  96  97  98  99  9a  9b  9c  9d  9e  9f
a0  a1  a2  a3  a4  a5  a6  a7  a8  a9  aa  ab  ac  ad  ae  af
b0  b1  b2  b3  b4  b5  b6  b7  b8  b9  ba  bb  bc  bd  be  bf
c0  c1  c2  c3  c4  c5  c6  c7  c8  c9  ca  cb  cc  cd  ce  cf
d0  d1  d2  d3  d4  d5  d6  d7  d8  d9  da  db  dc  dd  de  df
e0  e1  e2  e3  e4  e5  e6  e7  e8  e9  ea  eb  ec  ed  ee  ef
f0  f1  f2  f3  f4  f5  f6  f7  f8  f9  fa  fb  fc  fd  fe  ff
```

In shorthand hexadecimal notation, the six digits are cut down to three, and each digit represents two of itself. For example, `#f03` would be `#ff0033`. Both of those notations mean that the color is made from `ff` of red, `00` of green, and `33` of blue. The result is sort of a rose color.

Evaluation Copy

❖ 3.2.3. RGB Functional Notation

RGB colors can also be expressed in `rgb()` functional notation, which takes three parameters (`red`, `green`, and `blue`). These can either be integers from `0` to `255` or percentages from `0%` to `100%`. The syntax for RGB functional notation is:

```
rgb(red, green, blue)
```

❖ 3.2.4. HSL Functional Notation

HSL colors are expressed using the `hsl()` function notation. The `hsl()` function notation takes three parameters:

1. **Hue** - An angle of the color circle in degrees (`deg`), radians (`rad`), gradians (`grad`), or turns (`turn`). If units are omitted, `deg` are assumed.
    A. red = `0°` or `360°`
    B. green = `120°`

C. blue = `240°`

2. **Saturation** - A percentage where `100%` is completely saturated (full color) and `0%` is a shade of gray.
3. **Lightness** - A percentage where `100%` is white, `50%` is "normal" lightness, and `0%` is black.

The syntax for `hsl()` function notation is:

```
hsl(hue, saturation, lightness)
```

rgba(), hsla(), and #rgba

Both RGB and HSL functional notations have ways to specify opacity (alpha) at the same time as color. Along with their respective normal notations (`rgb()` and `hsl()`), they also have alpha notations (`rgba()` and `hsla()`) that accept a fourth parameter: `alpha`, which is expressed either as a percentage from `0%` to `100%` or a number between `0.0` and `1.0`. These are both well-supported in modern browsers.

Evaluation Copy

In modern browsers (i.e., not IE), the `rgb()` and `rgba()` notations are interchangeable, meaning that `rgb()` can also take the `alpha` parameter.

You can also specify opacity using hexadecimal notation by adding the hexadecimal value for opacity at the end. For example: `#ff0033aa`.

3.3. color

`color` is the CSS property used to apply color to the *foreground* of an element. The most common example of a "foreground" is just plain old text. See the samples below:

EVALUATION COPY: Not to be used in class.

Demo 3.1: ColorAndOpacity/Demos/color-styles.css

```
#keyword {
  color: red;
}

#rgb-longhand {
  color: #ff0000;
}

#rgb-shorthand {
  color: #f00;
}

#rgb-function-numbers {
  color: rgb(255, 0, 0);
}

#rgb-function-percentage {
  color: rgb(100%, 0%, 0%);
}

#rgba {
  color: rgba(255, 0, 0, 0.5);
}

#hsl-function {
  color: hsl(0, 100%, 50%);
}

#hsla {
  color: hsla(0, 100%, 50%, 50%);
}
```

Evaluation Copy

Demo 3.2: ColorAndOpacity/Demos/color-demo.html

```
      -------Lines 1 through 9 Omitted-------
10.   <body class="webucator">
11.     <h1>color</h1>
12.     <div id="keyword">color: red;</div>
13.     <div id="rgb-longhand">color: #ff0000;</div>
14.     <div id="rgb-shorthand">color: #f00;</div>
15.     <div id="rgb-function-numbers">color: rgb(255, 0, 0);</div>
16.     <div id="rgba">color: rgba(255, 0, 0, 0.5);</div>
17.     <div id="rgb-function-percentage">color: rgb(100%, 0%, 0%);</div>
18.     <div id="hsl-function">color: hsl(0, 100%, 50%);</div>
19.     <div id="hsla">color: hsla(0, 100%, 50%, 50%);</div>
20.   </body>
21.   </html>
```

The code above will render the following:[21]

21. If you are reading this in black and white, be sure to open the page in your browser.

EVALUATION COPY: Not to be used in class.

3.4. opacity

`opacity` is the CSS property used to set the transparency of an entire element. See the demo below:

Demo 3.3: ColorAndOpacity/Demos/opacity-styles.css

```
#my-div {
  /* To make the changes in opacity
  easier to see, we have added a yellow
  background color and made the text red */
  background-color: yellow;
  color: red;
}

.invisible {
  opacity: 0;
}

.light {
  opacity: .25;
}

.medium {
  opacity: .5;
}

.heavy {
  opacity: .75;
}

.opaque {
  opacity: 1;
}
```

Evaluation Copy

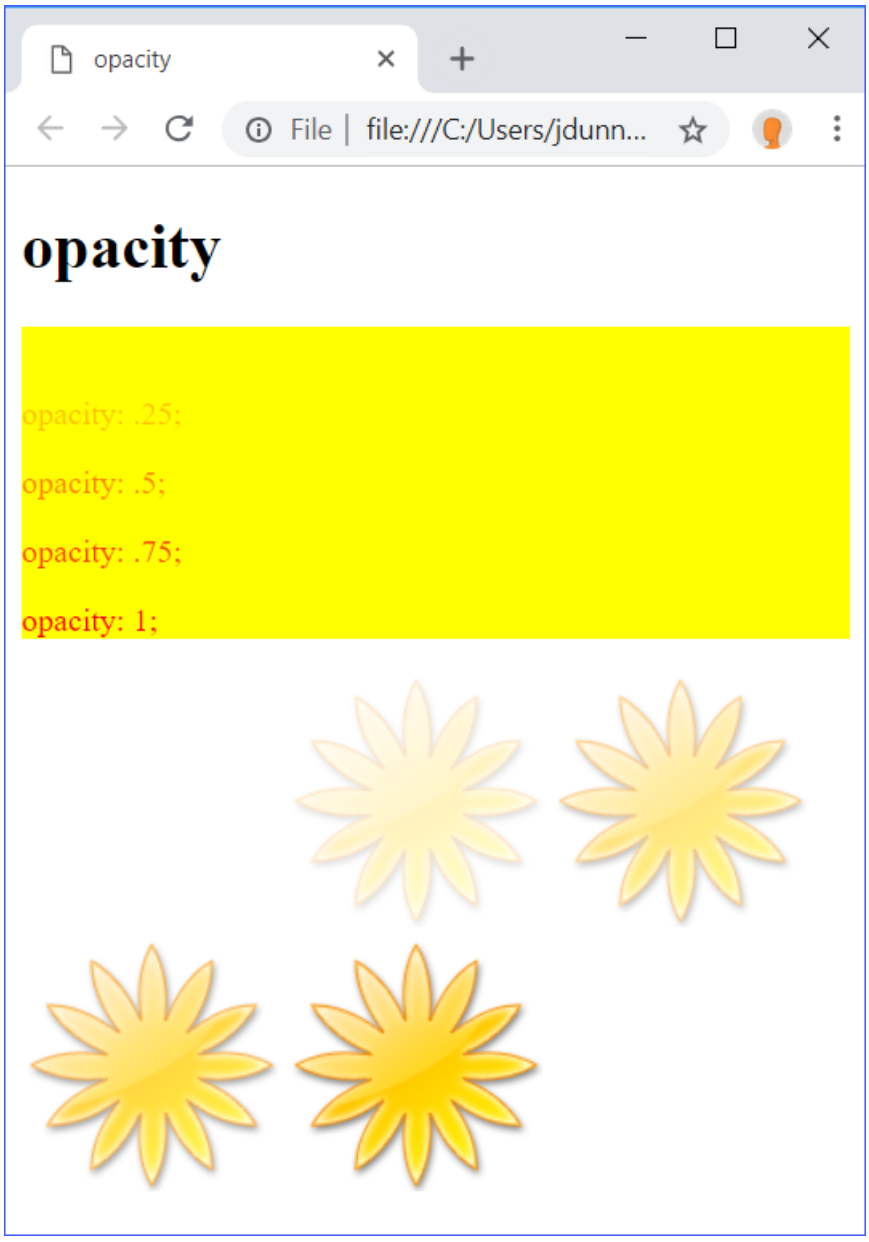
In this demo, `opacity` was also used to change the transparency of images. This is something that the `color` property cannot do as images are not affected by the `color` property.

Demo 3.4: ColorAndOpacity/Demos/opacity-demo.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.    <h1>opacity</h1>
12.    <div id="my-div">
13.      <p class="invisible">opacity: 0;</p>
14.      <p class="light">opacity: .25;</p>
15.      <p class="medium">opacity: .5;</p>
16.      <p class="heavy">opacity: .75;</p>
17.      <p class="opaque">opacity: 1;</p>
18.    </div>
19.    <img src="images/sun.png" class="invisible">
20.    <img src="images/sun.png" class="light">
21.    <img src="images/sun.png" class="medium">
22.    <img src="images/sun.png" class="heavy">
23.    <img src="images/sun.png" class="opaque">
24.  </body>
25.  </html>
```

The code above will render the following:

EVALUATION COPY: Not to be used in class.


opacity
File | file:///C:/Users/jdunn...
opacity
opacity: .25;
opacity: .5;
opacity: .75;
opacity: 1;

Exercise 6: Adding Color and Opacity to Text

25 to 40 minutes

In this exercise, you will add `color` and `opacity` properties to an HTML file containing three children's stories.

Evaluation Copy

1. Open `Exercises/stories.html` in your editor.
2. Create a CSS file called `stories-styles.css` and link to it from `stories.html`.
3. Using CSS, edit the color and opacity of the different elements in `stories.html`. The object of this exercise is to practice using the `color` and `opacity` properties.
4. When you are done, open `stories.html` in your browser. You are welcome to go back to the code and keep working.

You can design it however you like, or you can try to make it look something like: [22]

22. The images used in this section are in the public domain (`https://commons.wikimedia.org/wiki/Public_domain`):

    - `https://commons.wikimedia.org/wiki/File:Cinderella_and_the_Fairy_Godmother.jpg`
    - `https://commons.wikimedia.org/wiki/File:Boys_and_Girls_of_Bookland_Alice_in_Wonderland.jpg`
    - `https://commons.wikimedia.org/wiki/File:Joshua_Reynolds_-_Cupid_as_Link_Boy.jpg`

EVALUATION COPY: Not to be used in class.

Stories
'CSS-101_print/ClassFiles/ColorAndOpacity/Solutions/stories.html
Stories
A Mad Tea-Party
Cinderella
The Naughty Boy
A MAD TEA-PARTY - Lewis Carroll
There was a table set out under a tree in front of the house, and the March Hare and the Hatter were having tea at it: a Dormouse was sitting between them, fast asleep, and the other two were using it as a cushion, resting their elbows on it, and talking over its head. "Very uncomfortable for the Dormouse," thought Alice; "only, as it's asleep, I suppose it doesn't mind." Continue reading
Back to top

Solution: ColorAndOpacity/Solutions/stories-styles.css

```
a {
  color: red;
}

a[href='#top'],
h1 {
  color: limegreen;
}

nav a {
  color: hsl(300, 20%, 50%);
}

h2 {
  font-size: 1.2em;
  font-style: italic;
  font-weight: normal;
}

#cinderella h2 {
  color: hsl(56, 100%, 45%);
}

#alice h2 {
  color: #f60;
}

#naughtyboy h2 {
  color: #e96df2;
}

p {
  color: rgb(51, 102, 102);
  font-family: Cambria, Cochin, Georgia, Times, serif;
  font-size: .9em;
}

img {
  opacity: .5;
}
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

Conclusion

In this lesson, you have learned to work with color and opacity in CSS.

Evaluation Copy

EVALUATION COPY: Not to be used in class.

LESSON 4
CSS Text

Topics Covered

☑ `letter-spacing`

☑ `text-align`

☑ `text-decoration`

☑ `text-indent`

☑ `text-shadow`

☑ `text-transform`

☑ `white-space`

☑ `word-break`

☑ `word-spacing`

Evaluation Copy

Introduction

In this lesson, you will learn to use CSS properties for formatting text.

4.1. letter-spacing

The `letter-spacing` property is used to specify the amount of space between letters. The amount indicated is in addition to the default spacing. The property either takes the keyword `normal`, which is the default defined by the current font and/or browser, or a length value. See the following example:

Demo 4.1: CssText/Demos/letter-spacing-styles.css

```
1.   #normal {
2.     letter-spacing: normal;
3.   }
4.
5.   #wide {
6.     letter-spacing: 0.2rem;
7.   }
8.
9.   #wider {
10.    letter-spacing: 0.4rem;
11.  }
12.
13.  #widest {
14.    letter-spacing: 0.6rem;
15.  }
```

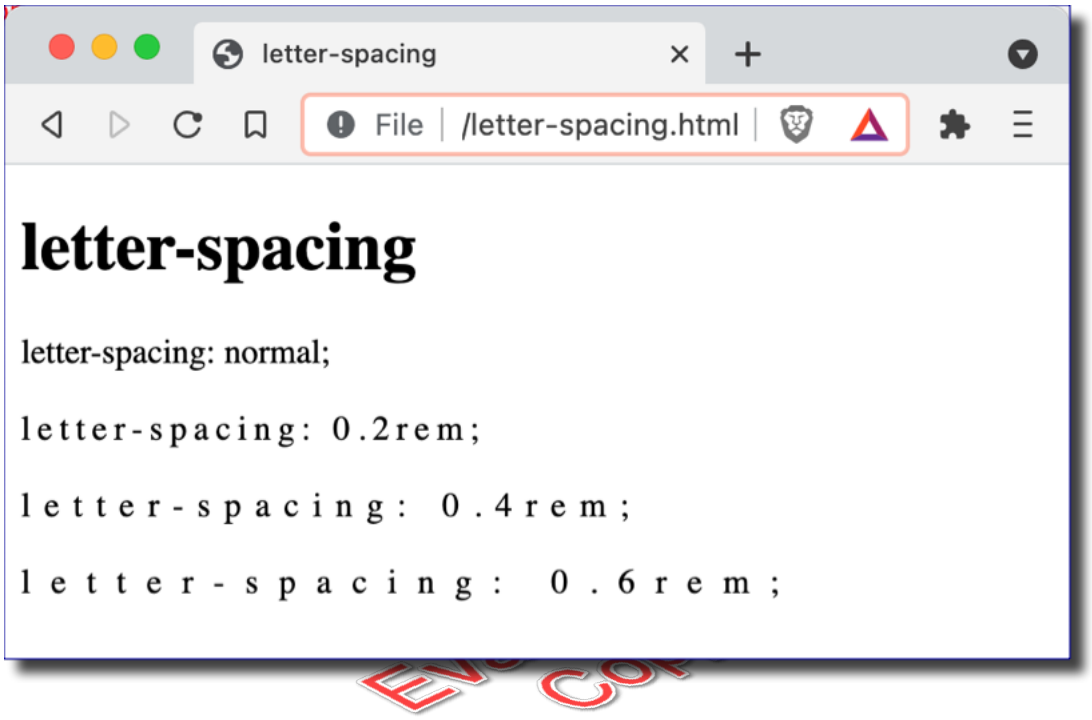
Evaluation Copy

Demo 4.2: CssText/Demos/letter-spacing.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.    <h1>letter-spacing</h1>
12.    <div id="normal">letter-spacing: normal;</div>
13.    <br>
14.    <div id="wide">letter-spacing: 0.2rem;</div>
15.    <br>
16.    <div id="wider">letter-spacing: 0.4rem;</div>
17.    <br>
18.    <div id="widest">letter-spacing: 0.6rem;</div>
19.  </body>
20.  </html>
```

The code above will render the following:

EVALUATION COPY: Not to be used in class.



4.2. text-align

The `text-align` property is used to specify how inline content should be aligned horizontally within a block. The values are listed below:

- `left`
- `right`
- `center`
- `justify`

Demo 4.3: CssText/Demos/text-align-styles.css

```
1.   .left {
2.     text-align: left;
3.   }
4.
5.   .center {
6.     text-align: center;
7.   }
8.
9.   .right {
10.    text-align: right;
11.  }
12.
13.  .justify {
14.    text-align: justify;
15.  }
```

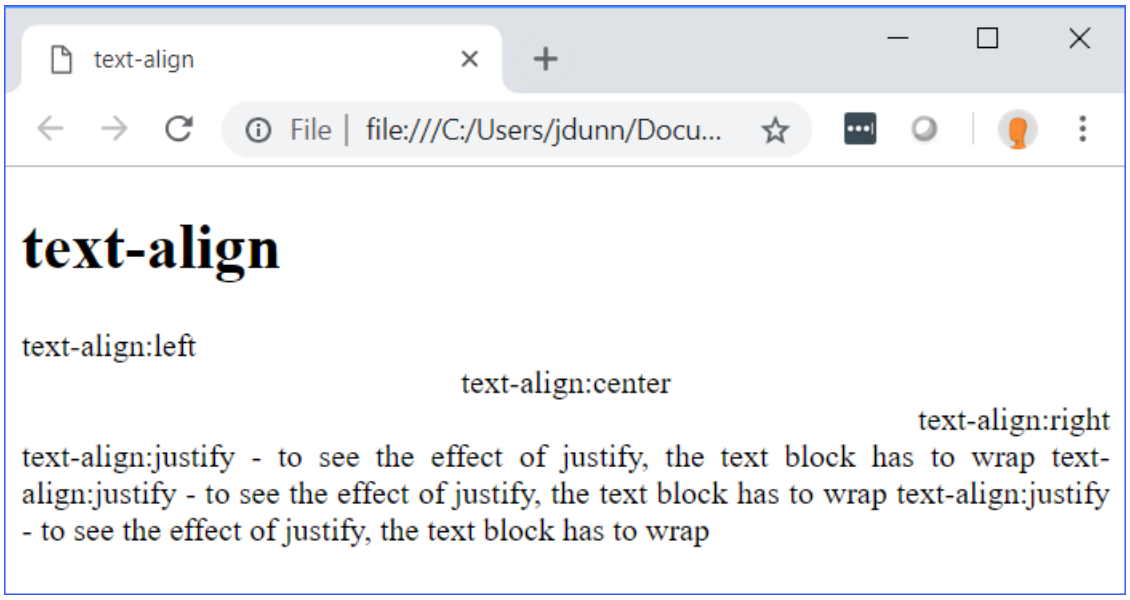
Demo 4.4: CssText/Demos/text-align.html

Evaluation Copy

```
     -------Lines 1 through 8 Omitted-------
9.   <body class="webucator">
10.  <h1>text-align</h1>
11.  <div class="left">text-align:left</div>
12.  <div class="center">text-align:center</div>
13.  <div class="right">text-align:right</div>
14.  <div class="justify">
15.    text-align:justify - to see the effect of justify,
16.      the text block has to wrap
17.    text-align:justify - to see the effect of justify,
18.      the text block has to wrap
19.    text-align:justify - to see the effect of justify,
20.      the text block has to wrap
21.  </div>
22.  </body>
23.  </html>
```

The code above will render the following:

EVALUATION COPY: Not to be used in class.



Evaluation Copy

4.3. text-decoration

The `text-decoration` property is used to add effects to text, such as underlines and line-throughs. The values are listed below:

- `none`
- `underline`
- `overline`
- `line-through`

The `none` value of the `text-decoration` property can be used to remove the underline from links, as shown below:

```
a {
  text-decoration: none;
}
```

Demo 4.5: CssText/Demos/text-decoration-styles.css

```
1.    .none {
2.      text-decoration: none;
3.    }
4.
5.    .overline {
6.      text-decoration: overline;
7.    }
8.
9.    .underline {
10.     text-decoration: underline;
11.   }
12.
13.   .line-through {
14.     text-decoration: line-through;
15.   }
```

Evaluation Copy

Demo 4.6: CssText/Demos/text-decoration.html

```
      -------Lines 1 through 8 Omitted-------
9.    <body class="webucator">
10.     <h1>text-decoration</h1>
11.     <div class="none">text-decoration: none</div>
12.     <div class="overline">text-decoration: overline</div>
13.     <div class="underline">text-decoration: underline</div>
14.     <div class="line-through">text-decoration: line-through</div>
15.     <div><a href="https://www.webucator.com" class="none">
16.       Webucator</a>
17.     </div>
18.   </body>
19.   </html>
```

The code above will render the following:

EVALUATION COPY: Not to be used in class.

Note how we have removed the underline from the link using `text-decoration: none`.

`text-decoration` is actually a shorthand property for the following longhand properties:

1. `text-decoration-line`: Values include `none`, `underline`, `overline`, and `line-through`.
2. `text-decoration-color`: Any valid color.
3. `text-decoration-style`: Values include `solid`, `double`, `dotted`, `dashed`, and `wavy`.
4. `text-decoration-thickness`: Values can be a length (e.g., `3px`), a percentage (e.g, `10%`), or a keyword (e.g., `auto` or `from-font`).

Evaluation Copy

For more details on `text-decoration`, see `https://developer.mozilla.org/en-US/docs/Web/CSS/text-decoration`.

4.4. text-indent

The `text-indent` property is used to indent (or outdent) the first line of a block of text. The value can be specified in number of units or in percentage of the width of the containing block.

The following code sample shows the effects of `text-indent`:

Demo 4.7: CssText/Demos/text-indent-styles.css

```
1.    .indent-length {
2.      text-indent: 3em;
3.    }
4.
5.    .indent-percentage {
6.      text-indent: 10%;
7.    }
8.
9.    .outdent-length {
10.     text-indent: -3em;
11.   }
12.
13.   .outdent-percentage {
14.     text-indent: -10%;
15.   }
```

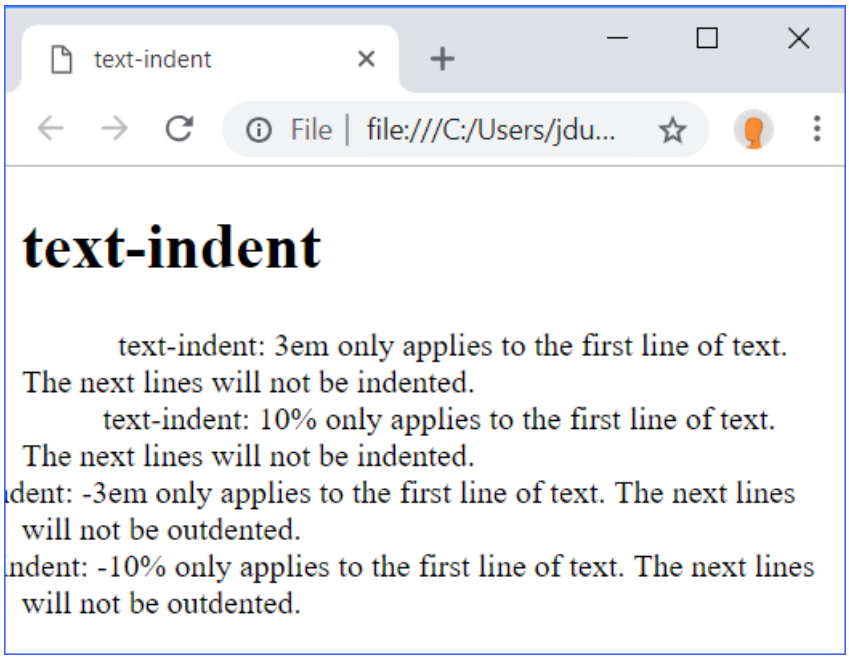
Demo 4.8: CssText/Demos/text-indent.html

```
      -------Lines 1 through 8 Omitted-------
9.    <body class="webucator">
10.     <h1>text-indent</h1>
11.     <div class="indent-length">
12.       text-indent: 3em only applies to the first line of text.
13.       The next lines will not be indented.
14.     </div>
15.     <div class="indent-percentage">
16.       text-indent: 10% only applies to the first line of text.
17.       The next lines will not be indented.
18.     </div>
19.     <div class="outdent-length">
20.       text-indent: -3em only applies to the first line of text.
21.       The next lines will not be outdented.
22.     </div>
23.     <div class="outdent-percentage">
24.       text-indent: -10% only applies to the first line of text.
25.       The next lines will not be outdented.
26.     </div>
27.   </body>
28.   </html>
```

Evaluation Copy

The code above will render the following:

EVALUATION COPY: Not to be used in class.



Note how the outdenting moves some of the content out of the viewport. Normally, you would add padding on the left to prevent this from happening. We will learn how to do that in a later lesson.

Evaluation Copy

4.5. text-shadow

The `text-shadow` property is used to add shadow to text. It can take a comma-delimited list of shadows, each described by a combination of x and y offsets, blur radius, and color. The syntax is shown below:

```
text-shadow: x-offset y-offset blur-radius color /* shadow1 */,
             x-offset y-offset blur-radius color /* shadow2 */,
             x-offset y-offset blur-radius color /* shadow3 */
```

The values are explained below:

1. `x-offset` and `y-offset` – Required length values. `x-offset` must come immediately before `y-offset`. `x-offset` specifies the horizontal displacement (positive values to the right and negative values to the left). `y-offset` specifies the vertical displacement (positive values below and negative values above). If both values are zero, the shadow will be directly behind the text.

2. `blur-radius` – An optional length value that specifies the size of the blur effect. If included, this value must come after the two offset values.

3. `color` – An optional color value that has to be either the first value included or the last value. If not included, the browser picks the color, so for consistency across browsers, it is a good idea to specify the shadow's color.

Demo 4.9: CssText/Demos/text-shadow-styles.css

```
1.   #simple-shadow {
2.     color: yellow;
3.     text-shadow: -.1rem -.1rem blue;
4.   }
5.
6.   #blurred-shadow {
7.     text-shadow: .1rem .1rem .25rem red;
8.   }
9.
10.  #double-shadow {
11.    color: red;
12.    text-shadow: .1rem .1rem white,
13.      .2rem .2rem blue;
14.  }
```

Evaluation Copy

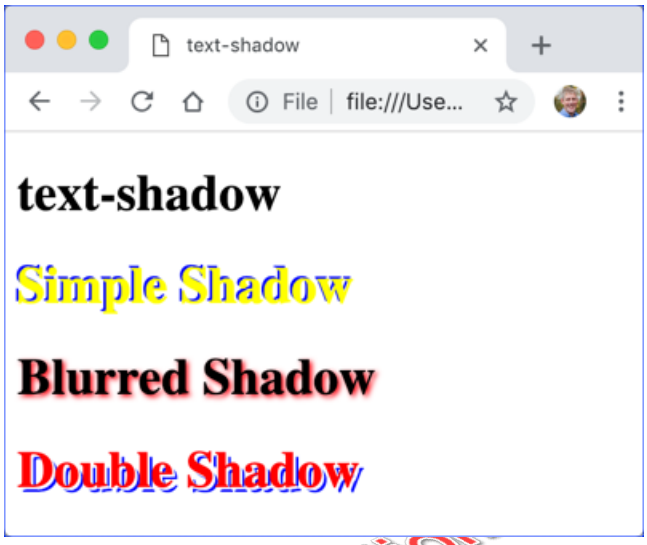
Demo 4.10: CssText/Demos/text-shadow.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.    <h1>text-shadow</h1>
12.    <h1 id="simple-shadow">Simple Shadow</h1>
13.    <h1 id="blurred-shadow">Blurred Shadow</h1>
14.    <h1 id="double-shadow">Double Shadow</h1>
15.  </body>
16.  </html>
```

The code above will render the following:[23]

23. If you are reading this in black and white, be sure to open the page in your browser to see the color effects.

EVALUATION COPY: Not to be used in class.



Evaluation Copy

4.6. text-transform

The `text-transform` property is used to change the capitalization of text. The most common values are listed below:

- `none`
- `capitalize`
- `uppercase`
- `lowercase`

The following code sample shows the effects of `text-transform`:

Demo 4.11: CssText/Demos/text-transform-styles.css

```
1.   #none {
2.     text-transform: none;
3.   }
4.
5.   #caps {
6.     text-transform: capitalize;
7.   }
8.
9.   #lower {
10.    text-transform: lowercase;
11.  }
12.
13.  #upper {
14.    text-transform: uppercase;
15.  }
```

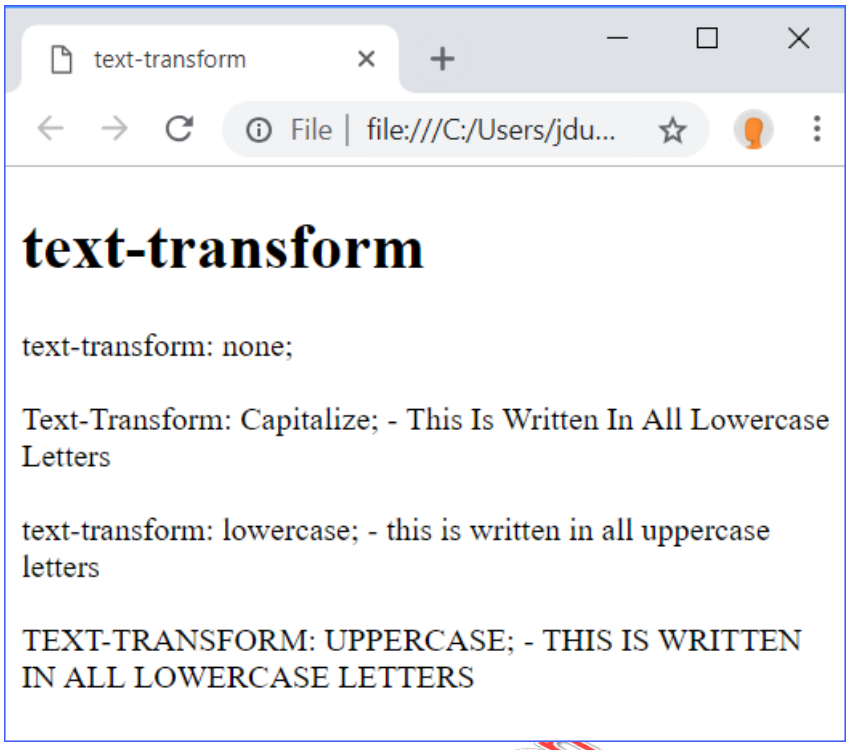
Demo 4.12: CssText/Demos/text-transform.html

Evaluation Copy

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.    <h1>text-transform</h1>
12.    <div id="none">text-transform: none;</div><br>
13.    <div id="caps">text-transform: capitalize;
14.      - this is written in all lowercase letters
15.    </div><br>
16.    <div id="lower">text-transform: lowercase;
17.      - THIS IS WRITTEN IN ALL UPPERCASE LETTERS
18.    </div><br>
19.    <div id="upper">text-transform: uppercase;
20.      - this is written in all lowercase letters
21.    </div>
22.  </body>
23.  </html>
```

The code above will render the following:

EVALUATION COPY: Not to be used in class.



Notice in the last two examples that the lowercase letters have been transformed to uppercase, and the uppercase letters have been transformed to lowercase.

Evaluation Copy

4.7. white-space

The `white-space` property determines how sequences of whitespace are displayed. Below we list the most common values of the `white-space` property and their effects:

1. `normal`
    - Collapses adjacent spaces and tabs.
    - Collapses line breaks.
    - Wraps to fit containing box.
2. `nowrap`
    - Collapses adjacent spaces and tabs.
    - Collapses line breaks.
    - Does not wrap to fit containing box.

3. `pre`
    - Does not collapse adjacent spaces and tabs.
    - Does not collapse line breaks.
    - Does not wrap to fit containing box.
4. `pre-line`
    - Collapses adjacent spaces and tabs.
    - Does not collapse line breaks.
    - Wraps to fit containing box.
5. `pre-wrap`
    - Does not collapse adjacent spaces and tabs.
    - Does not collapse line breaks.
    - Wraps to fit containing box.

The following code sample shows the effects of `white-space`:

Evaluation Copy

Demo 4.13: CssText/Demos/white-space-styles.css

```
#normal {
  white-space: normal;
}

#nowrap {
  white-space: nowrap;
}
#pre {
  white-space: pre;
}

#pre-line {
  white-space: pre-line;
}

#pre-wrap {
  white-space: pre-wrap;
}
```

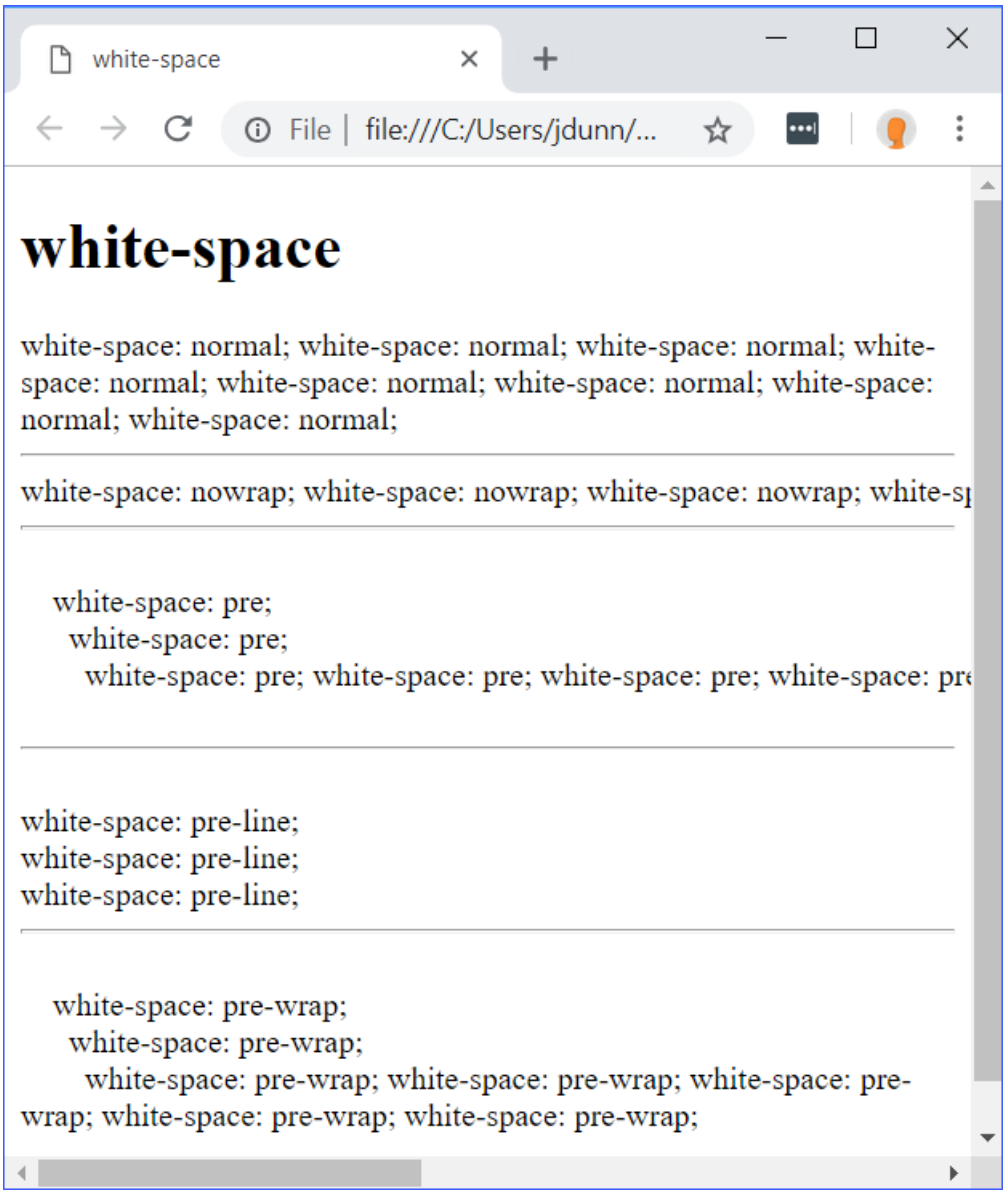

EVALUATION COPY: Not to be used in class.

Demo 4.14: CssText/Demos/white-space.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.    <h1>white-space</h1>
12.    <div id="normal">
13.      white-space: normal;
14.      white-space: normal;
15.      white-space: normal; white-space: normal; white-space: normal; white-space:
               normal;
16.      white-space: normal; white-space: normal;
17.    </div>
18.    <hr>
19.    <div id="nowrap">
20.      white-space: nowrap;
21.      white-space: nowrap;
22.      white-space: nowrap; white-space: nowrap; white-space: nowrap; white-space:
               nowrap;
23.      white-space: nowrap; white-space: nowrap;
24.    </div>
25.    <hr>
26.    <div id="pre">
27.      white-space: pre;
28.        white-space: pre;
29.          white-space: pre; white-space: pre; white-space: pre; white-space: pre;
               white-space: pre;
30.    </div>
31.    <hr>
32.    <div id="pre-line">
33.      white-space: pre-line;
34.        white-space: pre-line;
35.          white-space: pre-line;
36.    </div>
37.    <hr>
38.    <div id="pre-wrap">
39.      white-space: pre-wrap;
40.        white-space: pre-wrap;
41.          white-space: pre-wrap; white-space: pre-wrap; white-space: pre-wrap;
                white-space: pre-wrap; white-space: pre-wrap;
42.    </div>
43.  </body>
44.  </html>
```

Evaluation Copy

The code above will render the following:



Notice that the user would need to scroll right to see the content that we have forced not to wrap.

4.8. word-break

The `word-break` property specifies where it is permissible to have line breaks when text would otherwise overflow its content box. The most common values are explained below:

EVALUATION COPY: Not to be used in class.

1. `normal` -- default line breaks (at the end of words or at hyphens)
2. `break-all` -- line breaks can happen between any two characters

The following example shows the effects of `word-break`:

Demo 4.15: CssText/Demos/word-break-styles.css

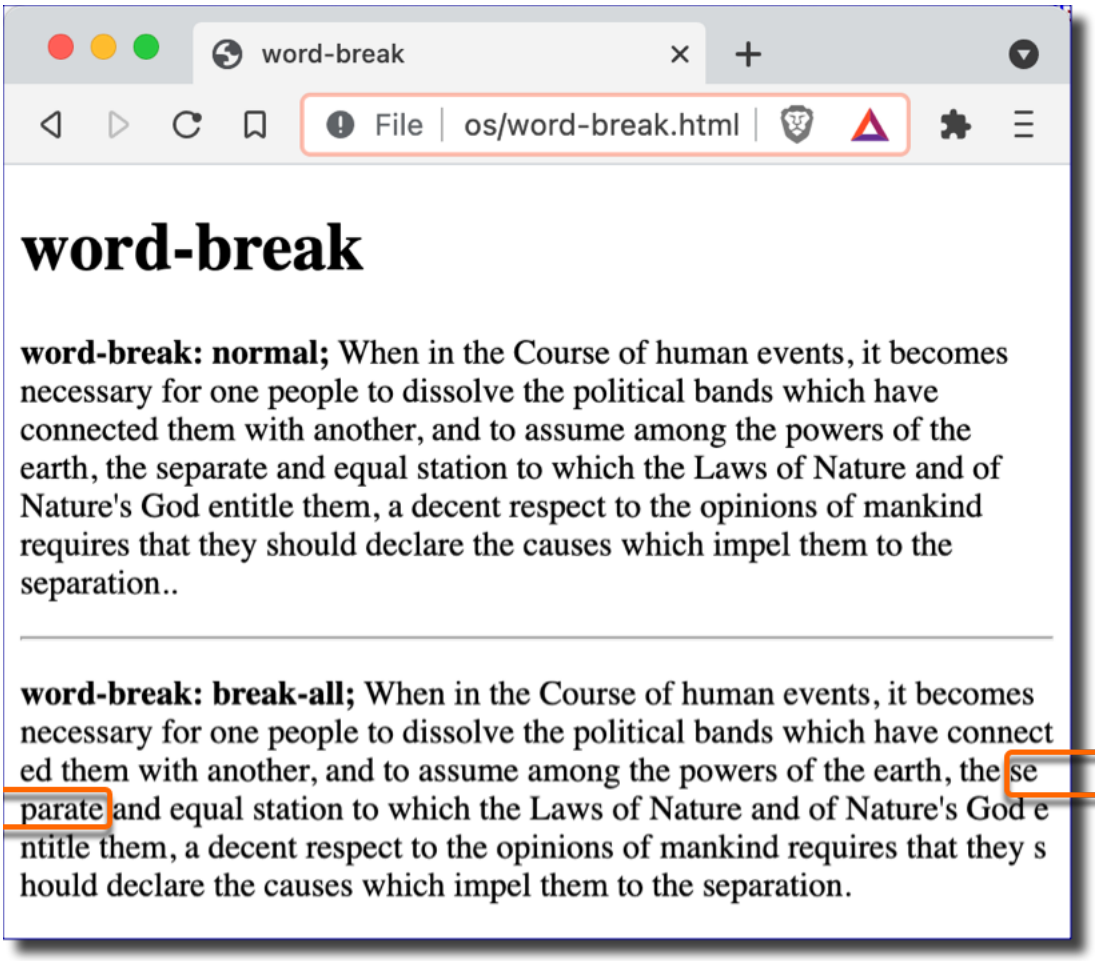
```
1.   #div1 {
2.     word-break: normal;
3.   }
4.
5.   #div2 {
6.     word-break: break-all;
7.   }
```

Demo 4.16: CssText/Demos/word-break.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.    <h1>word-break</h1>
12.    <div id="div1">
13.      <p><strong>word-break: normal;</strong> When in the Course of human
14.      events, it becomes necessary for one people to dissolve the
15.      political bands which have connected them with another, and to
16.      assume among the powers of the earth, the separate and equal station
17.      to which the Laws of Nature and of Nature's God entitle them,  a
18.      decent respect to the opinions of mankind requires that they should
19.      declare the causes which impel them to the separation..</p>
20.    </div>
21.    <hr>
22.    <div id="div2">
23.      <p><strong>word-break: break-all;</strong> When in the Course of
24.      human events, it becomes necessary for one people to dissolve the
25.      political bands which have connected them with another, and to
26.      assume among the powers of the earth, the separate and equal station
27.      to which the Laws of Nature and of Nature's God entitle them,  a
28.      decent respect to the opinions of mankind requires that they should
29.      declare the causes which impel them to the separation.</p>
30.    </div>
31.  </body>
32.  </html>
```

Evaluation Copy

The code above will render the following:



Notice that in the second rendering of the paragraph, some of the wrapping splits in the middle of a word (e.g., "separate").

4.9. word-spacing

The `word-spacing` property is used to specify the amount of space between words. The amount indicated is in addition to the default spacing. The property either takes the keyword `normal`, which is the default defined by the current font and/or browser, or a length value. See the following example:

EVALUATION COPY: Not to be used in class.

Demo 4.17: CssText/Demos/word-spacing-styles.css

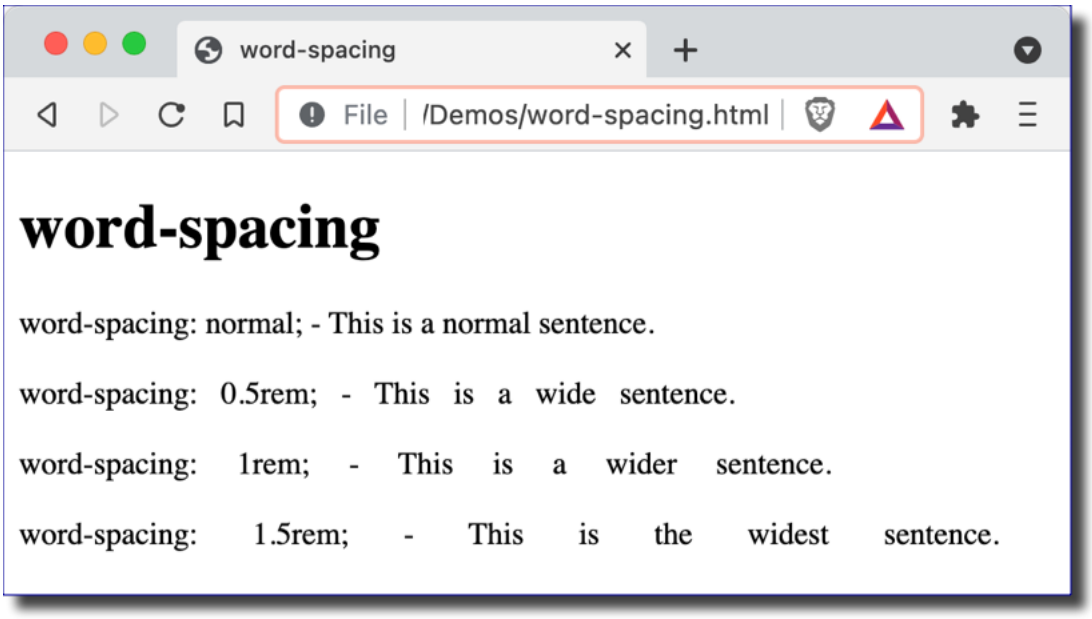
```
1.   #normal {
2.     word-spacing: normal;
3.   }
4.
5.   #wide {
6.     word-spacing: 0.5rem;
7.   }
8.
9.   #wider {
10.    word-spacing: 1rem;
11.  }
12.
13.  #widest {
14.    word-spacing: 1.5rem;
15.  }
```

Demo 4.18: CssText/Demos/word-spacing.html

Evaluation Copy

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.    <h1>word-spacing</h1>
12.    <div id="normal">word-spacing: normal;
13.      - This is a normal sentence.
14.    </div><br>
15.    <div id="wide">word-spacing: 0.5rem;
16.      - This is a wide sentence.
17.    </div><br>
18.    <div id="wider">word-spacing: 1rem;
19.      - This is a wider sentence.
20.    </div><br>
21.    <div id="widest">word-spacing: 1.5rem;
22.      - This is the widest sentence.
23.    </div>
24.  </body>
25.  </html>
```

The code above will render the following:



EVALUATION COPY: Not to be used in class.

Exercise 7: Text Properties

25 to 40 minutes

In this exercise, you will continue to work on the Runners Home page that you started in the Fonts lesson (see page 82).

1. Open `CssText/Exercises/index.html` in your editor. You will see the Runners Home home page.
2. Open `CssText/Exercises/styles.css`. This is the stylesheet from the solution to the Fonts exercise. Feel free to continue working from your own stylesheet.
3. Using your new knowledge of text properties, add some styles to this page. The object of this exercise is to practice using the text properties covered in this lesson. Feel free to use `color` and `opacity` and any other properties you have learned as well.
4. When you are done, open `index.html` in your browser to see the results. You are welcome to go back to the code and continue to work.

Evaluation Copy

You can design it however you like, or you can try to make it look something like:

- Home
- Races
- Resources
- Calculator
- Running Log
- My Account
- Log out

Hello, Stranger!

Welcome to Runners Home™

Runners Home™ is dedicated to providing you with:

1. the most up-to-date information on running races.
2. the best resources for runners.

Best Running Tips

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam et gravida sapien, facilisis condimentum arcu. Morbi eget dui iaculis, porttitor eros et, tincidunt erat. Sed sollicitudin, mauris vitae semper consequat, purus velit aliquet mi, sit amet posuere eros tortor eget neque. Phasellus hendrerit justo... Click here to read more!

Best Health Tips

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam et gravida sapien, facilisis condimentum arcu. Morbi eget dui iaculis, porttitor eros et, tincidunt erat. Sed sollicitudin, mauris vitae semper consequat, purus velit aliquet mi, sit amet posuere eros tortor eget neque. Phasellus hendrerit justo... Click here to read more!

Newsletter

Email: Sign Up

Be the first to hear about our great offers.
Sign up for our newsletter today!

© 2022 Runners Home. All rights reserved. For questions, send email to info@runners-home.com.

EVALUATION COPY: Not to be used in class.

Solution: CssText/Solutions/styles.css

```
1.  html {
2.    font-size: 16px;
3.  }
4.
5.  body {
6.    font: 1rem/1.3 Verdana, Geneva, Tahoma, sans-serif;
7.  }
8.
9.  nav a {
10.   text-decoration: none;
11. }
12.
13. #welcome {
14.   font-size: larger;
15. }
16.
17. #welcome h1 {
18.   color: rgb(8, 5, 211);
19.   text-shadow: 0.2rem 0.2rem 0.1rem rgb(211, 200, 238);
20. }
21.
22. article {
23.   text-align: center;
24. }
25.
26. article h2 {
27.   font-variant: small-caps;
28.   letter-spacing: 0.1em;
29.   word-spacing: 0.2em;
30. }
31.
32. article p {
33.   line-height: 2;
34.   text-align: left;
35.   text-indent: 3rem;
36. }
37.
38. .read-more {
39.   text-decoration: none;
40. }
41.
42. aside label {
43.   font-weight: bold;
44. }
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
45.
46.   #newsletter {
47.     text-align: right;
48.   }
49.
50.   footer {
51.     font-size: smaller;
52.     font-style: italic;
53.   }
```

Evaluation Copy

Conclusion

In this lesson, you have learned to use CSS text properties.

EVALUATION COPY: Not to be used in class.

LESSON 5

Borders, Margins, and Padding

Topics Covered

☑ The CSS basic box model.

☑ `padding`

☑ `margin`

☑ `border` properties.

☑ `box-sizing`

☑ `box-shadow`

Evaluation Copy

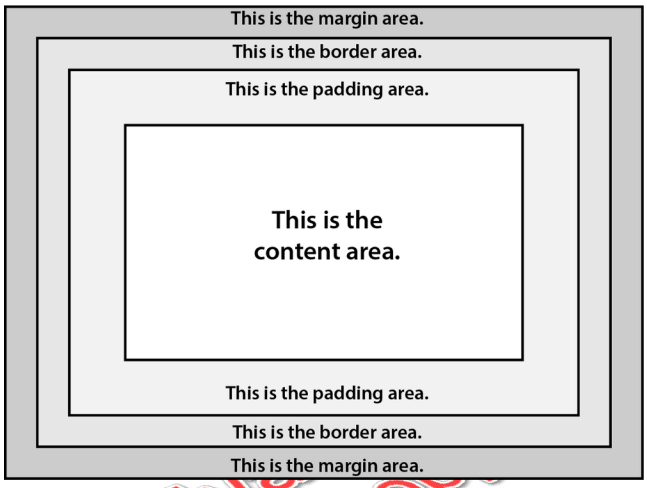
Introduction

In this lesson, you will learn about the CSS box model and how it relates to borders, margin, and padding.

5.1. The CSS Basic Box Model

Every element in CSS can be thought of as a box made up of four parts, or areas:

1. The **content area** contains the actual content of the element box (e.g., text, an image, a video, etc.).
2. The **padding area** extends the content area to include padding inside the element box.
3. The **border area** extends the padding area to include a border around the element box.
4. The **margin area** extends the border area to include a margin outside the element box.



Throughout this lesson, we will be learning how to size and style these areas.

❖ 5.1.1. Introduction to using Google Chrome DevTools with CSS

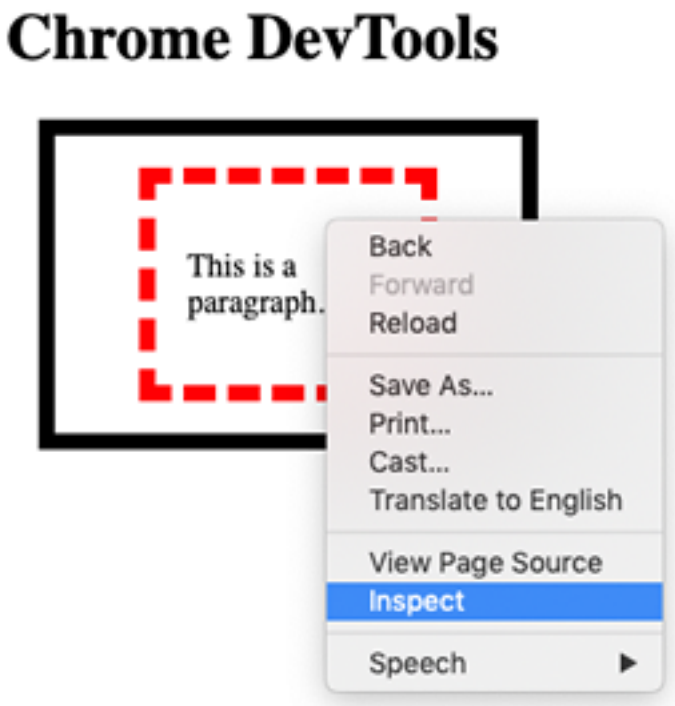
Most modern browsers provide built-in development tools. We will be looking at how the **Elements** panel in Google Chrome DevTools[24] can be used with CSS.

The **Elements** panel of Chrome DevTools can be opened using the following methods:

1. Right-click the page and select **Inspect**:

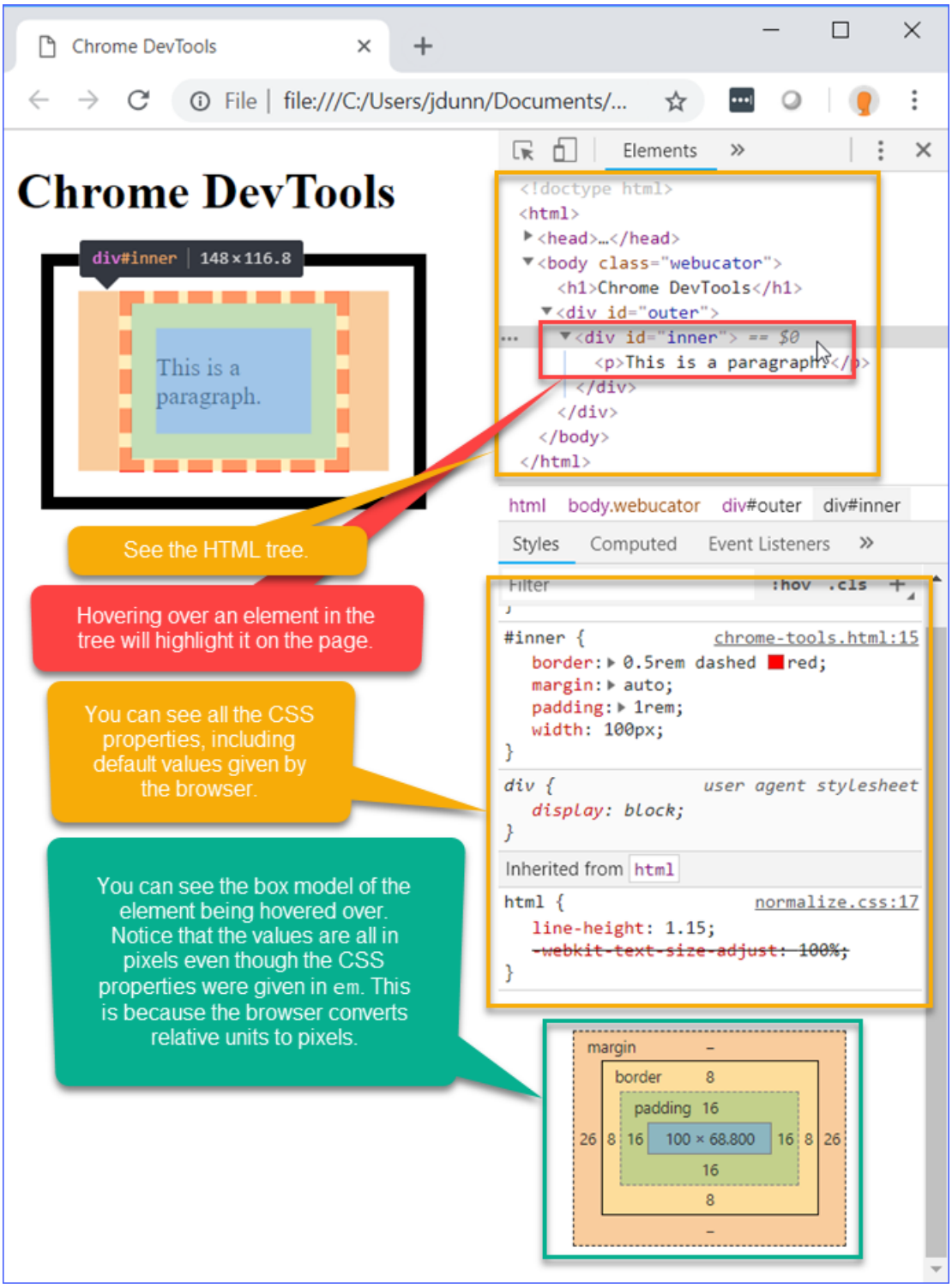
24. https://developers.google.com/web/tools/chrome-devtools/css/

EVALUATION COPY: Not to be used in class.



2. **Control + Shift + C** for Windows, or
3. **Command + Option + C** for Mac.

See the image below of DevTools in action:



This file is `BordersMarginAndPadding/Demos/dev-tools.html`.

EVALUATION COPY: Not to be used in class.

Some things to note:

1. DevTools show the entire HTML tree.
2. Going through this tree and selecting the different elements will show the CSS rules that affect those elements, including rules that are default values coming from the browser.
3. DevTools will show the calculated values for `width`, `height`, `margin`, `border`, and `padding` in the element's box model diagram.
4. DevTools allows you to edit CSS properties in the browser by checking and unchecking existing properties (see the unchecked `border` property) or by creating new rules and properties (see the `#inner p` rule. These are created with the `+` sign in the upper right corner).

5.2. Padding

The `padding` shorthand property is used to size an element's padding area. `padding` takes either length values specified in units (e.g., `em`, `rem`, `px`, etc.), or percentage values relative to the width of the containing element. All values must be positive.

1. When one value is specified, it applies to all four sides of the element.
2. When two values are specified, the first value specifies the padding for the top and bottom and the second value specifies the padding for the left and right.
3. When three values are specified, the first value is for the top, the second for the left and right, and the third for the bottom.

4. When four values are specified, the values specify padding in a clockwise direction starting with the top (top, right, bottom, left).

5. Alternatively, padding can be applied with the `padding-top`, `padding-right`, `padding-bottom`, and `padding-left` properties.

Demo 5.1: BordersMarginsAndPadding/Demos/padding-styles.css

```
1.   .border {
2.     border: 1px solid black;
3.   }
4.
5.   #div1 {
6.     padding: 1em;
7.   }
8.
9.   #div2 {
10.    padding: 1em 2em;
11.  }
12.
13.  #div3 {
14.    padding: 1em 2em 3em;
15.  }
16.
17.  #div4 {
18.    padding: 1em 2em 3em 4em;
19.  }
20.
21.  #div5 {
22.    padding-top: 1em;
23.    padding-right: 2em;
24.    padding-bottom: 3em;
25.    padding-left: 4em;
26.  }
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

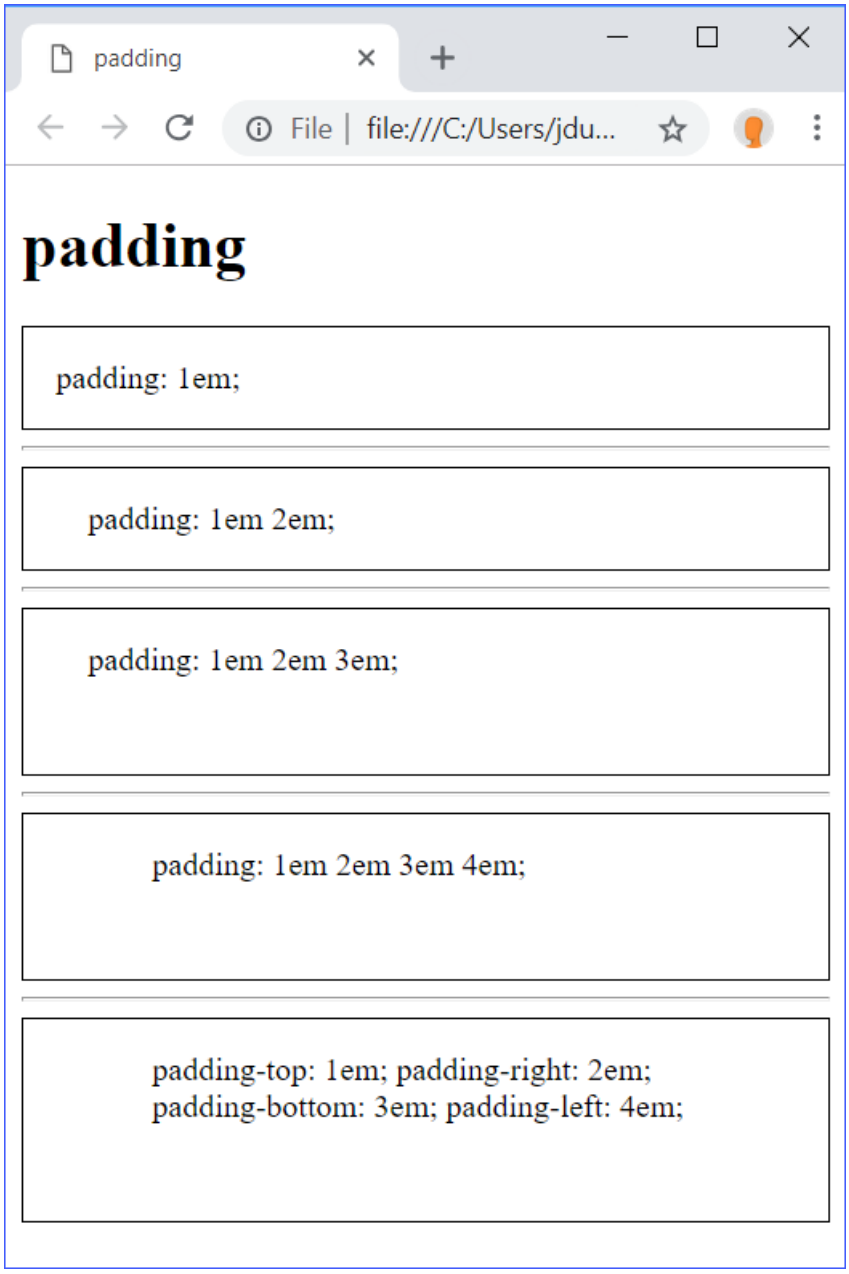
Demo 5.2: BordersMarginsAndPadding/Demos/padding.html

```
     -------Lines 1 through 8 Omitted-------
9.   <body class="webucator">
10.    <h1>padding</h1>
11.    <div id="div1" class="border">
12.      padding: 1em;
13.    </div>
14.    <hr>
15.    <div id="div2" class="border">
16.      padding: 1em 2em;
17.    </div>
18.    <hr>
19.    <div id="div3" class="border">
20.      padding: 1em 2em 3em;
21.    </div>
22.    <hr>
23.    <div id="div4" class="border">
24.      padding: 1em 2em 3em 4em;
25.    </div>
26.    <hr>
27.    <div id="div5" class="border">
28.      padding-top: 1em;
29.      padding-right: 2em;
30.      padding-bottom: 3em;
31.      padding-left: 4em;
32.    </div>
33.  </body>
34.  </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:



EVALUATION COPY: Not to be used in class.

Note

We have included a `border` class to add a simple 1px black border around each `div` block so that the padding is visible. We will cover borders soon.

5.3. Margin

The `margin` shorthand property is used to size an element's margin area. Just like `padding`, `margin` takes either length values specified in units (e.g., `em`, `rem`, `px`, etc.), or percentage values relative to the width of the containing element. Negative values bring an element closer to its neighbors and can cause elements to overlap. The same shorthand specification rules that apply to `padding` also apply to `margin`:

Evaluation Copy

1. When one value is specified, it applies the same margin to all four sides of the element.
2. When two values are specified, the first value specifies the margin for the top and bottom and the second value specifies the margin for the left and right.
3. When three values are specified, the first value is for the top, the second for the left and right, and the third for the bottom.
4. When four values are specified, the values specify margin in a clockwise direction starting with the top (top, right, bottom, left).
5. Alternatively, margin can be applied with the `margin-top`, `margin-right`, `margin-bottom`, and `margin-left` properties.

Demo 5.3: BordersMarginsAndPadding/Demos/margin-styles.css

```
1.    .border {
2.      border: 1px solid black;
3.    }
4.
5.    #div1 {
6.      margin: 1em;
7.    }
8.
9.    #div2 {
10.     margin: 1em 2em;
11.   }
12.
13.   #div3 {
14.     margin: 1em 2em 3em;
15.   }
16.
17.   #div4 {
18.     margin: 1em 2em 3em 4em;
19.   }
20.
21.   #div5 {
22.     margin-top: 1em;
23.     margin-right: 2em;
24.     margin-bottom: 3em;
25.     margin-left: 4em;
26.   }
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

Demo 5.4: BordersMarginsAndPadding/Demos/margin.html

```
     -------Lines 1 through 8 Omitted-------
9.    <body class="webucator">
10.     <h1>margin</h1>
11.     <div id="container" class="border">
12.       <div id="div1" class="border">
13.         margin: 1em;
14.       </div>
15.       <hr>
16.       <div id="div2" class="border">
17.         margin: 1em 2em;
18.       </div>
19.       <hr>
20.       <div id="div3" class="border">
21.         margin: 1em 2em 3em;
22.       </div>
23.       <hr>
24.       <div id="div4" class="border">
25.         margin: 1em 2em 3em 4em;
26.       </div>
27.       <hr>
28.       <div id="div5" class="border">
29.         margin-top: 1em;
30.         margin-right: 2em;
31.         margin-bottom: 3em;
32.         margin-left: 4em;
33.       </div>
34.     </div>
35.   </body>
36.   </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:

5.4. Border

The `border` shorthand property is used to size and add basic styles to all four sides of an element's border area. `border` can take up to three of the values listed below in any order:

- `border-width` - Specifies the thickness of the border in units (`em`, `rem`, `px`, etc.) or as a keywords (`thin`, `medium`, `thick`). The default is `medium`.

EVALUATION COPY: Not to be used in class.

- `border-style` - Specifies the line style of the border as a keyword (`none`, `solid`, `dotted`, `double`, etc.). The default is `none`. For a full list of line style keywords, see Mozilla's `border-style` values list[25].
- `border-color` - Specifies the color of the border as a color value. The default is the current value specified by the `color` property.

Shorthand Defaults

As with any shorthand property, any unspecified values will be set to their default values.

Evaluation Copy

The outline Property

The `outline`[26] shorthand property works exactly like the `border` shorthand property. However, while a border takes up space and is part of the box model, an outline does not take up space and is not part of the box model.

25. https://developer.mozilla.org/en-US/docs/Web/CSS/border-style#Values
26. https://developer.mozilla.org/en-US/docs/Web/CSS/outline

Demo 5.5: BordersMarginsAndPadding/Demos/border-styles.css

```
1.   .padding {
2.     padding: 1rem;
3.   }
4.
5.   #div1 {
6.     border: 1px solid red;
7.   }
8.
9.   #div2 {
10.    border: green dotted;
11.  }
12.
13.  #div3 {
14.    border: 0.3rem blue;
15.  }
16.
17.  #div4 {
18.    border-top: 1px solid red;
19.    border-right: green dotted;
20.    border-bottom: 0.3rem dashed purple;
21.    border-left: 1rem blue double;
22.  }
```

Evaluation Copy

Note

We have included a `padding` class to add a little padding in each `div` block to push the border away from the content.

EVALUATION COPY: Not to be used in class.

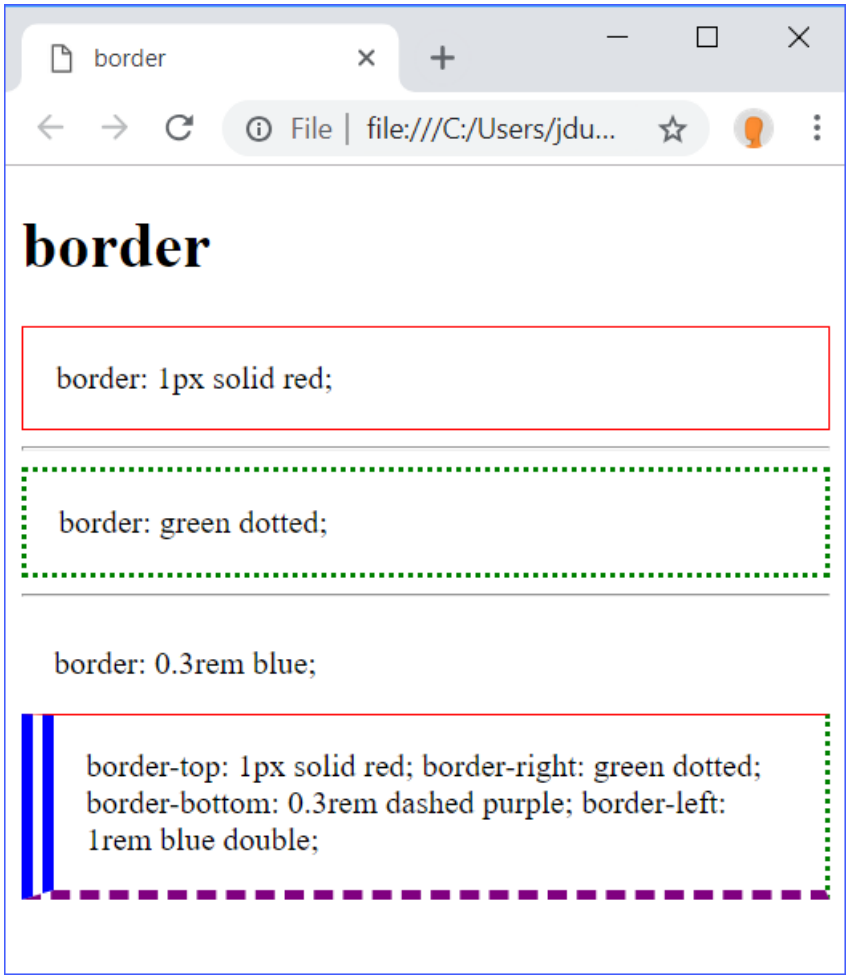
Demo 5.6: BordersMarginsAndPadding/Demos/border.html

```
     -------Lines 1 through 8 Omitted-------
9.   <body class="webucator">
10.    <h1>border</h1>
11.    <div id="div1" class="padding">
12.      border: 1px solid red;
13.    </div>
14.    <hr>
15.    <div id="div2" class="padding">
16.      border: green dotted;
17.    </div>
18.    <hr>
19.    <div id="div3" class="padding">
20.      border: 0.3rem blue;
21.    </div>
22.    <div id="div4" class="padding">
23.      border-top: 1px solid red;
24.      border-right: green dotted;
25.      border-bottom: 0.3rem dashed purple;
26.      border-left: 1rem blue double;
27.  </body>
28.  </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:



The `div` with the `border: 0.3rem blue` style does not render because no `border-style` was specified and the default `border-style` is `none`.

❖ 5.4.1. border-width

The `border-width` shorthand property is used to define the widths of the top, right, bottom, and left borders of an element. It takes either a border thickness in units (`em`, `rem`, `px`, etc.) or a keyword (`thin`, `medium`, `thick`). You specify individual border widths in the same way we have seen with the previous properties:

1. When one value is specified, it applies the same border width to all four sides of the element.
2. When two values are specified, the first value specifies the border width for the top and bottom and the second value specifies the border width for the left and right.

EVALUATION COPY: Not to be used in class.

3. When three values are specified, the first value is for the top, the second for the left and right, and the third for the bottom.
4. When four values are specified, the values specify border width in a clockwise direction starting with the top (top, right, bottom, left).
5. Alternatively, border width can be applied with the `border-top-width`, `border-right-width`, `border-bottom-width`, and `border-left-width` properties.

Demo 5.7: BordersMarginsAndPadding/Demos/border-width-styles.css

```
1.   /* The "box" class is used for display purposes only */
2.   .box {
3.     border-style: solid;
4.     margin: 1rem;
5.     padding: 1rem;
6.   }
7.
8.   #div1 {
9.     border-width: 1em;
10.  }
11.
12.  #div2 {
13.    border-width: 1em 1.5em;
14.  }
15.
16.  #div3 {
17.    border-width: 1em 1.5em 2em;
18.  }
19.
20.  #div4 {
21.    border-width: 1em 1.5em 2em 2.5em;
22.  }
23.
24.  #div5 {
25.    border-top-width: 1em;
26.    border-right-width: 1.5em;
27.    border-bottom-width: 2em;
28.    border-left-width: 2.5em;
29.  }
```

Evaluation Copy

Demo 5.8: BordersMarginsAndPadding/Demos/border-width.html

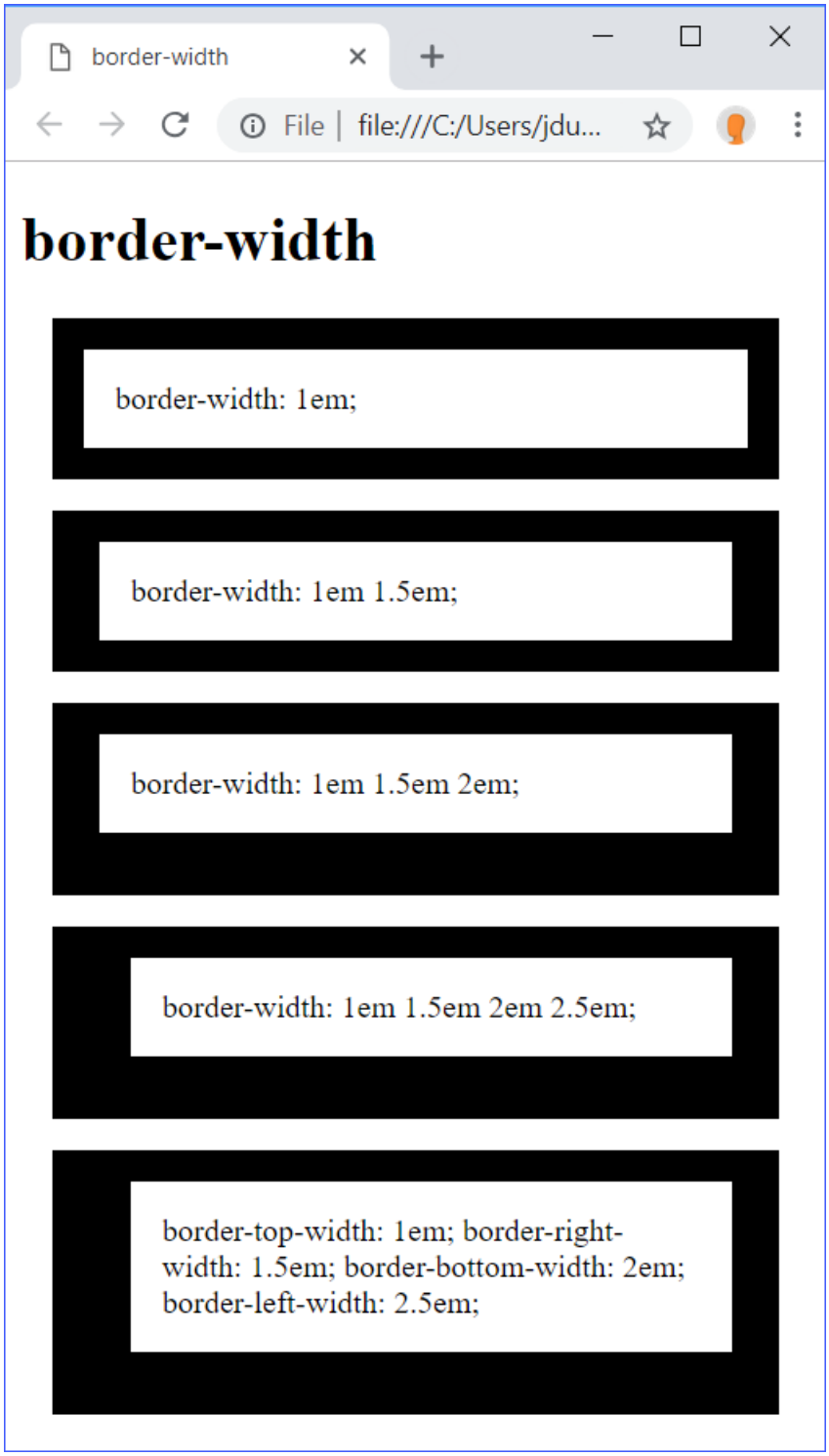
```
     -------Lines 1 through 8 Omitted-------
9.   <body class="webucator">
10.    <h1>border-width</h1>
11.    <div id="div1" class="box">
12.      border-width: 1em;
13.    </div>
14.    <div id="div2" class="box">
15.      border-width: 1em 1.5em;
16.    </div>
17.    <div id="div3" class="box">
18.      border-width: 1em 1.5em 2em;
19.    </div>
20.    <div id="div4" class="box">
21.      border-width: 1em 1.5em 2em 2.5em;
22.    </div>
23.    <div id="div5" class="box">
24.      border-top-width: 1em;
25.      border-right-width: 1.5em;
26.      border-bottom-width: 2em;
27.      border-left-width: 2.5em;
28.    </div>
29.  </body>
30.  </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:

EVALUATION COPY: Not to be used in class.


border-width
File | file:///C:/Users/jdu...
border-width
border-width: 1em;
border-width: 1em 1.5em;
border-width: 1em 1.5em 2em;
border-width: 1em 1.5em 2em 2.5em;
border-top-width: 1em; border-right-width: 1.5em; border-bottom-width: 2em; border-left-width: 2.5em;

No border when border-style is none.

Even with a `border-width`, an element will not show a border unless a `border-style` is specified.

Notice the `box` class that we included to add a `border-style` of `solid` and a `margin` and `padding` of `1rem` each.

❖ 5.4.2. border-style

The `border-style` shorthand property is used to define the styles of the top, right, bottom, and left borders of an element. Possible values are listed below:

1. `dashed`
2. `dotted`
3. `double`
4. `groove`
5. `hidden`
6. `inset`
7. `none`
8. `outset`
9. `ridge`
10. `solid`

Evaluation Copy

To apply different styles to individual sides, `border-style` works like the other properties we have seen:

1. When one value is specified, it applies the same border style to all four sides of the element.
2. When two values are specified, the first value specifies the border style for the top and bottom and the second value specifies the border style for the left and right.
3. When three values are specified, the first value is for the top, the second for the left and right, and the third for the bottom.
4. When four values are specified, the values specify border style in a clockwise direction starting with the top (top, right, bottom, left).

EVALUATION COPY: Not to be used in class.

5. Alternatively, border style can be applied with the `border-top-style`, `border-right-style`, `border-bottom-style`, and `border-left-style` properties.

Demo 5.9: BordersMarginsAndPadding/Demos/border-style-styles.css

```
1.   /* The "box" class is used for display purposes only */
2.   .box {
3.     border-width: 0.5rem;
4.     margin: 1rem;
5.     padding: 1rem;
6.   }
7.
8.   #dashed {
9.     border-style: dashed;
10.  }
11.
12.  #dotted-double {
13.    border-style: dotted double;
14.  }
15.
16.  #groove-hidden-inset {
17.    border-style: groove hidden inset;
18.  }
19.
20.  #none-outset-ridge-solid {
21.    border-style: none outset ridge solid;
22.  }
23.
24.  #longhand {
25.    border-top-style: double;
26.    border-right-style: groove;
27.    border-bottom-style: dashed;
28.    border-left-style: dotted;
29.  }
```

Evaluation Copy

Demo 5.10: BordersMarginsAndPadding/Demos/border-style.html

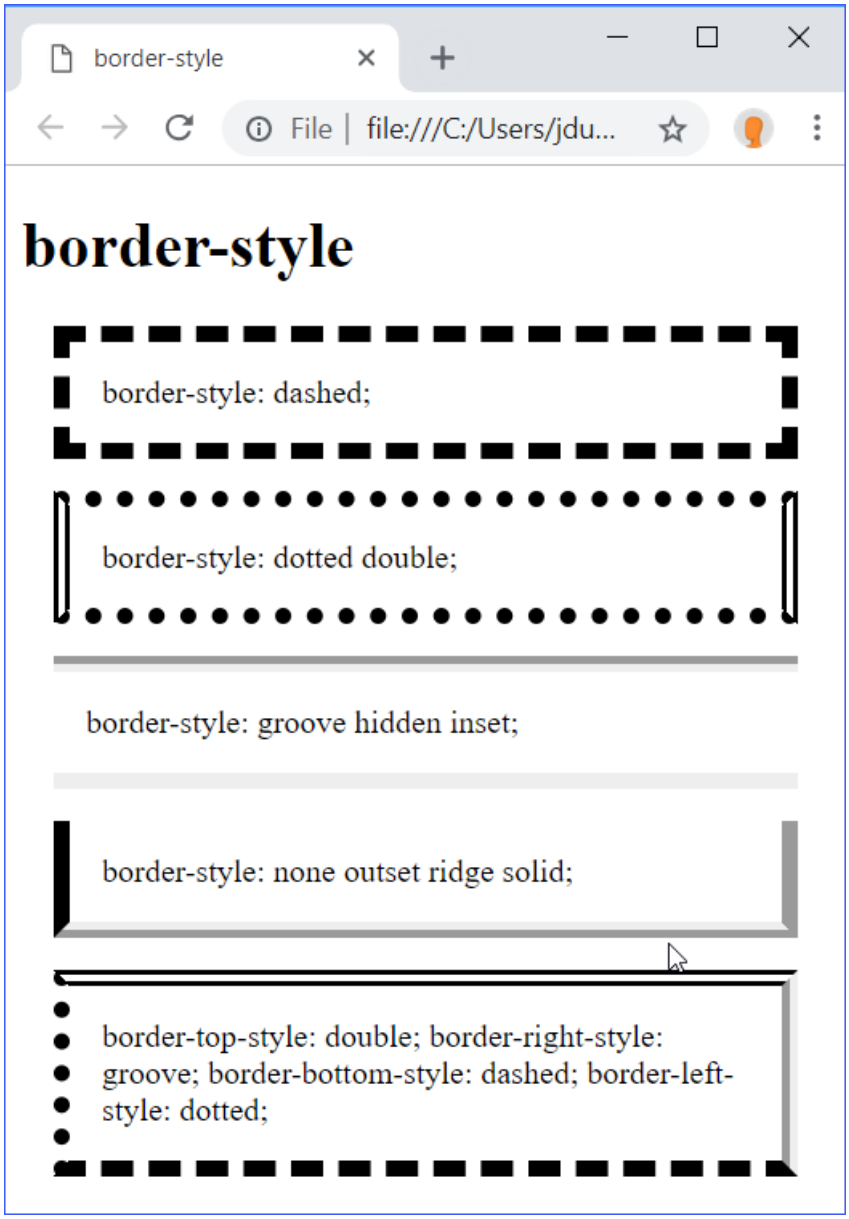
```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.    <h1>border-style</h1>
12.    <div id="dashed" class="box">
13.      border-style: dashed;
14.    </div>
15.    <div id="dotted-double" class="box">
16.      border-style: dotted double;
17.    </div>
18.    <div id="groove-hidden-inset" class="box">
19.      border-style: groove hidden inset;
20.    </div>
21.    <div id="none-outset-ridge-solid" class="box">
22.      border-style: none outset ridge solid;
23.    </div>
24.    <div id="longhand" class="box">
25.      border-top-style: double;
26.      border-right-style: groove;
27.      border-bottom-style: dashed;
28.      border-left-style: dotted;
29.    </div>
30.  </body>
31.  </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:

EVALUATION COPY: Not to be used in class.



❖ 5.4.3. border-color

The `border-color` shorthand property is used to define the color of the top, right, bottom, and left borders. It can take up to four color values.

To apply different border colors to individual sides, `border-color` works like the other properties we have seen:

1. When one value is specified, it applies the same border color to all four sides of the element.
2. When two values are specified, the first value specifies the border color for the top and bottom and the second value specifies the border color for the left and right.
3. When three values are specified, the first value is for the top, the second for the left and right, and the third for the bottom.
4. When four values are specified, the values specify border color in a clockwise direction starting with the top (top, right, bottom, left).
5. Alternatively, border color can be applied with the `border-top-color`, `border-right-color`, `border-bottom-color`, and `border-left-color` properties.

Demo 5.11: BordersMarginsAndPadding/Demos/border-color-styles.css

```
/* The "box" class is used for display purposes only */
.box {
  border-style: solid;
  border-width: 0.5rem;
  margin: 1rem;
  padding: 1rem;
}

#red {
  border-color: red;
}

#red-blue {
  border-color: red blue;
}

#red-blue-green {
  border-color: red blue green;
}

#red-blue-green-purple {
  border-color: red blue green purple;
}

#longhand {
  border-top-color: red;
  border-right-color: blue;
  border-bottom-color: green;
  border-left-color: purple;
}
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

Demo 5.12: BordersMarginsAndPadding/Demos/border-color.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.    <h1>border-color</h1>
12.    <div id="red" class="box">
13.      border-color: red;
14.    </div>
15.    <div id="red-blue" class="box">
16.      border-color: red blue;
17.    </div>
18.    <div id="red-blue-green" class="box">
19.      border-color: red blue green;
20.    </div>
21.    <div id="red-blue-green-purple" class="box">
22.      border-color: red blue green purple;
23.    </div>
24.    <div id="longhand" class="box">
25.      border-top-color: red;
26.      border-right-color: blue;
27.      border-bottom-color: green;
28.      border-left-color: purple;
29.    </div>
30.  </body>
31.  </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:

❖ 5.4.4. border-radius

The `border-radius` property offers an easy method for adding rounded corners to elements. Each corner can take either one (for circular corners) or two (for elliptical corners) radii value specified as either a length value in units (`em`, `rem`, `px`, etc.) or as a percentage of the width (horizontal) or height (vertical) of the box.

1. To specify one radius value for each corner, `border-radius` works similarly to other properties we have looked at:
    A. When one value is specified, it applies the border radius to all four corners of the element.
    B. When two values are specified, the first value specifies the border radius for the top-left corner and the bottom-right and the second value specifies the border radius for the top-right corner and the bottom-left corner.
    C. When three values are specified, the first value is for the top-left, the second for the top-right and bottom-left, and the third for the bottom-right.
    D. When four values are specified, the values specify border radius in a clockwise direction starting with the top-left (top-left, top-right, bottom-right, bottom-left).

2. To specify two radii values for each corner, the second radii are separated from the first radii by a forward slash (`/`), and they are applied to each corner in the same manner that the first radii are applied. For example, `1rem 2rem 3rem / 0.5rem 1.5rem` would apply `1rem` and `0.5rem` radii to the top-left corner, `2rem` and `1.5rem` radii to the top-right and bottom-left corners, and `3rem` and `0.5rem` radii to the bottom-right corner.

Evaluation Copy

EVALUATION COPY: Not to be used in class.

Demo 5.13: BordersMarginsAndPadding/Demos/border-radius-styles.css

```
.border {
  border-style: solid;
  padding: 1rem;
}

#div1 {
  border-radius: 1rem;
}

#div2 {
  border-radius: 1rem 2rem;
}

#div3 {
  border-radius: 1rem 2rem / 0.5rem;
}

#div4 {
  border-radius: 1rem 2rem 3rem / 0.5rem 1.5rem;
}

#div5 {
  border-radius: 1rem 2rem 3rem 4rem / 0.5rem 1.5rem 2.5rem 3.5rem;
}
```

Evaluation Copy

Demo 5.14: BordersMarginsAndPadding/Demos/border-radius.html

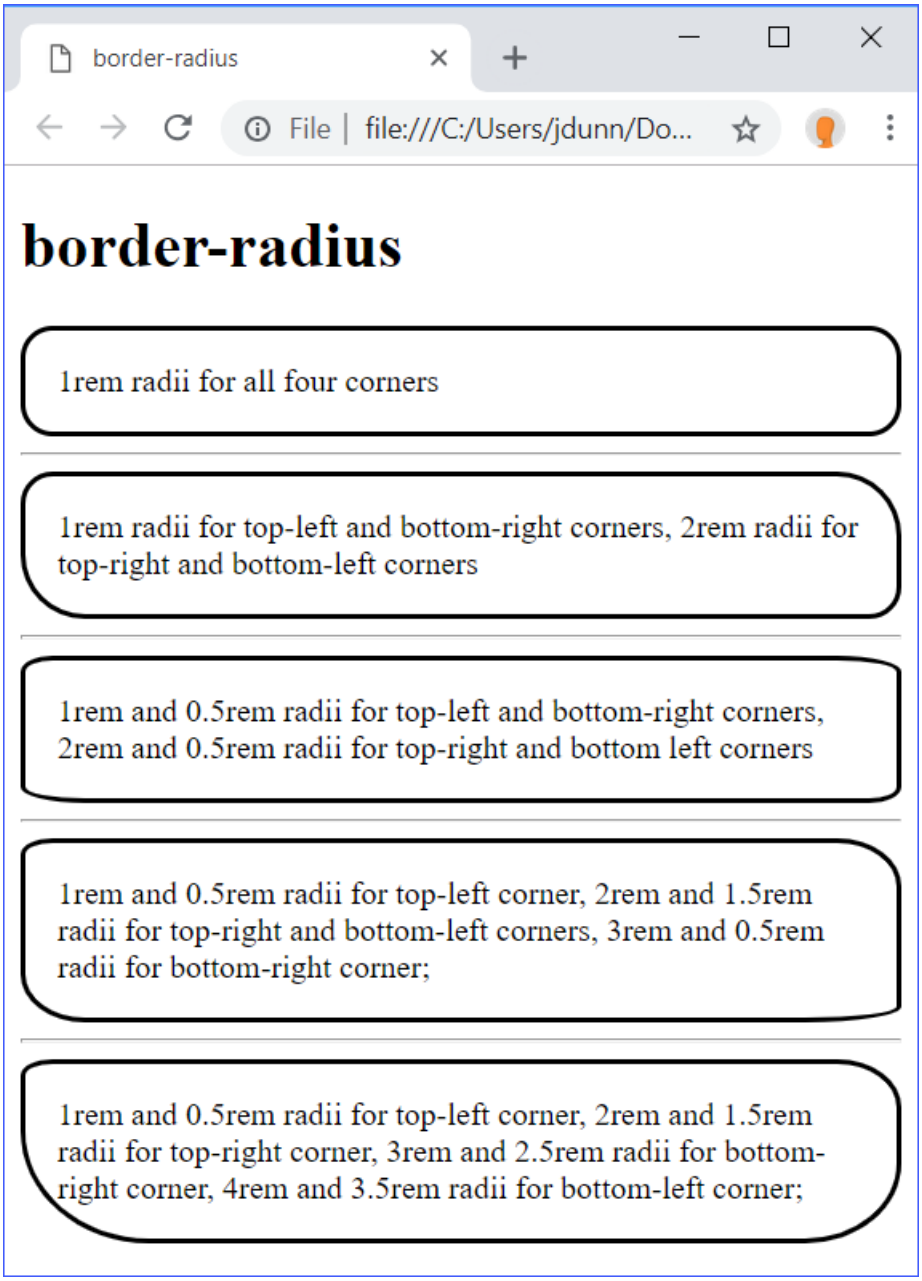
```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.    <h1>border-radius</h1>
12.    <div id="div1" class="border">
13.      1rem radii for all four corners
14.    </div>
15.    <hr>
16.    <div id="div2" class="border">
17.      1rem radii for top-left and bottom-right corners,
18.      2rem radii for top-right and bottom-left corners
19.    </div>
20.    <hr>
21.    <div id="div3" class="border">
22.      1rem and 0.5rem radii for top-left and bottom-right corners,
23.      2rem and 0.5rem radii for top-right and bttom left corners
24.    </div>
25.    <hr>
26.    <div id="div4" class="border">
27.      1rem and 0.5rem radii for top-left corner,
28.      2rem and 1.5rem radii for top-right and bottom-left corners,
29.      3rem and 0.5rem radii for bottom-right corner;
30.    </div>
31.    <hr>
32.    <div id="div5" class="border">
33.      1rem and 0.5rem radii for top-left corner,
34.      2rem and 1.5rem radii for top-right corner,
35.      3rem and 2.5rem radii for bottom-right corner,
36.      4rem and 3.5rem radii for bottom-left corner;
37.    </div>
38.  </body>
39.  </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:

EVALUATION COPY: Not to be used in class.



5.5. box-sizing

The `box-sizing` property defines how the `height` and `width` properties work to define the total dimensions of your box. `box-sizing` accepts two values:

1. `content-box` - the default. When `content-box` is specified, `width` and `height` refer to the size of the content and **do not** include padding, border, or margin.
2. `border-box` - When `border-box` is specified, the `width` and `height` include content, padding, and border, but **do not** include margin.

When trying to fit one block element inside of another, it is often best to set `box-sizing` to `border-box` as the following demo illustrates:

Demo 5.15: BordersMarginsAndPadding/Demos/box-sizing-styles.css

```
1.   .parent {
2.     background-color: lightyellow;
3.     border: 1px solid black;
4.     height: 300px;
5.     width: 300px;
6.   }
7.
8.   #parent1 {
9.     margin-bottom: 60px;
10.  }
11.
12.  /* The "box" class is used for display purposes only */
13.  .box {
14.    height: 280px;
15.    width: 280px;
16.    padding: 10px;
17.    border: 10px solid red;
18.    margin: 10px;
19.    white-space: pre;
20.  }
21.
22.  #content-box {
23.    box-sizing: content-box;
24.  }
25.
26.  #border-box {
27.    box-sizing: border-box;
28.  }
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

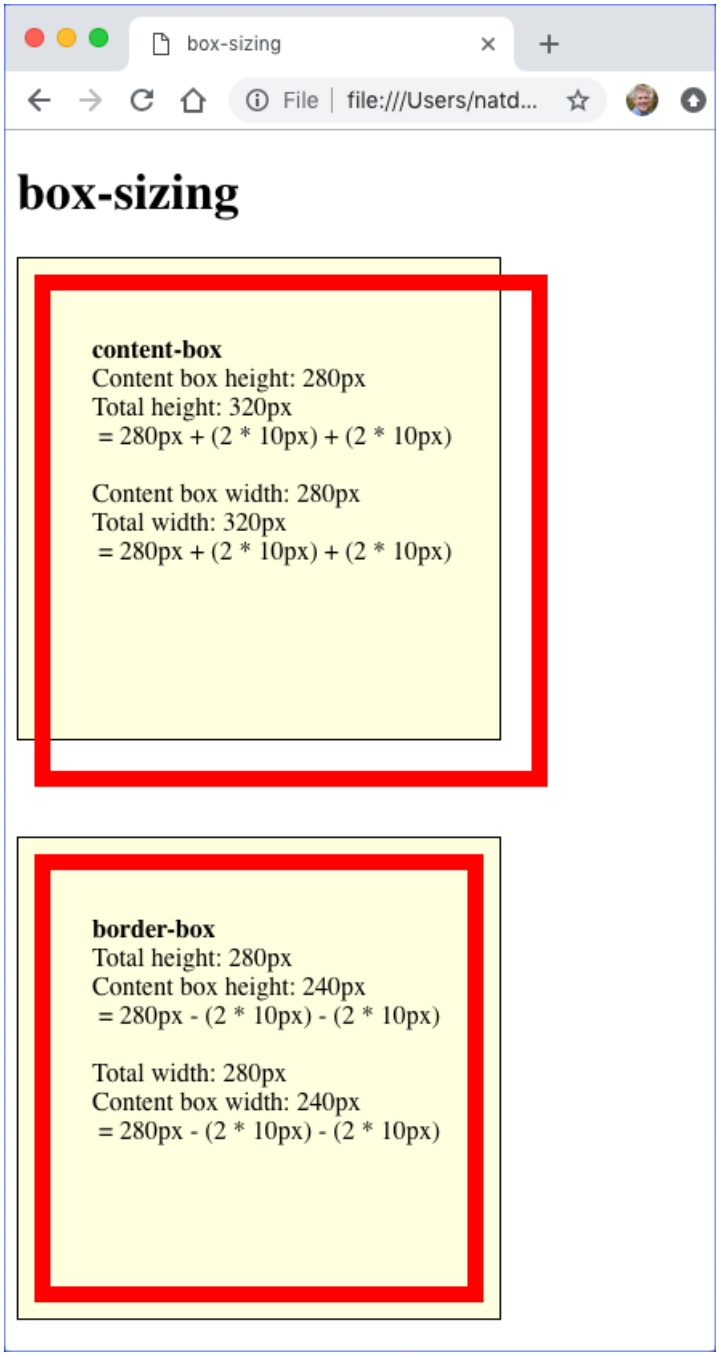
Demo 5.16: BordersMarginsAndPadding/Demos/box-sizing.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.  <h1>box-sizing</h1>
12.  <div id="parent1" class="parent">
13.    <div id="content-box" class="box">
14.      <strong>content-box</strong>
15.      Content box height: 280px
16.      Total height: 320px
17.       = 280px + (2 * 10px) + (2 * 10px)
18.
19.      Content box width: 280px
20.      Total width: 320px
21.       = 280px + (2 * 10px) + (2 * 10px)
22.    </div>
23.  </div>
24.  <div class="parent">
25.    <div id="border-box" class="box">
26.      <strong>border-box</strong>
27.      Total height: 280px
28.      Content box height: 240px
29.       = 280px - (2 * 10px) - (2 * 10px)
30.
31.      Total width: 280px
32.      Content box width: 240px
33.       = 280px - (2 * 10px) - (2 * 10px)
34.    </div>
35.  </div>
36.  </body>
37.  </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:



Things to note:

1. The containing parent boxes are both 300px by 300px.
2. The children boxes are both set to 280px by 280px.

EVALUATION COPY: Not to be used in class.

3. The border and padding of the first child box, which has `box-sizing` set to `content-box`, **are added** to the sizes, extending the box past the borders of the parent box.
4. The border and padding of the second child box, which has `box-sizing` set to `border-box`, **are not added** to the sizes, so the box stays within the parent box.

---

min-height, min-width, max-height, max-width

In addition to the `height` and `width` properties, there are also `min-height`, `min-width`, `max-width`, and `max-height` properties. These properties work exactly like `height` and `width`, except that they specify minimums and maximums. This means that if the window size increases, the box can expand up to its maximum values, and if the window shrinks, the box won't shrink more than its minimum values. Using these properties instead of `height` and `width` allows for more responsive designs.

5.6. box-shadow

Evaluation Copy

The `box-shadow` adds a shadow to an element in the same way that `text-shadow` adds a shadow to text. It can take a comma-delimited list of shadows, each described by a combination of x and y offsets, blur radius, spread radius, and color. There is also an optional `inset` value. The values are explained below:

1. `x-offset` and `y-offset`: required length values.
    A. `x-offset` must come immediately before `y-offset`.
    B. `x-offset` specifies the horizontal displacement (positive values to the right and negative values to the left).
    C. `y-offset` specifies the vertical displacement (positive values below and negative values above).
    D. If both values are zero, the shadow will be directly behind the box.
2. `blur-radius`: an optional length value that specifies the size of the blur effect.
    A. If included, this value must come after the two offset values.
3. `spread-radius`: an optional length value that determines the size of the shadow.

A. Positive values will make the shadow grow.

B. Negative values will make the shadow shrink.

C. 0 (the default) means it will be the same size as the box.

4. `color`: optional color value.

    A. If included, must be placed either immediately before or after the length values.

    B. If not included, the browser picks the color, so for consistency across browsers, it is a good idea to specify the shadow's color.

5. `inset`: an optional keyword value that, when included, makes the shadow appear inside instead of outside of the box.

    A. If included, must be placed either immediately before or after the length values.

Demo 5.17: BordersMarginsAndPadding/Demos/box-shadow-styles.css

```
/* The "box" class is used for display purposes only */
.box {
  background-color: lightblue;
  border: 1px solid black;
  float: left; /* allows boxes to be side by side */
  height: 100px;
  margin: 15px;
  padding: 15px;
  text-align: center;
  width: 200px;
}

#div1 {
  box-shadow: .5rem .5rem gray;
}

#div2 {
  box-shadow: .5rem .5rem 1rem gray;
}

#div3 {
  box-shadow: .5rem .5rem 1rem .25rem gray;
}

#div4 {
  box-shadow: inset -.5rem -.5rem 1.5rem .25rem gray;
}
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

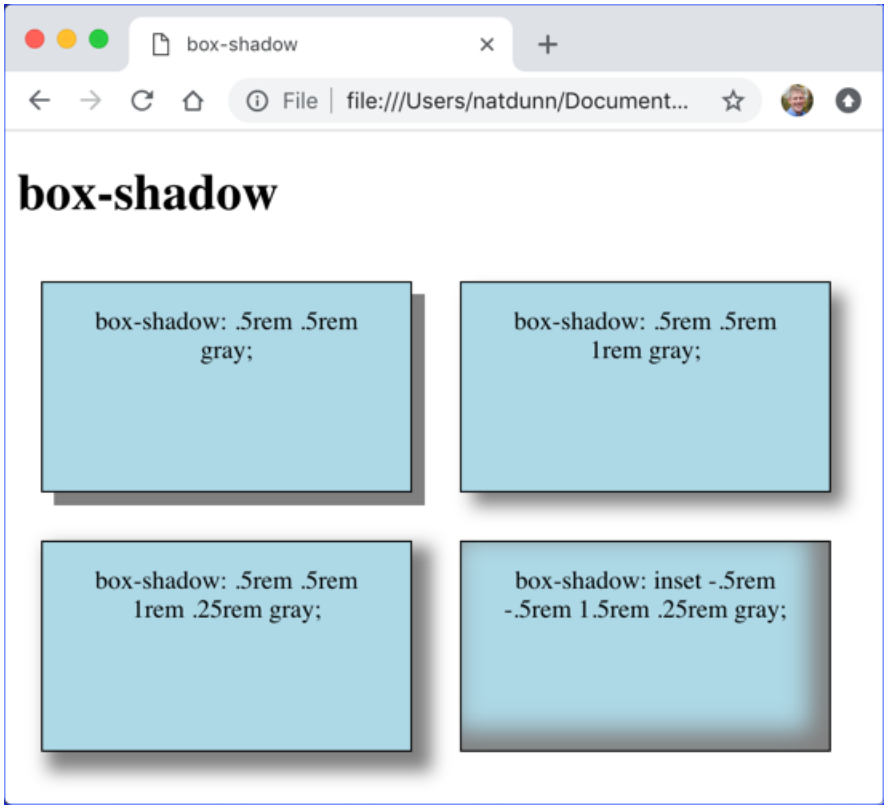
Demo 5.18: BordersMarginsAndPadding/Demos/box-shadow.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.    <h1>box-shadow</h1>
12.    <div id="div1" class="box">
13.      box-shadow: .5rem .5rem gray;
14.    </div>
15.    <div id="div2" class="box">
16.      box-shadow: .5rem .5rem 1rem gray;
17.    </div>
18.    <div id="div3" class="box">
19.      box-shadow: .5rem .5rem 1rem .25rem gray;
20.    </div>
21.    <div id="div4" class="box">
22.      box-shadow: inset -.5rem -.5rem 1.5rem .25rem gray;
23.    </div>
24.  </body>
25.  </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:



See Mozilla's Box-shadow generator[27] for a great tool to help design box shadows.

27. https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Background_and_Borders/Box-shadow_generator

EVALUATION COPY: Not to be used in class.

Exercise 8: Borders, Margin, and Padding

25 to 40 minutes

In this exercise, you will continue to work on the Runners Home page.

1. Open `BordersMarginAndPadding/Exercises/index.html` in your editor. You will see the home page for a website called Runners Home.
2. Open `BordersMarginAndPadding/Exercises/styles.css`. This is the stylesheet from the solution to the Text exercise. Feel free to continue working from your own stylesheet.
3. Using your new knowledge of borders, margin, and padding, add some styles to this page. The object of this exercise is to practice using the properties you just learned about. Also, feel free to add or change any other properties that you know.
4. When you are done, open `index.html` in your browser to see the results. You are welcome to go back to the code and continue to work.

Evaluation Copy

You can design it however you like, or you can try to make it look something like:

- Home
- Races
- Resources
- Calculator
- Running Log
- My Account
- Log out

Hello, Stranger!

Welcome to Runners Home™

Runners Home™ is dedicated to providing you with:

1. the most up-to-date information on running races.
2. the best resources for runners.

Best Running Tips

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam et gravida sapien, facilisis condimentum arcu. Morbi eget dui iaculis, porttitor eros et, tincidunt erat. Sed sollicitudin, mauris vitae semper consequat, purus velit aliquet mi, sit amet posuere eros tortor eget neque. Phasellus hendrerit justo... Click here to read more!

Best Health Tips

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam et gravida sapien, facilisis condimentum arcu. Morbi eget dui iaculis, porttitor eros et, tincidunt erat. Sed sollicitudin, mauris vitae semper consequat, purus velit aliquet mi, sit amet posuere eros tortor eget neque. Phasellus hendrerit justo... Click here to read more!

Newsletter

Email: Sign Up

Be the first to hear about our great offers.
Sign up for our newsletter today!

© 2020 Runners Home. All rights reserved. For questions, send email to info@runnershome.com.

EVALUATION COPY: Not to be used in class.

Solution: BordersMarginsAndPadding/Solutions/styles.css

```
1.  html {
2.    font-size: 16px;
3.  }
4.
5.  body {
6.    font: 1rem/1.3 Verdana, Geneva, Tahoma, sans-serif;
7.  }
8.
9.  nav {
10.   border: solid rgb(110, 78, 226) 0.1rem;
11.   border-radius: 1rem;
12.   box-shadow: 0.2rem 0.2rem 1rem rgb(110, 78, 226, .5);
13.   margin: 1rem;
14.   width: 40%;
15. }
16.
17. nav a {
18.   text-decoration: none;
19. }
20.
21. main {
22.   padding: 2%;
23.   max-width: 1000px;
24.   min-width: 750px;
25. }
26.
27. #welcome {
28.   font-size: larger;
29. }
30.
31. #welcome h1 {
32.   color: rgb(8, 5, 211);
33.   text-shadow: 0.2rem 0.2rem 0.1rem rgb(211, 200, 238);
34. }
35.
36. article {
37.   border: 1px ridge rgb(226, 77, 40);
38.   box-sizing: border-box;
39.   margin: 1em auto;
40.   padding: 1rem;
41.   max-width: 75%;
42.   text-align: center;
43. }
44.
```

EVALUATION COPY: Not to be used in class.

```
45.  article h2 {
46.    font-variant: small-caps;
47.    letter-spacing: 0.1em;
48.    word-spacing: 0.2em;
49.  }
50.
51.  article p {
52.    line-height: 2;
53.    text-align: left;
54.    text-indent: 3rem;
55.  }
56.
57.  .read-more {
58.    text-decoration: none;
59.  }
60.
61.  aside label {
62.    font-weight: bold;
63.  }
64.
65.  #newsletter {
66.    border: 1px dashed rgb(110, 78, 226);
67.    border-radius: 1em;
68.    margin: 0 1rem;
69.    padding: .5rem 1rem;
70.    text-align: right;
71.  }
72.
73.  footer {
74.    font-size: smaller;
75.    font-style: italic;
76.    margin: 1rem;
77.  }
```

Evaluation Copy

Conclusion

In this lesson, you have learned to use the CSS border, margin, and padding properties, and you have learned about the CSS basic box model.

EVALUATION COPY: Not to be used in class.

LESSON 6
Backgrounds

Topics Covered

☑ `background-color`

☑ `background-image`

☑ `background-repeat`

☑ `overflow`

☑ `background-attachment`

☑ `background-position`

☑ `background-size`

☑ `background-origin`

☑ `background-clip`

☑ `background`

Evaluation Copy

Introduction

In this lesson, you will learn to add backgrounds to elements on the page.

6.1. background-color

The `background-color` property sets the background color of an element. `background-color` can take any of the color values we discussed in the Color and Opacity lesson (see page 85).

Demo 6.1: Backgrounds/Demos/background-color-styles.css

```
1.   /* The "box" class is used for display purposes only */
2.   .box {
3.     border: 1px solid black;
4.     float: left; /* allows boxes to be side by side */
5.     height: 100px;
6.     padding: 1rem;
7.     margin: 1rem;
8.     width: 200px;
9.   }
10.
11.  #div1 {
12.    background-color: red;
13.  }
14.
15.  #div2 {
16.    background-color: #345;
17.  }
18.
19.  #div3 {
20.    background-color: rgba(6, 114, 6, 0.2);
21.  }
22.
23.  #div4 {
24.    background-color: hsl(230, 50%, 70%);
25.  }
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

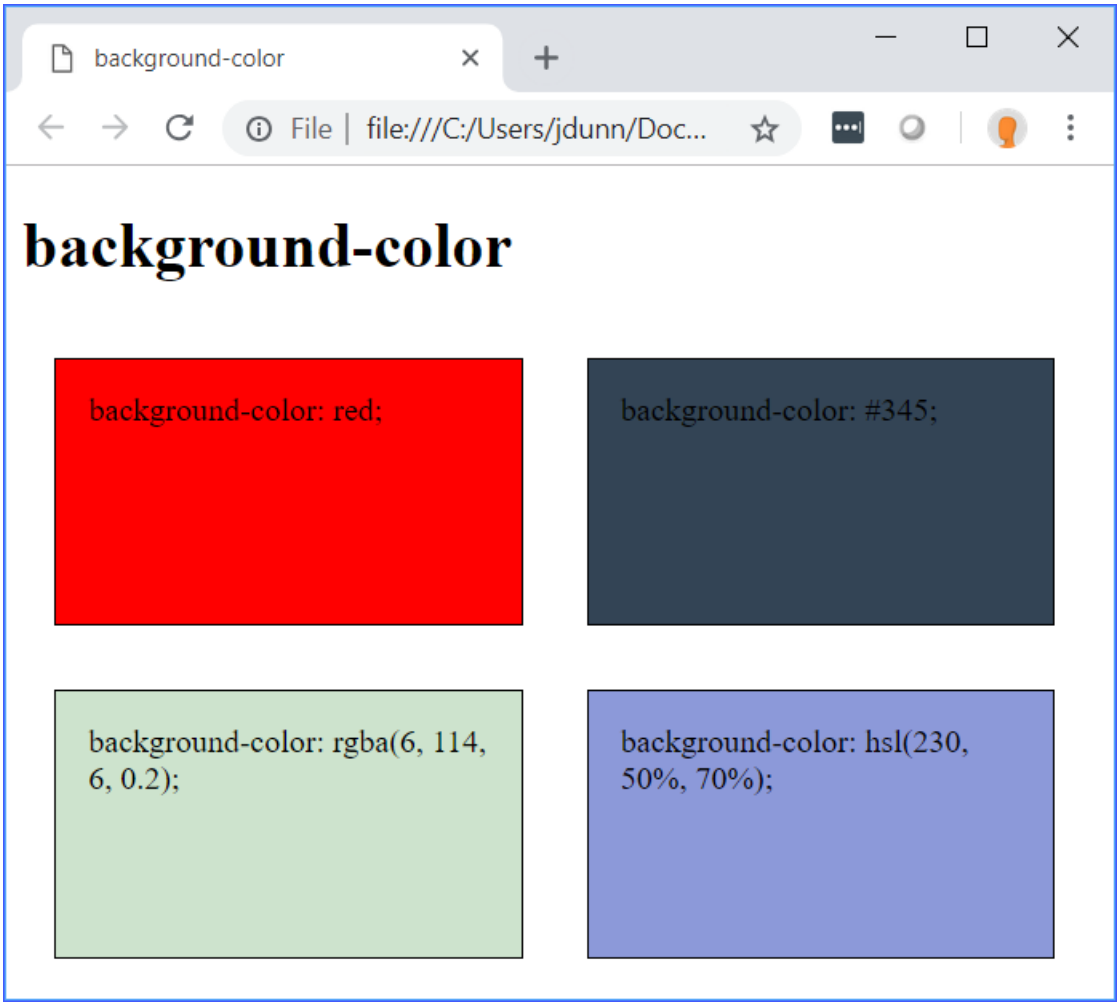
Demo 6.2: Backgrounds/Demos/background-color.html

```
      -------Lines 1 through 9 Omitted-------
10.   <body class="webucator">
11.     <h1>background-color</h1>
12.     <div id="div1" class="box">
13.       background-color: red;
14.     </div>
15.     <div id="div2" class="box">
16.       background-color: #345;
17.     </div>
18.     <div id="div3" class="box">
19.       background-color: rgba(6, 114, 6, 0.2);
20.     </div>
21.     <div id="div4" class="box">
22.       background-color: hsl(230, 50%, 70%);
23.     </div>
24.   </body>
25.   </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:



6.2. background-image

The `background-image` property sets a background image for an element. `background-image` takes a path to an image file. The borders of the element are drawn on top of the background image, and the background color is drawn below it. By default, a background image will repeat until it fills up the entire background of the element. In the following demo, we will use a single star as a background image:

EVALUATION COPY: Not to be used in class.

Gradient Values

`background-image` also takes gradient values. See Mozilla's <gradient>[28].

Demo 6.3: Backgrounds/Demos/background-image-styles.css

```
1.   /* The "box" class is used for display purposes only */
2.   .box {
3.     border: 1px solid black;
4.     font-weight: bold;
5.     height: 150px;
6.     padding: 1rem;
7.     margin: 1rem;
8.     width: 300px;
9.   }
10.
11.  #div1 {
12.    background-image: url(images/star.png);
13.  }
```

Evaluation Copy

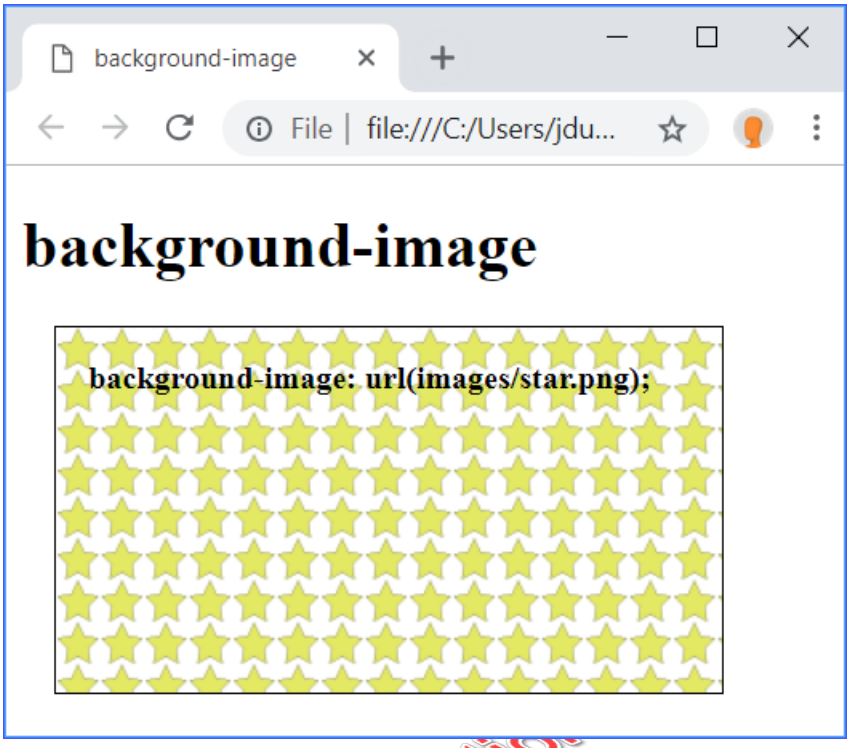
Demo 6.4: Backgrounds/Demos/background-image.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.    <h1>background-image</h1>
12.    <div id="div1" class="box">
13.      background-image: url(images/star.png);
14.    </div>
15.  </body>
16.  </html>
```

Code Explanation

The code above will render the following:

28. https://developer.mozilla.org/en-US/docs/Web/CSS/gradient



Evaluation Copy

6.3. background-repeat

The `background-repeat` property sets how background images are repeated. `background-repeat` takes the following values:

1. `repeat` - The default value. The image is repeated as much as needed to cover the whole background. The last images may be clipped if they don't fit.
2. `space` - The image is repeated as much as much as possible without clipping, and the extra whitespace is distributed evenly between all the images.
3. `round` - The repeated images are stretched to fill the space, so that there are no gaps between the images.
4. `no-repeat` - The image is not repeated.

`background-repeat` can take one or two of the above values.

1. If one value is specified, it applies to how the image repeats horizontally and vertically.

EVALUATION COPY: Not to be used in class.

2. If two values are specified, the first value refers to how the image repeats horizontally and the second value refers to how the images repeat vertically.
3. To repeat an image horizontally, but not vertically, the single value `repeat-x` can be used instead of two values (`repeat no-repeat`).
4. To repeat an image vertically, but not horizontally, the single value `repeat-y` can be used instead of two values (`no-repeat repeat`).

Evaluation Copy

Demo 6.5: Backgrounds/Demos/background-repeat-styles.css

```
1.   /* The "box" class is used  for display purposes only */
2.   .box {
3.     background-image: url(images/circle.png);
4.     border: 1px solid black;
5.     float: left; /* allows boxes to be side by side */
6.     font-weight: bold;
7.     height: 100px;
8.     padding: 1rem;
9.     margin: 1rem;
10.    width: 235px;
11.  }
12.
13.  #repeat {
14.    background-repeat: repeat;
15.  }
16.
17.  #space {
18.    background-repeat: space;
19.  }
20.
21.  #round {
22.    background-repeat: round;
23.  }
24.
25.  #no-repeat {
26.    background-repeat: no-repeat;
27.  }
28.
29.  #repeat-x {
30.    background-repeat: repeat-x;
31.  }
32.
33.  #repeat-y {
34.    background-repeat: repeat-y;
35.  }
36.
37.  #repeat-space {
38.    background-repeat: repeat space;
39.  }
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

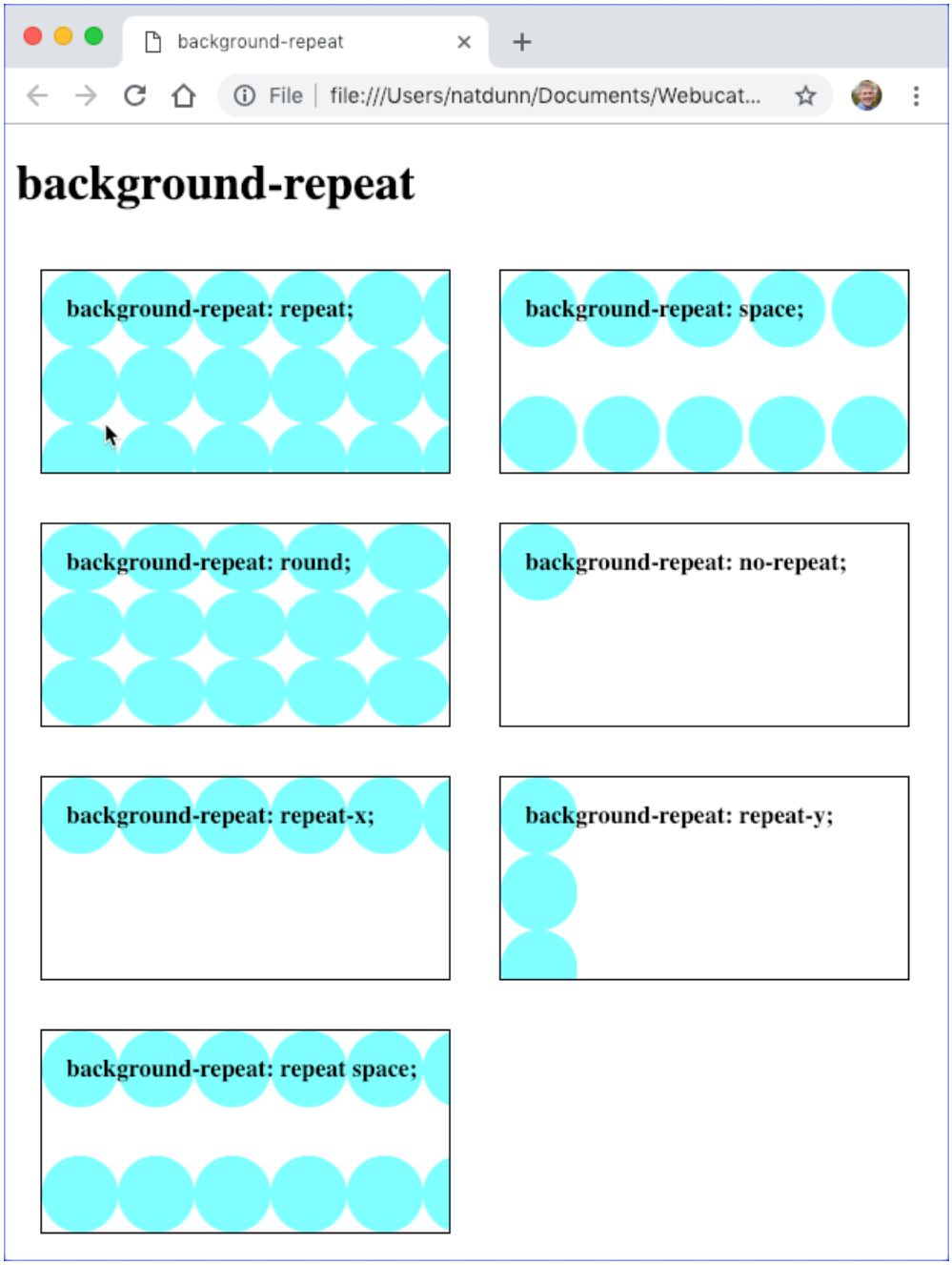
Demo 6.6: Backgrounds/Demos/background-repeat.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.    <h1>background-repeat</h1>
12.    <div id="repeat" class="box">
13.        background-repeat: repeat;
14.    </div>
15.    <div id="space" class="box">
16.      background-repeat: space;
17.    </div>
18.    <div id="round" class="box">
19.      background-repeat: round;
20.    </div>
21.    <div id="no-repeat" class="box">
22.      background-repeat: no-repeat;
23.    </div>
24.    <div id="repeat-x" class="box">
25.      background-repeat: repeat-x;
26.    </div>
27.    <div id="repeat-y" class="box">
28.      background-repeat: repeat-y;
29.    </div>
30.    <div id="repeat-space" class="box">
31.      background-repeat: repeat space;
32.    </div>
33.  </body>
34.  </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:



EVALUATION COPY: Not to be used in class.

6.4. overflow

The `overflow` property sets what happens when the content of a container element is too big for that element. It takes one of the following values:

1. `visible` - The default value. The content is not clipped and overflows outside of the container.
2. `hidden` - The content is cut off to fit inside the container.
3. `scroll` - The content is cut off to fit inside the container, and a scroll bar is added to allow the user to scroll to see the content.
4. `auto` - If the content fits, no scroll bar is added, but if it doesn't fit, then an appropriate scroll bar is added.

Demo 6.7: Backgrounds/Demos/overflow-styles.css

```
/* The "box" class is used for display purposes only */
.box {
  border: 1px solid black;
  float: left;
  height: 200px;
  margin: 0 0 4rem 0.5rem;
  width: 45%;
}

#visible {
  overflow: visible;
}

#hidden {
  overflow: hidden;
}

#scroll {
  overflow: scroll;
}

#auto {
  overflow: auto;
}
```

Evaluation Copy

Demo 6.8: Backgrounds/Demos/overflow.html

```
       -------Lines 1 through 9 Omitted-------
10.    <body class="webucator">
11.    <h1>overflow</h1>
12.      <div id="visible" class="box">
13.        <strong>visible</strong>
14.        <p>Lorem ipsum dolor sit amet,
15.          consectetur adipiscing elit.
16.          Praesent viverra dictum purus,
17.          ac iaculis mi maximus nec.
18.          Cras vitae diam dolor.
       -------Lines 19 through 23 Omitted-------
24.        </p>
25.      </div>
26.      <div id="hidden" class="box">
27.        <strong>hidden</strong>
28.        <p>Lorem ipsum dolor sit amet,
29.          consectetur adipiscing elit.
30.          Praesent viverra dictum purus,
31.          ac iaculis mi maximus nec.
32.          Cras vitae diam dolor.
       -------Lines 33 through 37 Omitted-------
38.        </p>
39.      </div>
40.      <div id="scroll" class="box">
41.        <strong>scroll</strong>
42.        <p>Lorem ipsum dolor sit amet,
43.          consectetur adipiscing elit.
44.          Praesent viverra dictum purus,
45.          ac iaculis mi maximus nec.
46.          Cras vitae diam dolor.
       -------Lines 47 through 51 Omitted-------
52.        </p>
53.      </div>
54.      <div id="auto" class="box">
55.        <strong>auto</strong>
56.        <p>Lorem ipsum dolor sit amet,
57.          consectetur adipiscing elit.
58.          Praesent viverra dictum purus,
59.          ac iaculis mi maximus nec.
60.          Cras vitae diam dolor.
       -------Lines 61 through 65 Omitted-------
66.        </p>
67.      </div>
```

Evaluation Copy

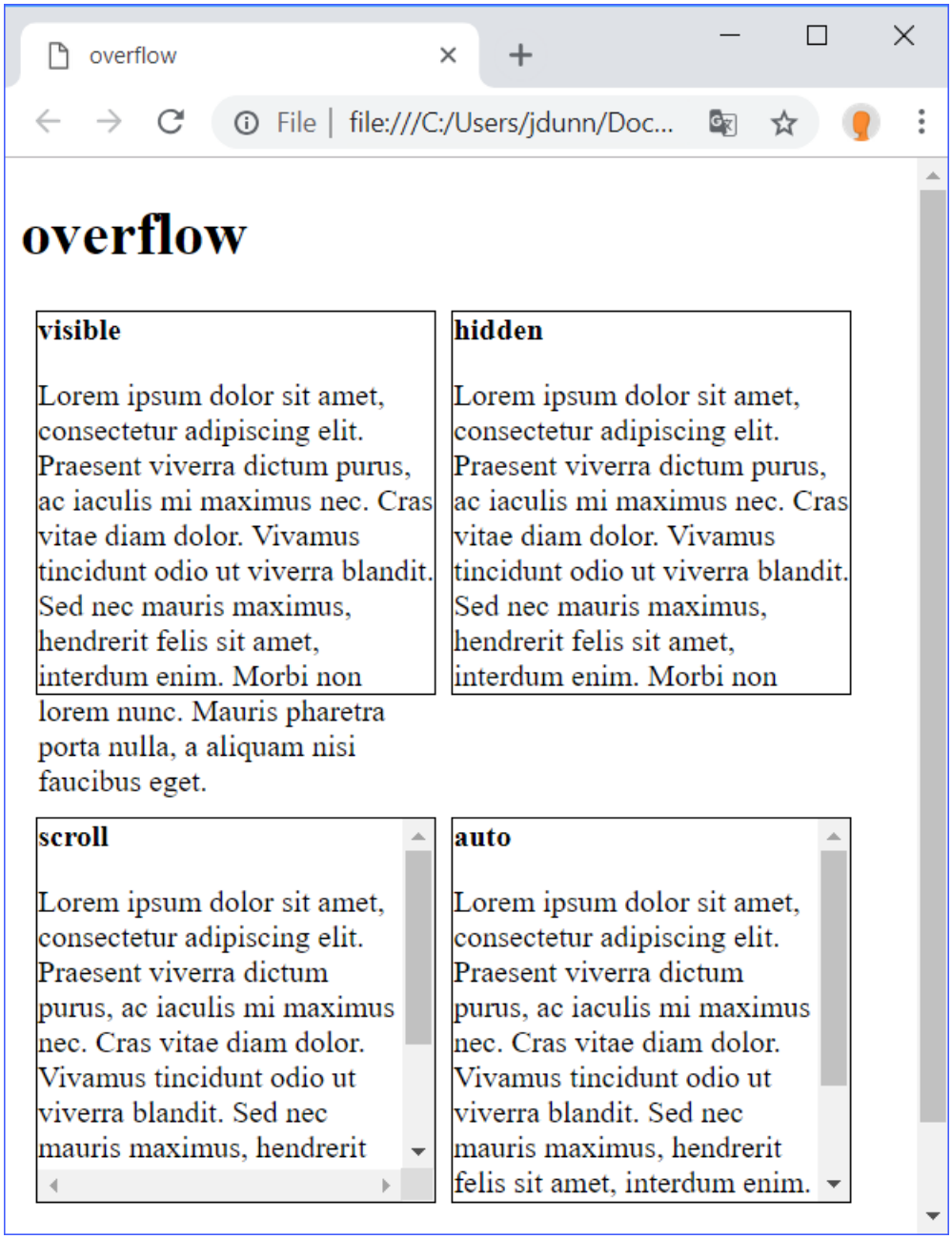
EVALUATION COPY: Not to be used in class.

```
68.   </body>
69.   </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:



EVALUATION COPY: Not to be used in class.

6.5. background-attachment

The `background-attachment` property sets whether the background image scrolls with its element or is fixed relative to the viewport. Possible values are:

1. `scroll` - The default value. The background image is fixed relative to and scrolls with its element.
2. `fixed` - The background image is fixed relative to the viewport, so it will not move with its element.
3. `local` - The background image is fixed to the contents of its element. This means that if the element has a scrolling device, the image will scroll with the contents, which differentiates `local` from `scroll`.

Evaluation Copy

Demo 6.9: Backgrounds/Demos/background-attachment-styles.css

```
1.   /* The "container" id is used for display purposes only */
2.   #container{
3.     border: 1px solid black;
4.     height: 400px;
5.     overflow: auto;
6.     width: 300px;
7.   }
8.
9.   /* The "box" class is used for display purposes only */
10.  .box {
11.    background-repeat: repeat-y;
12.    border: 1px solid black;
13.    height: 100px;
14.    margin: 1rem;
15.    padding: 1rem;
16.    overflow: auto;
17.    width: 200px;
18.  }
19.
20.  #scroll {
21.    background-attachment: scroll;
22.    background-image: url(images/circle.png);
23.  }
24.
25.  #fixed {
26.    background-attachment: fixed;
27.    background-image: url(images/circle.png);
28.  }
29.
30.  #local {
31.    background-attachment: local;
32.    background-image: url(images/circle.png);
33.  }
```

Evaluation Copy

Code Explanation

A few things to note:

1. We created a `container` `div` with a scrolling device to hold the `scroll`, `fixed`, and `local` `div` elements.
2. We used `strong` tags as labels for each inner `div`.

EVALUATION COPY: Not to be used in class.

Demo 6.10: Backgrounds/Demos/background-attachment.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.  <h1>background-attachment</h1>
12.  <div id="container">
13.    <strong>background-attachment: scroll;</strong>
14.    <div id="scroll" class="box">
15.      <p>Notice that:</p>
16.      <ol>
17.        <li>The background image moves when
18.          the "container" div is scrolled.</li>
19.        <li>The background image does not move when
20.          this div is scrolled.</li>
21.      </ol>
22.      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
     -------Lines 23 through 28 Omitted-------
29.      </p>
30.    </div>
31.    <strong>background-attachment: fixed;</strong>
32.    <div id="fixed" class="box">
33.      <p>Notice that:</p>
34.      <ol>
35.        <li>The background image does not move when
36.          the "container" div is scrolled.</li>
37.        <li>The background image does not move when
38.          this div is scrolled.</li>
39.      </ol>
40.      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
     -------Lines 41 through 46 Omitted-------
47.      </p>
48.    </div>
49.    <strong>background-attachment: local;</strong>
50.    <div id="local" class="box">
51.      <p>Notice that:</p>
52.      <ol>
53.        <li>The background image moves when
54.          the "container" div is scrolled.</li>
55.        <li>The background image moves when
56.          this div is scrolled.</li>
57.      </ol>
58.      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
     -------Lines 59 through 64 Omitted-------
65.      </p>
66.    </div>
```

Evaluation Copy

```
67.   </div>
68.   </body>
69.   </html>
```

Code Explanation

The code above will render the following:

EVALUATION COPY: Not to be used in class.

Open `background-attachment.html` in your browser and play with all the scrollbars to clearly see the different effects of `scroll`, `fixed`, and `local`.

6.6. background-position

The `background-position` property specifies the initial positions of the background images. These positions are relative to the area set by the `background-origin` property. There are two ways to specify `background-position`:

1. Keywords
2. Coordinates

❖ 6.6.1. Keywords

`background-position` takes the following keywords:

1. `center` - centers the image.

2. `top` - centers the image horizontally and positions it against the top edge of the element.

3. `bottom` - centers the image horizontally and positions it against the bottom edge of the element.

4. `left` - centers the image vertically and positions it against the left edge of the element.

5. `right` - centers the image vertically and positions it against the right edge of the element.

It is possible to use the keywords (except `center`) to specify different locations by adding offset values (either length values in units, or percentage values). For example:

```
background-position: top 10px left 10px;
```

This code would put the image `10px` from the top and left edges of the element.

Evaluation Copy

```
background-position: bottom 10% right 10%;
```

This code would put the image `10%` from the bottom and right edges of the element.

```
background-position: bottom 10% center;
```

This code would put the image `10%` from the bottom edge of the area specified by the element.

EVALUATION COPY: Not to be used in class.

When using offsets, you must always use two keywords, one for vertical positioning and one for horizontal positioning:

```
background-position: bottom 10%; /* not valid */
background-position: bottom 10% top 10%; /* not valid */
```

❖ 6.6.2. Coordinates

When using coordinates, the first value specified is the x-value and the second is the y-value. Both can be either length values in units or percentage values. They are relative to the top left corner of the element. For example:

```
background-position: 10px 10px;
```

Evaluation Copy

This code would put the image `10px` from the top and left edges of the element.

```
background-position: 10% 10%;
```

This code would put the image `10%` from the top and left edges of the element.

It is possible to use keywords as coordinates, as you can see below:

```
background-position: 10% bottom;
```

```
background-position: right 20px;
```

Notice that this is not an offset. Keyword offsets must have two keywords.

```
background-position: right bottom;
```

Evaluation Copy

You can see all of these examples in `Demos/background-position-styles.css` and `Demos/background-position.html`.

6.7. Creating a “hint” Class

In the following demo, we use `background-color`, `background-image`, `background-repeat`, `background-attachment`, and `background-position` to create a “hint” class for adding callout-style hints to pages:

EVALUATION COPY: Not to be used in class.

Demo 6.11: Backgrounds/Demos/hint1-styles.css

```
1.   body {
2.     background-color: hsl(0,0%,98%);
3.   }
4.
5.   .hint {
6.     border: 1px solid black;
7.     border-radius: 10px;
8.     height: 200px;
9.     overflow: hidden;
10.    width: 300px;
11.  }
12.
13.  .hint .hint-title {
14.    font-size: 1.2em;
15.    margin-bottom: 0;
16.  }
17.
18.  .hint .hint-content {
19.    background-attachment: scroll;
20.    background-color: #ffc;
21.    background-image: url(images/bulb.png);
22.    background-position: right 1.5em top 0;
23.    background-repeat: no-repeat;
24.    height: 100%;
25.    width: 85%;
26.    overflow: auto;
27.    padding: 0.5em 3.5em 0.5em 0.5em;
28.  }
```

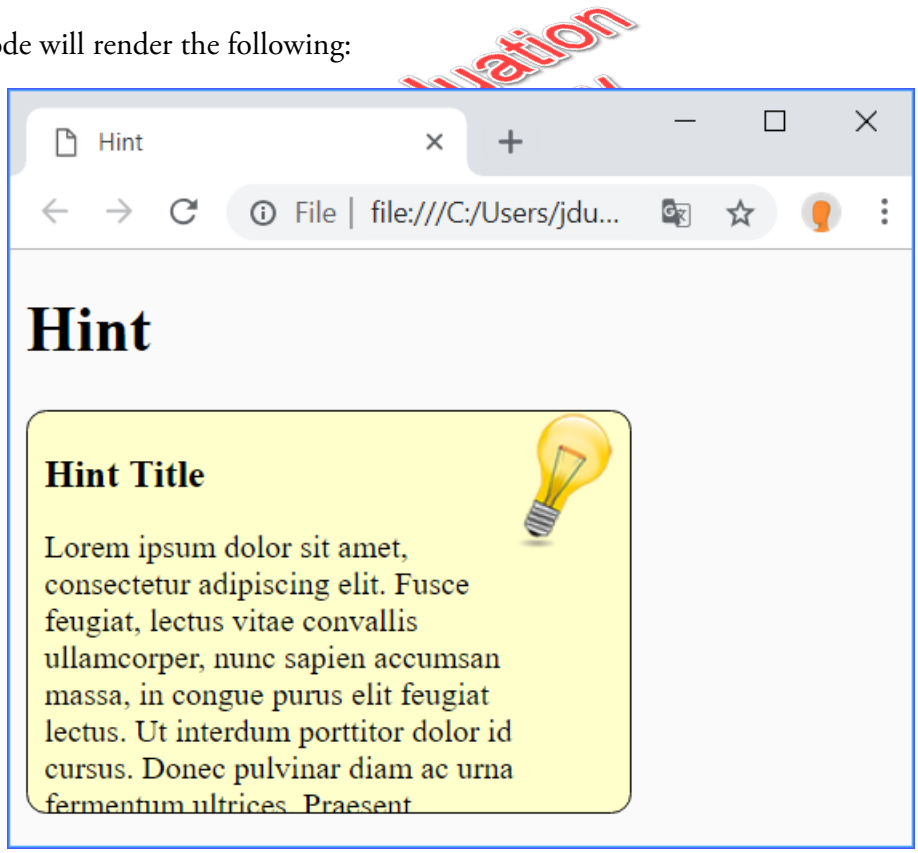
Evaluation Copy

Demo 6.12: Backgrounds/Demos/hint1.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.    <h1>Hint</h1>
12.    <div class="hint">
13.      <div class="hint-content">
14.        <h1 class="hint-title">Hint Title</h1>
15.        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
16.        Fusce feugiat, lectus vitae convallis ullamcorper, nunc sapien
17.        accumsan massa, in congue purus elit feugiat lectus. Ut interdum
     -------Lines 18 through 31 Omitted-------
32.        </p>
33.      </div>
34.    </div>
35.  </body>
36.  </html>
```

Code Explanation

The above code will render the following:



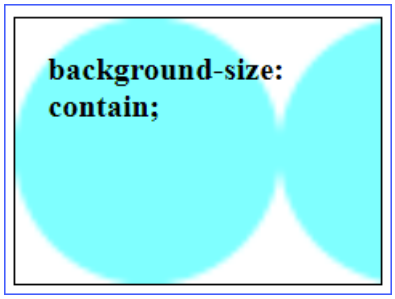
EVALUATION COPY: Not to be used in class.

Open `Backgrounds/Demos/hint1.html` in your browser and scroll the hint. Notice that the light bulb stays in the box. What CSS property would you change to make it scroll with the rest of the content?

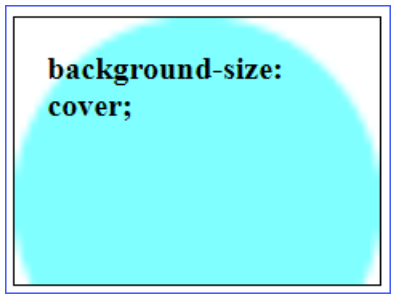
6.8. background-size

The `background-size` property specifies the size of the background image. `background-size` accepts the following possible values:

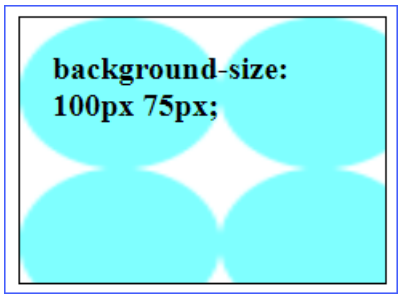
- `contain` - Scales the image as big as possible without stretching or clipping it.



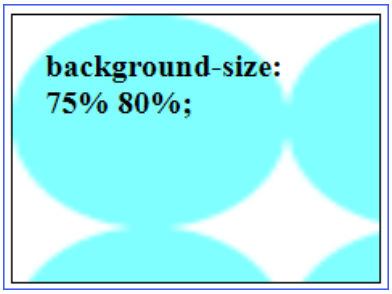
- `cover` - Scales the image as big as possible without stretching it. If the proportions of the containing element don't match the image's proportions, it will be clipped.



- A length value in units - The image stretches to the specified length. The first value is the width and the second value is the height.

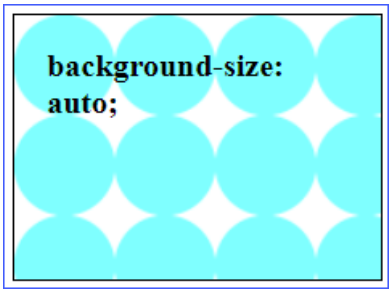


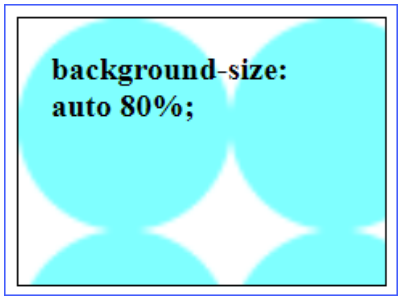
- A percentage value - The image stretches to the specified percentage. The first value is the width and the second value is the height.



- `auto` - The default value. Conserves the original size of the image, unless a length or percentage is also specified. If `auto` is used with a length or percentage value (e.g., `background-size: auto 80%;`), then the height will be `80%` and the width will expand or shrink to a value that conserves the original proportions of the image.

Evaluation Copy





EVALUATION COPY: Not to be used in class.

You can see all of these examples in `Demos/background-size-styles.css` and `Demos/background-size.html`.

6.9. background-origin

The `background-origin` property specifies the area of the element to which the background image is relative. `background-origin` takes the following values:

1. `border-box` - The background image is relative to the edge of the border.
2. `padding-box` - The default value. The background image is relative the edge of the padding.
3. `content-box` - The background image is relative to the edge the content.

Demo 6.13: Backgrounds/Demos/background-origin-styles.css

```
1.   /* The "box" class is used for display purposes only */
2.   .box {
3.     background-image: url(images/orange-background.png);
4.     background-repeat: no-repeat;
5.     border: 1rem double black;
6.     font-weight: bold;
7.     height: 100px;
8.     padding: 1rem;
9.     margin: 1rem;
10.    width: 150px;
11.  }
12.
13.  #border-box {
14.    background-origin: border-box;
15.  }
16.
17.  #padding-box {
18.    background-origin: padding-box;
19.  }
20.
21.  #content-box {
22.    background-origin: content-box;
23.  }
```

Evaluation Copy

Demo 6.14: Backgrounds/Demos/background-origin.html

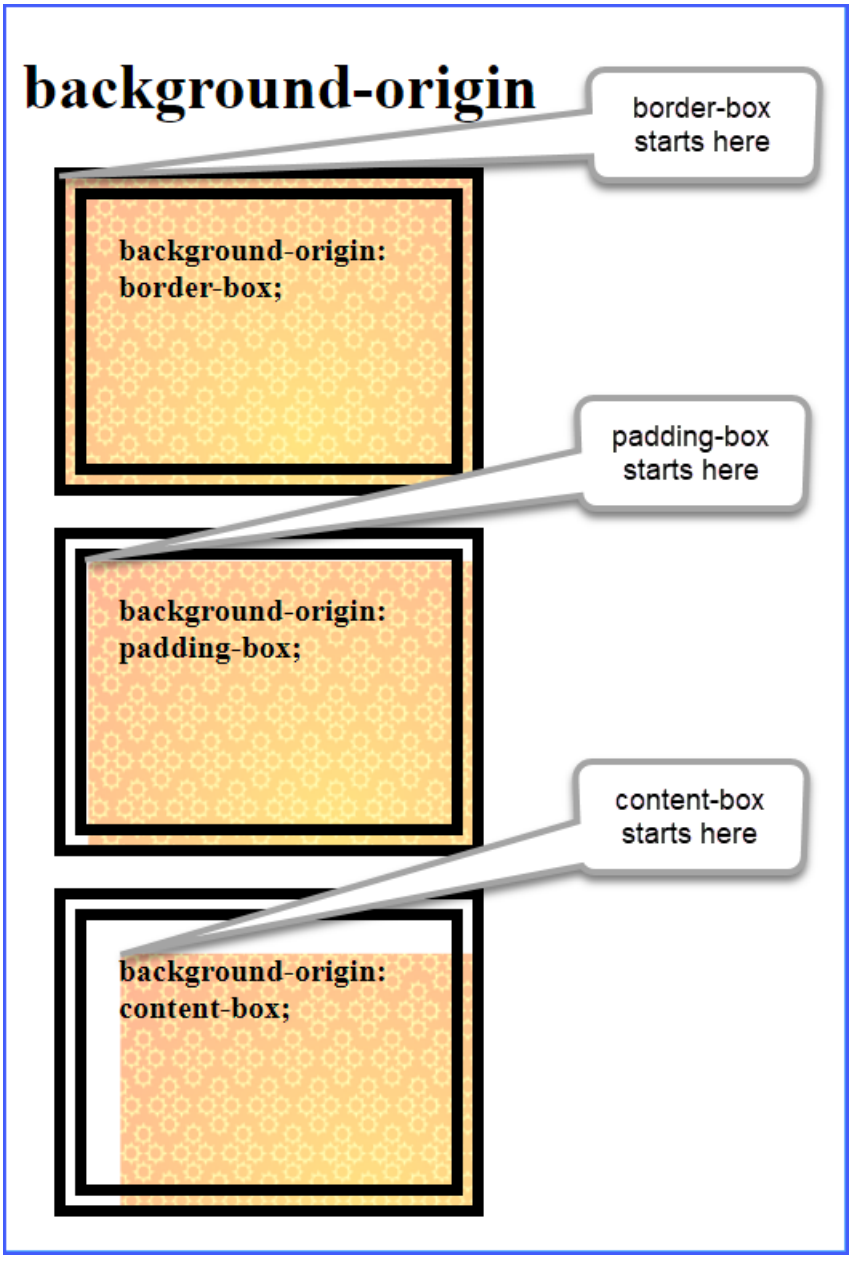
```
      -------Lines 1 through 9 Omitted-------
10.   <body class="webucator">
11.     <h1>background-origin</h1>
12.     <div id="border-box" class="box">
13.       background-origin: border-box;
14.     </div>
15.     <div id="padding-box" class="box">
16.       background-origin: padding-box;
17.     </div>
18.     <div id="content-box" class="box">
19.       background-origin: content-box;
20.     </div>
21.   </body>
22.   </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:

EVALUATION COPY: Not to be used in class.


background-origin
border-box starts here
background-origin: border-box;
padding-box starts here
background-origin: padding-box;
content-box starts here
background-origin: content-box;

6.10. background-clip

The `background-clip` property specifies the areas of the element under which the background color or image extends. `background-clip` takes the following values (the same values as `background-origin`):

1. `border-box` - The default value. The background color or image extends to the edge of the border.
2. `padding-box` - The background color or image extends to the edge of the padding.
3. `content-box` - The background color or image extends to the edge of the content.

Demo 6.15: Backgrounds/Demos/background-clip-styles.css

```
1.   /* The "box" class is used for display purposes only */
2.   .box {
3.     background-color: rgba(255, 0, 0, 50%);
4.     border: 1rem dotted black;
5.     font-weight: bold;
6.     height: 100px;
7.     padding: 1rem;
8.     margin: 1rem;
9.     width: 200px;
10.  }
11.
12.  #border-box {
13.    background-clip: border-box;
14.  }
15.
16.  #padding-box {
17.    background-clip: padding-box;
18.  }
19.
20.  #content-box {
21.    background-clip: content-box;
22.  }
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

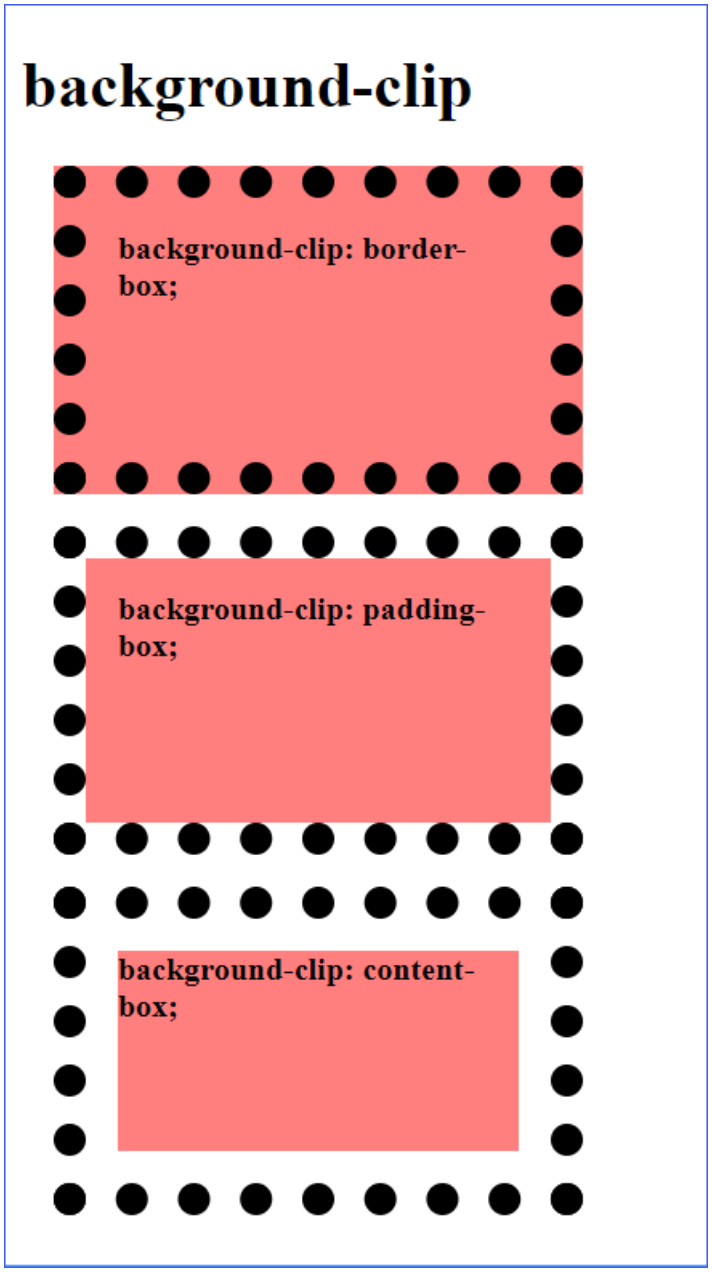
Demo 6.16: Backgrounds/Demos/background-clip.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.    <h1>background-clip</h1>
12.    <div id="border-box" class="box">
13.      background-clip: border-box;
14.    </div>
15.    <div id="padding-box" class="box">
16.      background-clip: padding-box;
17.    </div>
18.    <div id="content-box" class="box">
19.      background-clip: content-box;
20.    </div>
21.  </body>
22.  </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:



6.11. Modifying the “hint” Class

In the following demo, we use `background-size` and `background-clip` to modify our “hint” class.

EVALUATION COPY: Not to be used in class.

Demo 6.17: Backgrounds/Demos/hint2-styles.css

```
1.   body {
2.     background-color: hsl(0,0%,98%);
3.   }
4.
5.   .hint {
6.     background-color: white;
7.     border: 1px solid black;
8.     border-radius: 10px;
9.     height: 200px;
10.    padding: 1em;
11.    width: 300px;
12.  }
13.
14.  .hint .hint-title {
15.    background-image: url(images/bulb.png);
16.    background-position: right 0.5em top 0;
17.    background-repeat: no-repeat;
18.    background-size: contain;
19.    font-size: 1.2em;
20.    margin: .5em .5em 0 .5em;
21.    padding: .1em;
22.  }
23.
24.  .hint .hint-content {
25.    background-clip: padding-box;
26.    background-color: #ffc;
27.    border: double .5em rgba(0,102,255,.1);
28.    border-radius: 10px;
29.    height: 80%;
30.    overflow: auto;
31.    padding: 0 1em;
32.  }
```

Evaluation Copy

Demo 6.18: Backgrounds/Demos/hint2.html

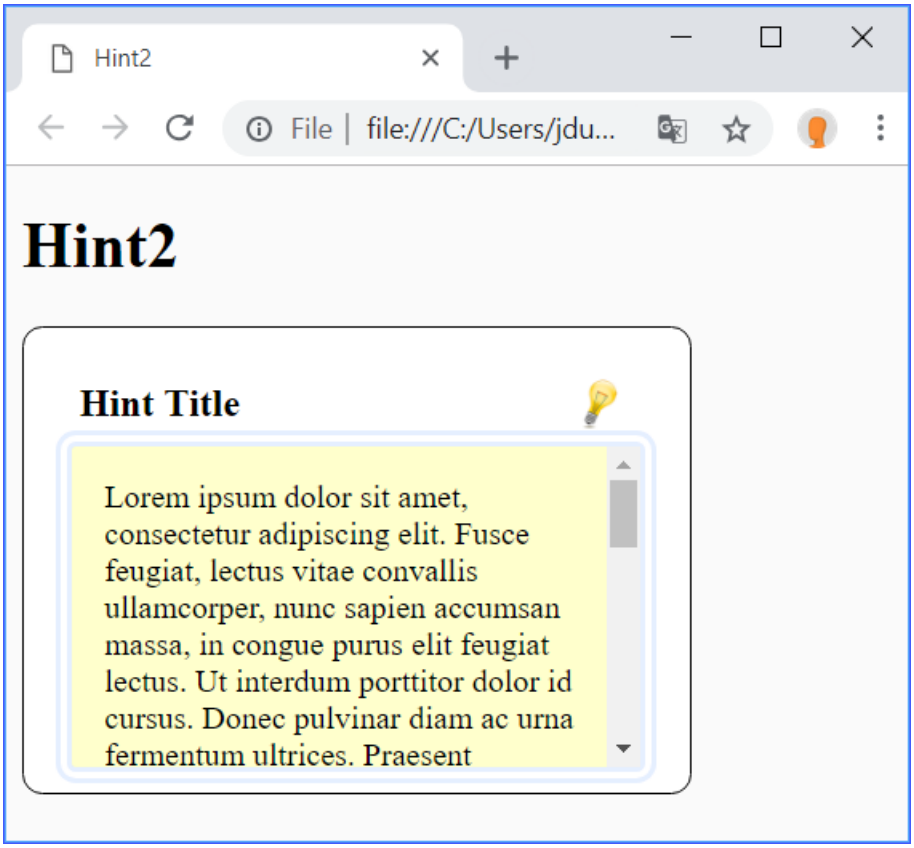
```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.    <h1>Hint2</h1>
12.    <div class="hint">
13.      <h1 class="hint-title">Hint Title</h1>
14.      <div class="hint-content">
15.        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
16.        Fusce feugiat, lectus vitae convallis ullamcorper, nunc sapien
17.        accumsan massa, in congue purus elit feugiat lectus. Ut interdum
     -------Lines 18 through 31 Omitted-------
32.        </p>
33.      </div>
34.    </div>
35.  </body>
36.  </html>
```

Evaluation Copy

Code Explanation

The above code will render the following:

EVALUATION COPY: Not to be used in class.



Things to note:

1. In the HTML, we moved the "hint-title" `h1` out of and above the "hint-content" `div`, so the heading stays present when the content is scrolled.
2. We moved the light bulb background image and associated properties from the "hint-content" to the "hint-title" class.
3. Open `Backgrounds/Demos/hint2.html` in your browser. Notice that the yellow background does not extend to the outside border. The space between the two lines in the double border is white. What property would you change to make the yellow background cover that space as well?

6.12. background

The `background` property is the shorthand property for all of the background properties covered in this lesson. The simplified syntax for `background` is shown below. The highlighted properties are required:

```
background: background-image and/or background-color
            background-attachment
            background-origin
            background-clip
            background-position/background-size
            background-repeat;
```

Some notes:

Evaluation Copy

1. You must include either a `background-image` value or `background-color` value (or both).
2. If a `background-size` value is specified, it must come directly after the `background-position` separated by a forward slash (`/`).
3. Since the `background-origin` and `background-clip` values are the same (`border-box`, `padding-box`, and `content-box`), if one value is specified, it applies to both, and if two values are specified, the first applies to `background-origin` and the second applies to `background-clip`.

Shorthand Defaults

As with any shorthand property, any unspecified values will be set to their default values.

EVALUATION COPY: Not to be used in class.

Demo 6.19: Backgrounds/Demos/background-styles.css

```
1.   /* The "box" class is used for display purposes only */
2.   .box {
3.     border: 1rem dashed black;
4.     float: left; /* allows boxes to be side by side */
5.     font-weight: bold;
6.     height: 100px;
7.     padding: 1rem;
8.     margin: 1rem;
9.     width: 200px;
10.  }
11.
12.  #div1 {
13.    background: url(images/orange-background.png) padding-box;
14.  }
15.
16.  #div2 {
17.    background: url(images/star.png) space content-box;
18.  }
19.
20.  #div3 {
21.    background: url(images/circle.png) no-repeat center/50%;
22.  }
```

Evaluation Copy

Demo 6.20: Backgrounds/Demos/background.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.    <h1>background</h1>
12.    <div id="div1" class="box">
13.      background: url(images/orange-background.png) padding-box;
14.    </div>
15.    <div id="div2" class="box">
16.      background: url(images/star.png) space content-box;
17.    </div>
18.    <div id="div3" class="box">
19.      background: url(images/circle.png) no-repeat center/50%;
20.    </div>
21.  </body>
22.  </html>
```

Code Explanation

The code above will render the following:

EVALUATION COPY: Not to be used in class.

Exercise 9: Backgrounds

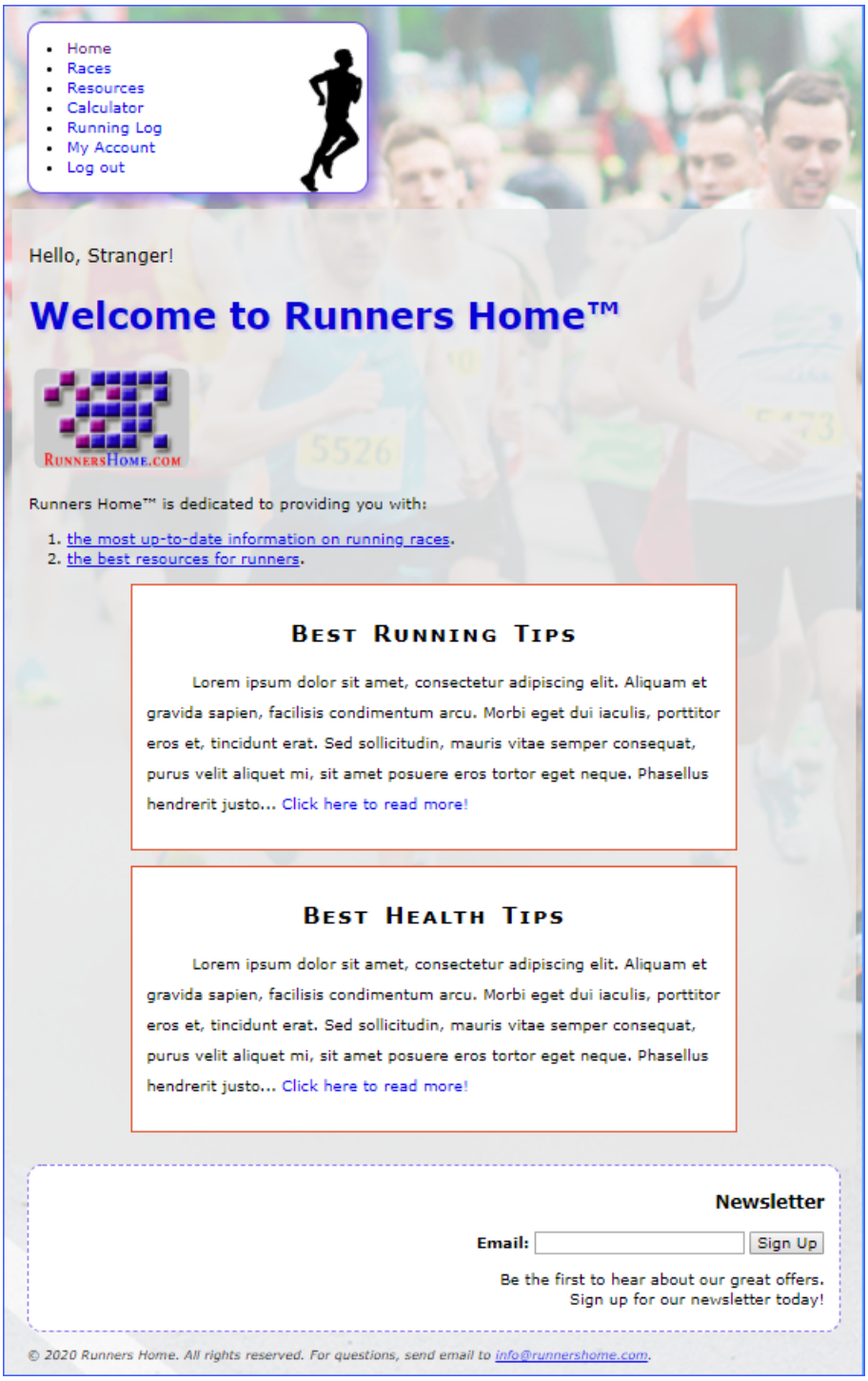
25 to 40 minutes

In this exercise, you will continue to work on the Runners Home page.

1. Open `Backgrounds/Exercises/index.html` in your editor. You will see the home page for a website called Runners Home.
2. Open `Backgrounds/Exercises/styles.css`. This is the stylesheet from the solution to the Borders, Margins, and Padding exercise. Feel free to continue working from your own stylesheet.
3. Using your new knowledge of backgrounds add some styles to this page. The object of this exercise is to practice using the background properties you just learned. Also, feel free to add or change any other properties that you know.
4. When you are done, open `index.html` in your browser to see the results. You are welcome to go back to the code and continue to work.

Evaluation Copy

You can design it however you like, or you can try to make it look something like:



The running image is in the `Exercises/images` folder. You are welcome to use other images that you download or own.

EVALUATION COPY: Not to be used in class.

Solution: Backgrounds/Solutions/styles.css

```
1.  html {
2.    font-size: 16px;
3.  }
4.
5.  body {
6.    background-image: url(images/runners.png);
7.    background-size: cover;
8.    font: 1rem/1.3 Verdana, Geneva, Tahoma, sans-serif;
9.  }
10.
11. nav {
12.   background: url(images/runner.png) no-repeat bottom right;
13.   background-color: white;
14.   border: solid rgb(110, 78, 226) 0.1rem;
15.   border-radius: 1rem;
16.   box-shadow: 0.5rem 0.5rem 1rem rgb(110, 78, 226, .5);
17.   margin: 1rem;
18.   width: 40%;
19. }
20.
21. nav a {
22.   text-decoration: none;
23. }
24.
25. main {
26.   background: rgba(232, 232, 232, 0.8);
27.   padding: 2%;
28.   max-width: 1000px;
29.   min-width: 750px;
30. }
31.
32. #welcome {
33.   font-size: larger;
34. }
35.
36. #welcome h1 {
37.   color: rgb(8, 5, 211);
38.   text-shadow: 0.2rem 0.2rem 0.1rem rgb(211, 200, 238);
39. }
40.
41. article {
42.   background: white;
43.   border: 1px ridge rgb(226, 77, 40);
44.   box-sizing: border-box;
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
45.     margin: 1em auto;
46.     padding: 1rem;
47.     max-width: 75%;
48.     text-align: center;
49.   }
50.
51.   article h2 {
52.     font-variant: small-caps;
53.     letter-spacing: 0.1em;
54.     word-spacing: 0.2em;
55.   }
56.
57.   article p {
58.     line-height: 2;
59.     text-align: left;
60.     text-indent: 3rem;
61.   }
62.
63.   .read-more {
64.     text-decoration: none;
65.   }
66.
67.   aside label {
68.     font-weight: bold;
69.   }
70.
71.   #newsletter {
72.     background: white;
73.     border: 1px dashed rgb(110, 78, 226);
74.     border-radius: 1em;
75.     margin: 0 1rem;
76.     padding: .5rem 1rem;
77.     text-align: right;
78.   }
79.
80.   footer {
81.     font-size: smaller;
82.     font-style: italic;
83.     margin: 1rem;
84.   }
```

Evaluation Copy

Conclusion

In this lesson, you have learned to use background properties.

EVALUATION COPY: Not to be used in class.

LESSON 7
Display and Visibility

Topics Covered

☑ `display`

☑ `visibility`

Introduction

In this lesson, you will learn to use the `display` and `visibility` properties to show and hide content.

7.1. display

Evaluation Copy

The `display` property is used to determine if and how an element appears. The most common values are shown below:

1. `block`
2. `inline`
3. `none`

For a full list of `display` values, see `https://developer.mozilla.org/en-US/docs/Web/CSS/display`.

We use the `display` property to control how an element displays on the page - to control its rendering box. The default value for most elements is either:

- `block`, for `<p>`, `<div>`, `<form>`, `<h1>`, and others;
- `inline`, for `<span>`, `<img>`, `<a>`, and others.

We can set `display` to `none` to hide an element. The `display` property is often used to:

1. Hide and show elements by using JavaScript to toggle its value between block and none.

2. To show elements hide elements on a particular media. For example, you might want to hide advertisements on the printed version of a page.

tables and lists

It is possible to use `display` to make non-table elements act like table elements and non-list elements act like list elements, but it is always better to use actual table and list elements to create tables and lists.

The following example shows how the `display` property can be used to turn links into block elements:

Demo 7.1: DisplayAndVisibility/Demos/link-buttons-styles.css

Evaluation Copy

```
1.   a.button-link {
2.     background-color: #eaf1dd;
3.     border-color: silver;
4.     border-radius: 0.5rem;
5.     border-style: none solid solid none;
6.     border-width: 0 0.2rem 0.2rem 0;
7.     color: #060;
8.     display: block;
9.     font-family: Verdana, Geneva, sans-serif;
10.    font-size: 1.5rem;
11.    margin: .5rem;
12.    padding: .5rem .2rem;
13.    text-decoration: none;
14.    width: 300px;
15.  }
```

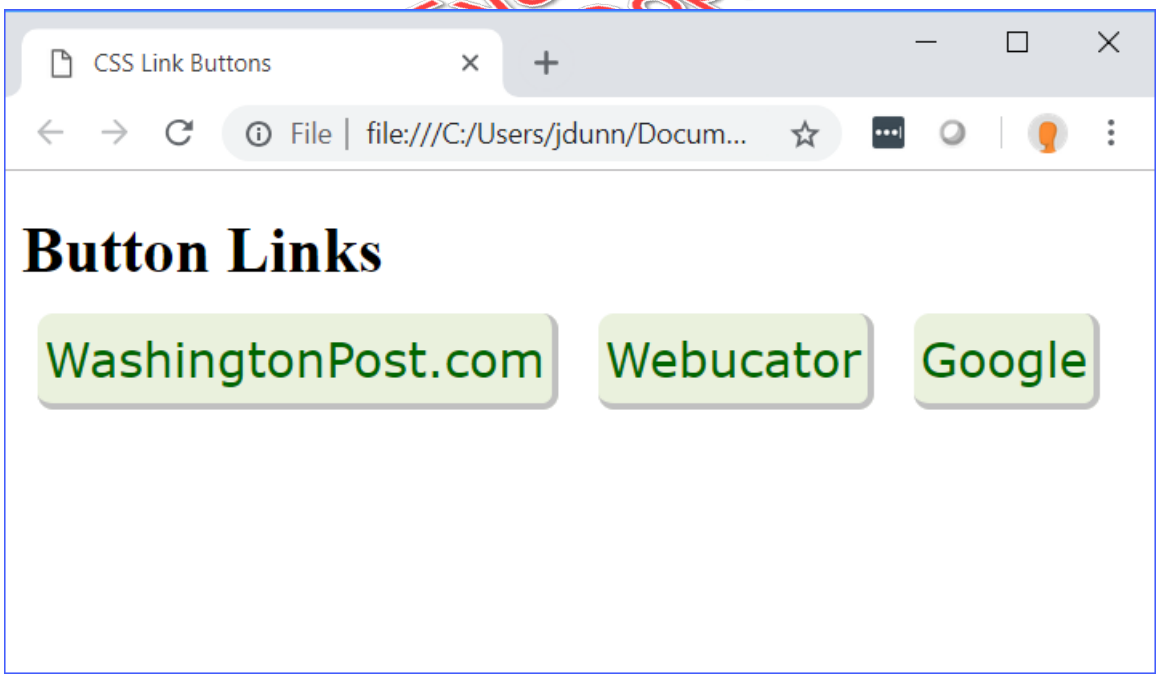

EVALUATION COPY: Not to be used in class.

Demo 7.2: DisplayAndVisibility/Demos/link-buttons.html

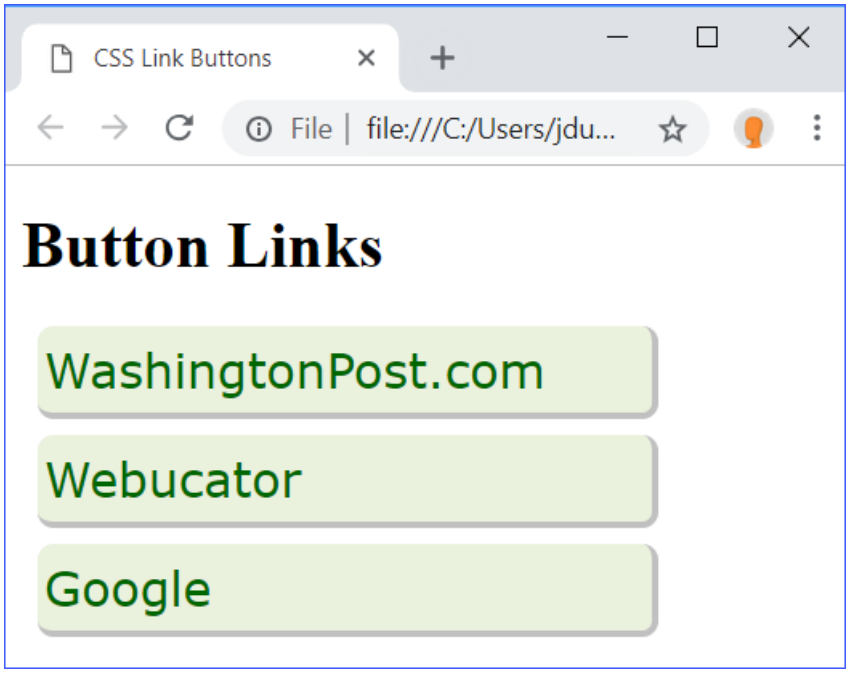
```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.  <h1>Button Links</h1>
12.  <div>
13.    <a class="button-link" href="https://www.washingtonpost.com">
14.      WashingtonPost.com
15.    </a>
16.    <a class="button-link" href="https://www.webucator.com">
17.      Webucator
18.    </a>
19.    <a class="button-link" href="https://www.google.com">
20.      Google
21.    </a>
22.  </div>
23.  </body>
24.  </html>
```

Code Explanation

If we didn't set `display` to `block`, the page would look like this:



But with `display` set to `block`, it looks like this:



Evaluation Copy

7.2. visibility

The `visibility` property is used to hide or show an element without affecting the overall layout of the page. The values are listed below:

1. `visible`
2. `hidden`

collapse

`visibility` also has a third value: `collapse`, which is not as well supported, and is just used on `table` elements. To learn more about `collapse`, see `https://developer.mozilla.org/en-US/docs/Web/CSS/visibility`.

We can set `visibility` to `hidden` to hide an element. Like with `display`, we can use this property with JavaScript to hide and show an element.

EVALUATION COPY: Not to be used in class.

Note that `visibility: hidden` is different from `display: none`: `visibility: hidden` will leave space for the hidden element, so that the layout is unaffected, and `display: none` will remove the element, changing the space of the layout. See the following demo:

Demo 7.3: DisplayAndVisibility/Demos/none-vs-hidden-styles.css

```
1.   /* The "box" class is used for display purposes only */
2.   .box {
3.     background: silver;
4.     border: 1px black solid;
5.     height: 50px;
6.     margin: 1rem;
7.     padding: 1rem;
8.     width: 50px;
9.   }
10.
11.  #none {
12.    display: none;
13.  }
14.
15.  #hidden {
16.    visibility: hidden;
17.  }
```

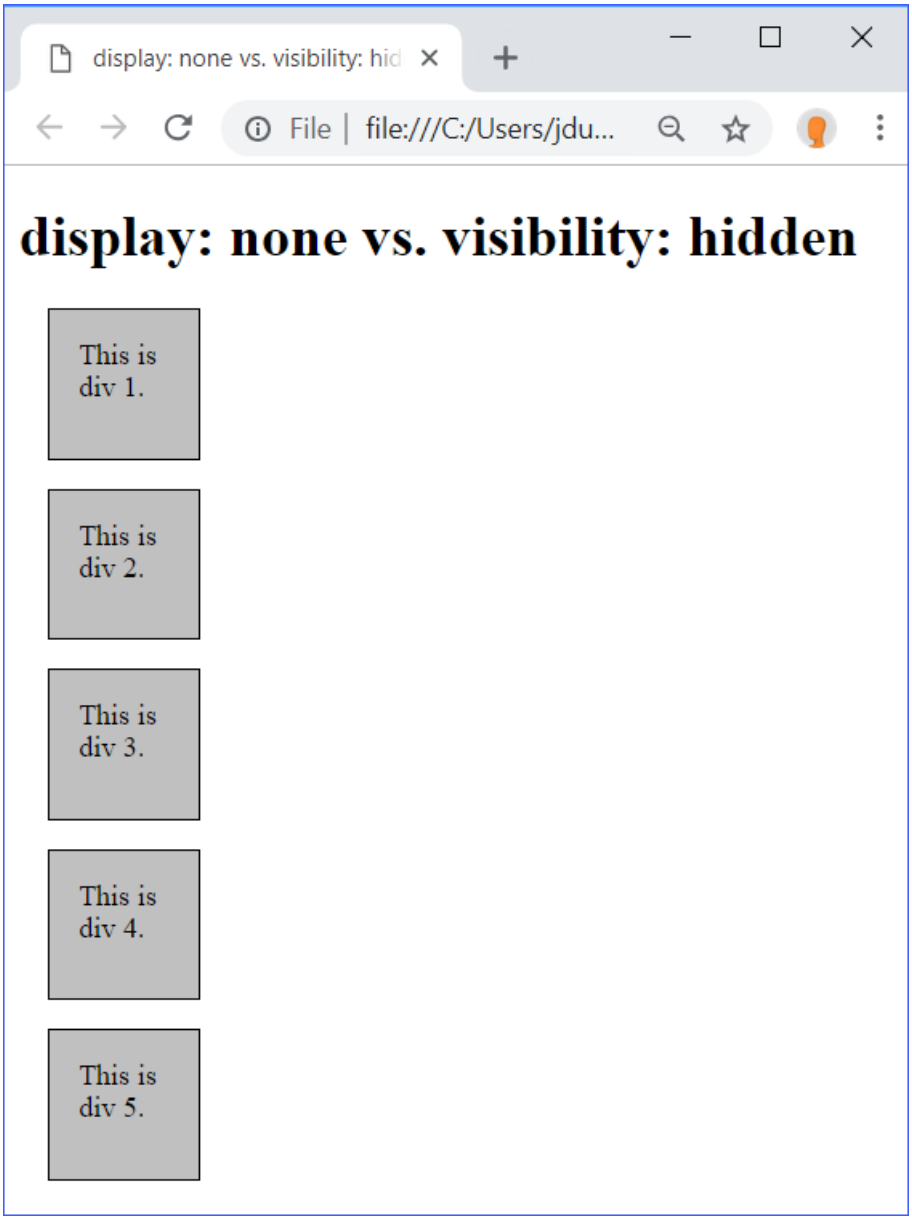
Evaluation Copy

Demo 7.4: DisplayAndVisibility/Demos/none-vs-hidden.html

```
       -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.  <h1>display: none vs. visibility: hidden</h1>
12.    <div class="box">This is div 1.</div>
13.    <div id="none" class="box">This is div 2.</div>
14.    <div class="box">This is div 3.</div>
15.    <div id="hidden" class="box">This is div 4.</div>
16.    <div class="box">This is div 5.</div>
17.  </body>
18.  </html>
```

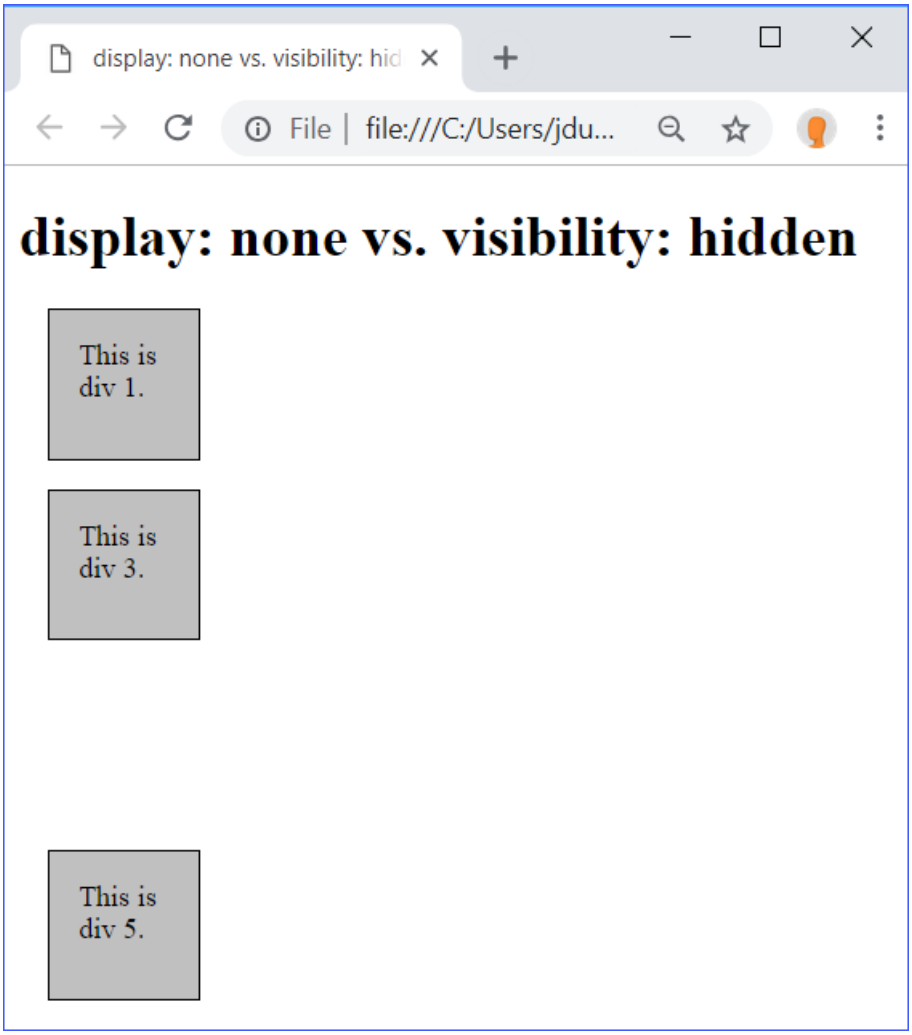
Code Explanation

Without the `display` or `visibility` declarations, the code would render like this:



With the `display` and `visibility` declarations, the code would render like this:

EVALUATION COPY: Not to be used in class.



Notice that no space is left where div 2 was, while there is still a gap where div 4 used to be.

Conclusion

In this lesson, you have learned to use the `display` and `visibility` properties.

EVALUATION COPY: Not to be used in class.

LESSON 8

Pseudo-classes and Pseudo-elements

Topics Covered

☑ Pseudo-classes.

☑ Pseudo-elements.

Introduction

In this lesson, you will learn to use pseudo-classes and pseudo-elements to target a specific state or part of a selected element or multiple selected elements.

Evaluation Copy

8.1. Pseudo-classes

Pseudo-classes are keywords that can be added to a selector to target a specific state of the selected element(s). For example, `:hover` can be used to target the style of an element when the mouse is hovering over it.

Below is a table of pseudo-classes:

Pseudo-classes

Pseudo-class	How it works
`:active`	Selects an element being activated by a user (e.g., clicked).
`:empty`	Selects an element that has no content (including whitespace).
`:first-child`	Selects the first element in a group of siblings.
`:first-of-type`	Selects the first element of its type in a group of siblings.
`:hover`	Selects an element that is being hovered over with the mouse.
`:lang()`	Selects elements that have content in the language specified in the parentheses, meaning that they have a `lang` attribute specifying the language. See a list of ISO 639-1 language codes on Wikipedia.[29]
`:last-child`	Selects the last element in a group of siblings.
`:last-of-type`	Selects the last element of its type in a group of siblings.
`:link`	Selects a link element whose link has not been visited.
`:not()`	Selects elements that are not specified in the parentheses.
`:nth-child()`	Selects elements based on their position among their siblings.
`:nth-last-child()`	Works exactly like `:nth-child()`, but starts from the last child and goes backward.
`:nth-last-of-type()`	Selects elements of certain type by position starting from the last element and going backward.
`:nth-of-type()`	Selects elements of certain type by position.
`:only-child`	Selects elements that do not have any siblings.
`:only-of-type`	Selects elements that do not have any siblings of the same type.
`:root`	Selects the root element of the document. In HTML, this is always the `html` element.
`:target`	Selects the element that is the current target, meaning that the URL is targeting that element's `id` (e.g.,`index.html#article1`).
`:visited`	Selects a link element whose link has been visited. The only allowable properties are: `color`, `background-color`, `border-color`, and `outline-color`.

Evaluation Copy

29. https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes

EVALUATION COPY: Not to be used in class.

Many of these pseudo-classes are rarely used. We will look at the most useful ones below.

8.2. Styling Links with Pseudo-classes

The most commonly used pseudo-classes when styling links are:

1. `:visited`
2. `:hover`
3. `:active`

We will also look at the `:target` pseudo-class. Although it is not used to style links, links determine what `:target` selects.

:link

Note that our demo does not include the `:link` pseudo-class. This is because you can use a simple `a` rule instead of `a:link`. If you do include both an `a` rule and an `a:link` rule the `a:link` rule will take precedence because it has a higher specificity.

Evaluation Copy

First, let's look at how most browsers style links by default:

1. Links are underlined.
2. Links that have not been visited are blue.
3. Links that have been visited are purple.
4. Hovering over/clicking a link makes the mouse a pointer.
5. Active links (links being clicked) are red.

Changing Link Styles

Be careful not to change link styles in a way that disguises the fact that they are links.

The demo below includes examples for how to style links.

Demo 8.1: PseudoClassesAndElements/Demos/links-styles.css

```
1.  body {
2.    background: aliceblue;
3.  }
4.
5.  header {
6.    background: gray;
7.    border: black solid 1px;
8.    border-radius: 1rem;
9.    height: 160px;
10.   margin: 0 auto 0.5rem;
11.   padding: 0.5rem;
12.   width: 410px;
13. }
14.
15. header h2 {
16.   padding-left: 0.5rem;
17.   text-decoration: underline;
18. }
19.
20. #outside-links {
21.   background: white;
22.   border: black solid 1px;
23.   float: left;
24.   margin-right: 0.2rem;
25.   width: 200px;
26. }
27.
28. #article-links {
29.   background: white;
30.   border: black solid 1px;
31.   float: left;
32.   width: 200px;
33. }
34.
35. main {
36.   background: white;
37.   border: gray solid;
38.   border-radius: 0.5rem;
39.   height: 370px;
40.   margin: auto;
41.   width: 500px;
42. }
43.
44. article {
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
45.     background: #ccc;
46.     border: solid black 1px;
47.     float: left;
48.     margin: 0.5rem;
49.     padding: 1rem;
50.     width: 200px;
51.   }
52.
53.   article h2 {
54.     font-variant: small-caps;
55.     margin: 0;
56.   }
57.
58.   a,
59.   a:visited {
60.     text-decoration: none;
61.     color: blue;
62.   }
63.
64.   a:hover {
65.     text-decoration: underline;
66.   }
67.
68.   a:active {
69.     color: gray;
70.   }
71.
72.   :target h2 {
73.     font-style: italic;
74.     text-decoration: underline;
75.   }
```

Evaluation Copy

Demo 8.2: PseudoClassesAndElements/Demos/links.html

```
      -------Lines 1 through 9 Omitted-------
10.   <body class="webucator">
11.   <h1>Links and Pseudo-classes</h1>
12.   <header>
13.     <aside id="outside-links">
14.       <h2>Links</h2>
15.       <ol>
16.         <li>
17.           <a href="https://www.webucator.com/">
18.           Webucator</a>
19.         </li>
20.         <li>
21.           <a href="https://www.google.com/">
22.           Google</a>
23.         </li>
24.         <li>
25.           <a href="https://www.washingtonpost.com/">
26.             Washington Post</a>
27.         </li>
28.         <li><a href="https://www.wikipedia.org/">Wikipedia</a></li>
29.       </ol>
30.     </aside>
31.     <nav id="article-links">
32.       <h2>Articles</h2>
33.       <ol>
34.         <li><a href="#article1">Article 1</a></li>
35.         <li><a href="#article2">Article 2</a></li>
36.         <li><a href="#article3">Article 3</a></li>
37.         <li><a href="#article4">Article 4</a></li>
38.       </ol>
39.     </nav>
40.   </header>
41.   <main>
42.     <article id="article1">
43.       <h2>Article 1</h2>
44.       <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
45.         Nulla non gravida massa, sit amet commodo nisi...</p>
46.     </article>
47.     <article id="article2">
48.       <h2>Article 2</h2>
49.       <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
50.           Nulla non gravida massa, sit amet commodo nisi...</p>
51.     </article>
52.     <article id="article3">
```

Evaluation Copy

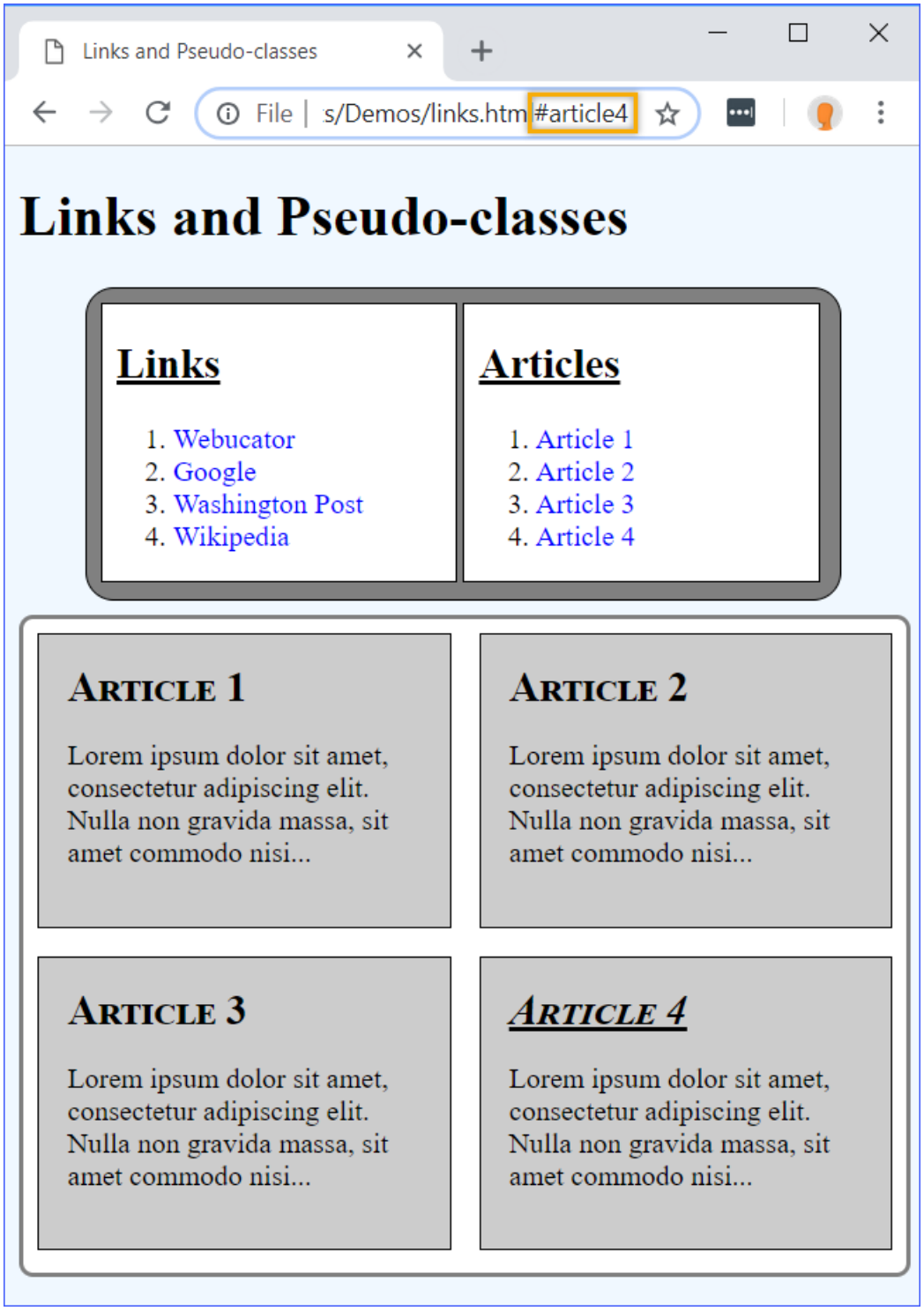
EVALUATION COPY: Not to be used in class.

```
53.       <h2>Article 3</h2>
54.       <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
55.           Nulla non gravida massa, sit amet commodo nisi...</p>
56.     </article>
57.     <article id="article4">
58.       <h2>Article 4</h2>
59.       <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
60.           Nulla non gravida massa, sit amet commodo nisi...</p>
61.     </article>
62.   </main>
63.   </body>
64.   </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:



Some things to note:

EVALUATION COPY: Not to be used in class.

1. We removed the underline from the links.
2. We kept the signature blue color of the links.
3. We made visited links appear exactly like links that haven't been visited.
4. We added the underline back when the links are hovered over.
5. We made the links turn gray when they are clicked.
6. We underlined and italicized the `h2` elements within the target. Note that `article4` is the current target.
7. In order to really see what's going on with the links, you will need to open `links.html` in the browser and interact with them.

8.3. Styling Tables and Articles with Pseudo-classes

The following demos show how to use these pseudo-classes:

Evaluation Copy

1. `:first-child`
2. `:nth-child()`
3. `:first-of-type`
4. `:last-of-type`

❖ 8.3.1. :nth-child()

The `:nth-child()` pseudo-class is a powerful selector that selects an element or elements based on their position. `:nth-child()` accepts several values:

1. A positive integer (e.g., 1, 2, 3, etc.) to select one element in a specific position.
2. The keywords `even` and `odd` to select either all the even elements in a selection or all the odd elements in a selection.
3. A function in the form of `An+B` where `A` and `B` are integers and `n` is a variable. For example, `#my-list li:nth-child(-n+3)` would select the first three `li` elements in `my-list` (-0+3=3, -1+3=2, -2+3=1). Negative values are not valid.

The demo below shows how pseudo-classes can be used to style tables:

Demo 8.3: PseudoClassesAndElements/Demos/table-pseudo-classes-styles.css

```
1.   body {
2.     background: aliceblue;
3.   }
4.
5.   table,
6.   th,
7.   td {
8.     border: solid black 1px;
9.     border-collapse: collapse; /* collapses
10.                        the separate borders into single borders
11.                        between the table elements */
12.    padding: 0.2rem;
13.  }
14.
15.  table {
16.    margin: auto;
17.    background: white;
18.  }
19.
20.  caption {
21.    font-size: 1.5rem;
22.    font-weight: bold;
23.    text-decoration: underline;
24.  }
25.
26.  thead {
27.    background: rgb(75, 75, 75);
28.    color: white;
29.    text-decoration: underline;
30.  }
31.
32.  span.title {
33.    font-style: italic;
34.  }
35.
36.  tfoot {
37.    background: rgb(75, 75, 75);
38.    color: white;
39.  }
40.
41.  tbody td:first-child {
42.    font-style: italic;
43.  }
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
44.
45.   tbody tr:nth-child(odd) {
46.     background-color: #ccc;
47.   }
```

Evaluation Copy

Demo 8.4: PseudoClassesAndElements/Demos/table-pseudo-classes.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.  <h1>Using Pseudo-classes on a Table</h1>
12.  <table>
13.    <caption>Great Books You Should Read</caption>
14.    <thead>
15.      <tr>
16.        <th>Title</th>
17.        <th>Author</th>
18.        <th>Genre</th>
19.      </tr>
20.    </thead>
21.    <tbody>
22.      <tr>
23.        <td>Harry Potter and the Sorcerer's Stone*</td>
24.        <td>J. K. Rowling</td>
25.        <td>Fantasy</td>
26.      </tr>
27.      <tr>
28.        <td>To Kill a Mockingbird</td>
29.        <td>Harper Lee</td>
30.        <td>Novel</td>
31.      </tr>
32.      <tr>
33.        <td>Pride and Prejudice</td>
34.        <td>Jane Austen</td>
35.        <td>Romance</td>
36.      </tr>
37.      <tr>
38.        <td>The Diary of Anne Frank</td>
39.        <td>Anne Frank</td>
40.        <td>Non-fiction</td>
41.      </tr>
42.      <tr>
43.        <td>1984</td>
44.        <td>George Orwell</td>
45.        <td>Dystopia</td>
46.      </tr>
47.      <tr>
48.        <td>The Hunger Games</td>
49.        <td>Suzanne Collins</td>
50.        <td>Dystopia</td>
```

Evaluation Copy

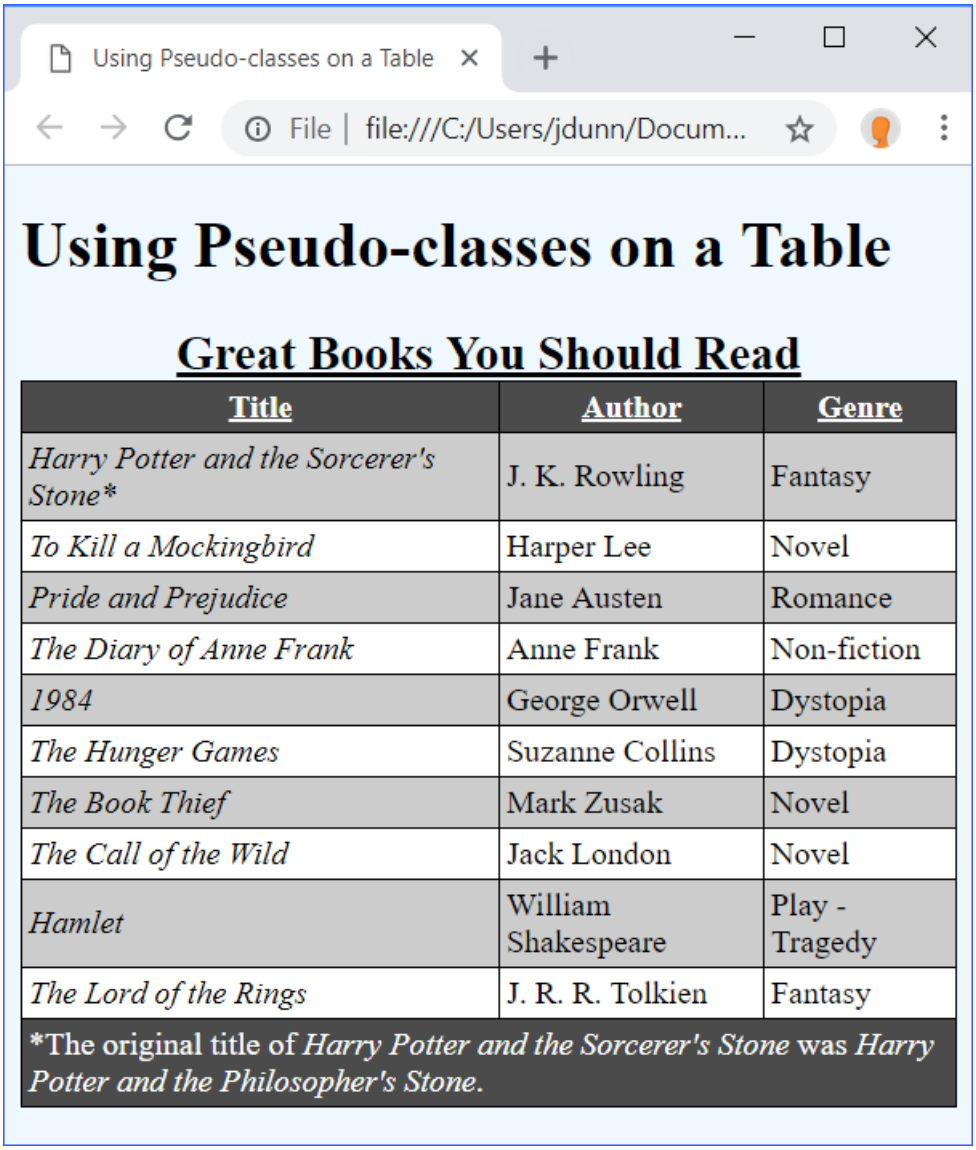
EVALUATION COPY: Not to be used in class.

```
      </tr>
      <tr>
        <td>The Book Thief</td>
        <td>Mark Zusak</td>
        <td>Novel</td>
      </tr>
      <tr>
        <td>The Call of the Wild</td>
        <td>Jack London</td>
        <td>Novel</td>
      </tr>
      <tr>
        <td>Hamlet</td>
        <td>William Shakespeare</td>
        <td>Play - Tragedy</td>
      </tr>
      <tr>
        <td>The Lord of the Rings</td>
        <td>J. R. R. Tolkien</td>
        <td>Fantasy</td>
      </tr>
    </tbody>
    <tfoot>
      <tr>
        <td colspan="3" class="info">*The original title of
          <span class="title">Harry Potter and
            the Sorcerer's Stone</span> was
          <span class="title">Harry Potter and
            the Philosopher's Stone</span>.
        </td>
      </tr>
    </tfoot>
  </table>
  </body>
  </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:



Some things to note:

1. There are a lot of rules that don't use pseudo-classes. These rules are used to create the display of the demo.
2. `:first-child` is used to select the book titles and make them italic.
3. `:nth-child()` is used to make the odd `tr` elements in the `table` gray.

Below is a demo showing how pseudo-classes can be used to help style articles.

EVALUATION COPY: Not to be used in class.

Demo 8.5: PseudoClassesAndElements/Demos/articles-pseudo-classes-styles.css

```
1.  body {
2.    background: aliceblue;
3.  }
4.
5.  header {
6.    background: gray;
7.    border: black solid 1px;
8.    border-radius: 1rem;
9.    margin: auto;
10.   width: 50%;
11. }
12.
13. a,
14. a:visited {
15.   color: blue;
16. }
17.
18. main {
19.   background: white;
20.   margin: auto;
21.   width: 75%;
22. }
23.
24. article {
25.   border: black solid 1px;
26.   border-radius: 0.5rem;
27.   padding: 0.2rem 1rem 0.5rem;
28.   margin: 0.5rem auto;
29. }
30.
31. article p:first-of-type {
32.   font-size: larger;
33. }
34.
35. article p:last-of-type {
36.   font-size: smaller;
37.   font-style: italic;
38. }
```

Evaluation Copy

Demo 8.6: PseudoClassesAndElements/Demos/articles-pseudo-classes.html

```
      -------Lines 1 through 9 Omitted-------
10.   <body class="webucator">
11.   <h1>Using Pseudo-classes to Style Some Articles</h1>
12.   <header>
13.     <nav>
14.       <ol>
15.         <li><a href="#article1">Article 1</a></li>
16.         <li><a href="#article2">Article 2</a></li>
17.         <li><a href="#article3">Article 3</a></li>
18.       </ol>
19.     </nav>
20.   </header>
21.   <main>
22.     <article id="article1">
23.       <h2>Article 1</h2>
24.       <p>
25.         This is the beginning of article 1. Lorem ipsum dolor
26.         sit amet, consectetur adipiscing elit.
27.         Curabitur tristique odio ac sem congue luctus.
28.         Donec id mauris porttitor,
      -------Lines 29 through 33 Omitted-------
34.       </p>
35.       <p>
36.         Proin dolor turpis, tincidunt ac
37.         consequat nec, luctus in justo.
38.         Mauris nec interdum ligula.
      -------Lines 39 through 42 Omitted-------
43.       </p>
44.       <p>Written by So-and-so.</p>
45.     </article>
46.     <article id="article2">
47.       <h2>Article 2</h2>
48.       <p>
49.         This is the beginning of article 1. Lorem ipsum dolor
50.         sit amet, consectetur adipiscing elit.
51.         Curabitur tristique odio ac sem congue luctus.
52.         Donec id mauris porttitor,
      -------Lines 53 through 57 Omitted-------
58.       </p>
59.       <p>
60.         Proin dolor turpis, tincidunt ac
61.         consequat nec, luctus in justo.
```

Evaluation Copy

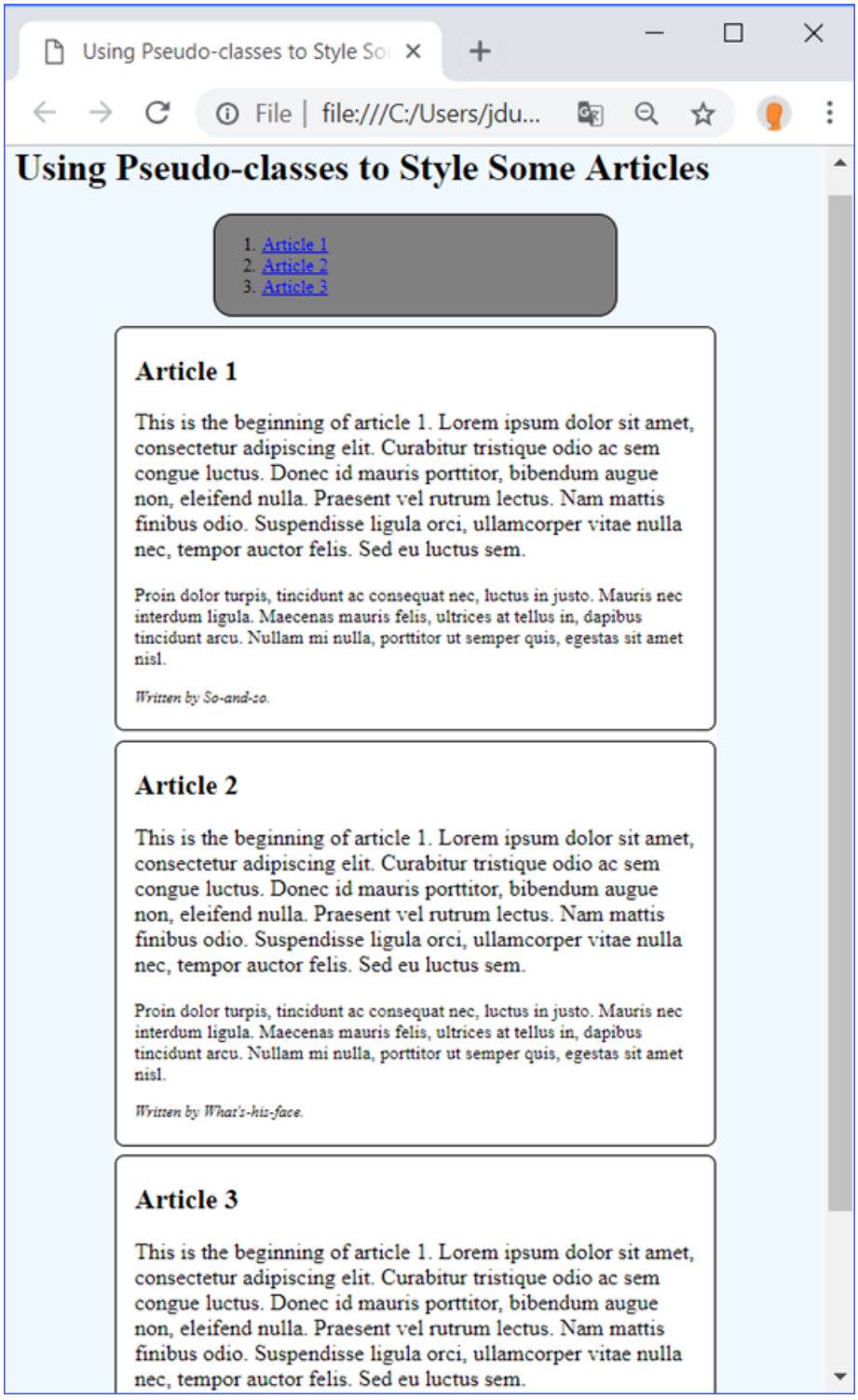
EVALUATION COPY: Not to be used in class.

```
62.         Mauris nec interdum ligula.
      -------Lines 63 through 66 Omitted-------
67.       </p>
68.       <p>Written by What's-his-face.</p>
69.     </article>
70.     <article id="article3">
71.       <h2>Article 3</h2>
72.       <p>
73.         This is the beginning of article 1. Lorem ipsum dolor
74.         sit amet, consectetur adipiscing elit.
75.         Curabitur tristique odio ac sem congue luctus.
76.         Donec id mauris porttitor,
      -------Lines 77 through 81 Omitted-------
82.       </p>
83.       <p>
84.         Proin dolor turpis, tincidunt ac
85.         consequat nec, luctus in justo.
86.         Mauris nec interdum ligula.
      -------Lines 87 through 90 Omitted-------
91.       </p>
92.       <p>Written by You-know-who.</p>
93.     </article>
94.   </main>
95.   </body>
96.   </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:



Some things to note:

EVALUATION COPY: Not to be used in class.

1. There are a lot of rules that don't use pseudo-classes. These rules are used to create the display of the demo.
2. `:first-of-type` is used to select the first `p` elements in each `article` and make them larger.
3. `:last-of-type` is used to select the last `p` elements in each `article` to make them smaller and italic.

8.4. Pseudo-elements

Pseudo-elements are keywords that can be added to a selector to target a specific part of the selected element(s). For example, `::first-letter` can be used to target the first letter of the text within an element.

Below is a table of pseudo-elements:

Pseudo-elements

Pseudo-class	How it works
`::after`	Creates a pseudo-element that is the last child of the selected element. This pseudo-element can be given content with the `content` property[30].
`::before`	Creates a pseudo-element that is the first child of the selected element. This pseudo-element can be given content with the `content` property.
`::first-letter`	Selects the first letter of the first line of text in the selected element(s).
`::first-line`	Selects the first line of text in the selected element(s).
`::selection`	Selects the content that has been highlighted by the user with the mouse. The properties that can be used with `::selection` are very limited[31].

Evaluation Copy

Two colons vs. one colon

You may see pseudo-elements written with just one colon (e.g., `:first-letter`), which is the old syntax. The new syntax is to use two colons to distinguish pseudo-elements from pseudo-classes. While browsers support both, you should use the new syntax.

30. https://developer.mozilla.org/en-US/docs/Web/CSS/content
31. https://developer.mozilla.org/en-US/docs/Web/CSS/::selection

❖ 8.4.1. content

The `content` property is used with `::before` and `::after`. It can be used to add:

1. A string (i.e., text).
2. An image as a URL value. Note that the image is displayed in its original size and cannot be manipulated in any way.
3. A counter, generally a number, using the CSS `counter()` or `counters()` function[32].
4. A string value of an element's attribute using `attr(x)` where `x` is the element's attribute (e.g., `attr(id)`).

Evaluation Copy

8.5. Using Pseudo-elements

Like pseudo-classes, pseudo-elements are infrequently used, but there are cases where they can be useful, like in the demo below:

32. https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Lists_and_Counters/Using_CSS_counters

EVALUATION COPY: Not to be used in class.

Demo 8.7: PseudoClassesAndElements/Demos/pseudo-elements-styles.css

```
1.  body {
2.    background: aliceblue;
3.  }
4.
5.  header {
6.    background: gray;
7.    border: black solid 1px;
8.    border-radius: 1rem;
9.    margin: auto;
10.   width: 200px;
11. }
12.
13. main {
14.   background: white;
15.   border: black solid 1px;
16.   margin: 0.5rem auto;
17.   width: 400px;
18. }
19.
20. section {
21.   margin: auto;
22.   width: 75%;
23. }
24.
25. a,
26. a:visited {
27.   color: blue;
28. }
29.
30. aside {
31.   margin: 0.5rem auto;
32.   width: 75%;
33. }
34.
35. .bio {
36.   font-style: italic;
37. }
38.
39. :target h2::before {
40.   content: "\2192";
41. }
42.
43. section p:first-of-type::first-line {
```

Evaluation Copy

```
44.     font-style: italic;
45.   }
46.
47.   section p:first-of-type::first-letter {
48.     font-size: xx-large;
49.   }
50.
51.   .note::after {
52.     content:"*"
53.   }
54.
55.   .bio::before {
56.     content:"*";
57.   }
```

EVALUATION COPY: Not to be used in class.

Demo 8.8: PseudoClassesAndElements/Demos/pseudo-elements.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.  <h1>Using Pseudo-elements</h1>
12.  <header>
13.    <nav>
14.      <ol>
15.        <li><a href="#chapter1">Chapter 1</a></li>
16.        <li><a href="#chapter2">Chapter 2</a></li>
17.        <li><a href="#acknowledgements">Acknowledgements</a></li>
18.      </ol>
19.    </nav>
20.  </header>
21.  <main>
22.    <section id="chapter1">
23.      <h2>Chapter 1</h2>
24.      <p>
25.        This is the beginning of chapter 1. Lorem ipsum
26.        dolor sit amet, consectetur adipiscing elit.
27.        Curabitur tristique odio ac sem congue luctus.
28.        Donec id mauris porttitor,
     -------Lines 29 through 33 Omitted-------
34.      </p>
35.      <p>
36.        Proin dolor turpis, tincidunt ac
37.        consequat nec, luctus in justo.
38.        Mauris nec interdum ligula.
     -------Lines 39 through 42 Omitted-------
43.      </p>
44.    </section>
45.    <section id="chapter2">
46.      <h2>Chapter 2</h2>
47.      <p>
48.        This is the beginning of chapter 2. Lorem ipsum dolor
49.        sit amet, consectetur adipiscing elit.
50.        Curabitur tristique odio ac sem congue luctus.
51.        Donec id mauris porttitor, bibendum augue non,
     -------Lines 52 through 55 Omitted-------
56.      </p>
57.      <p>
58.        Proin dolor turpis, tincidunt ac
59.        consequat nec, luctus in justo.
60.        Mauris nec interdum ligula.
     -------Lines 61 through 64 Omitted-------
```

Evaluation Copy

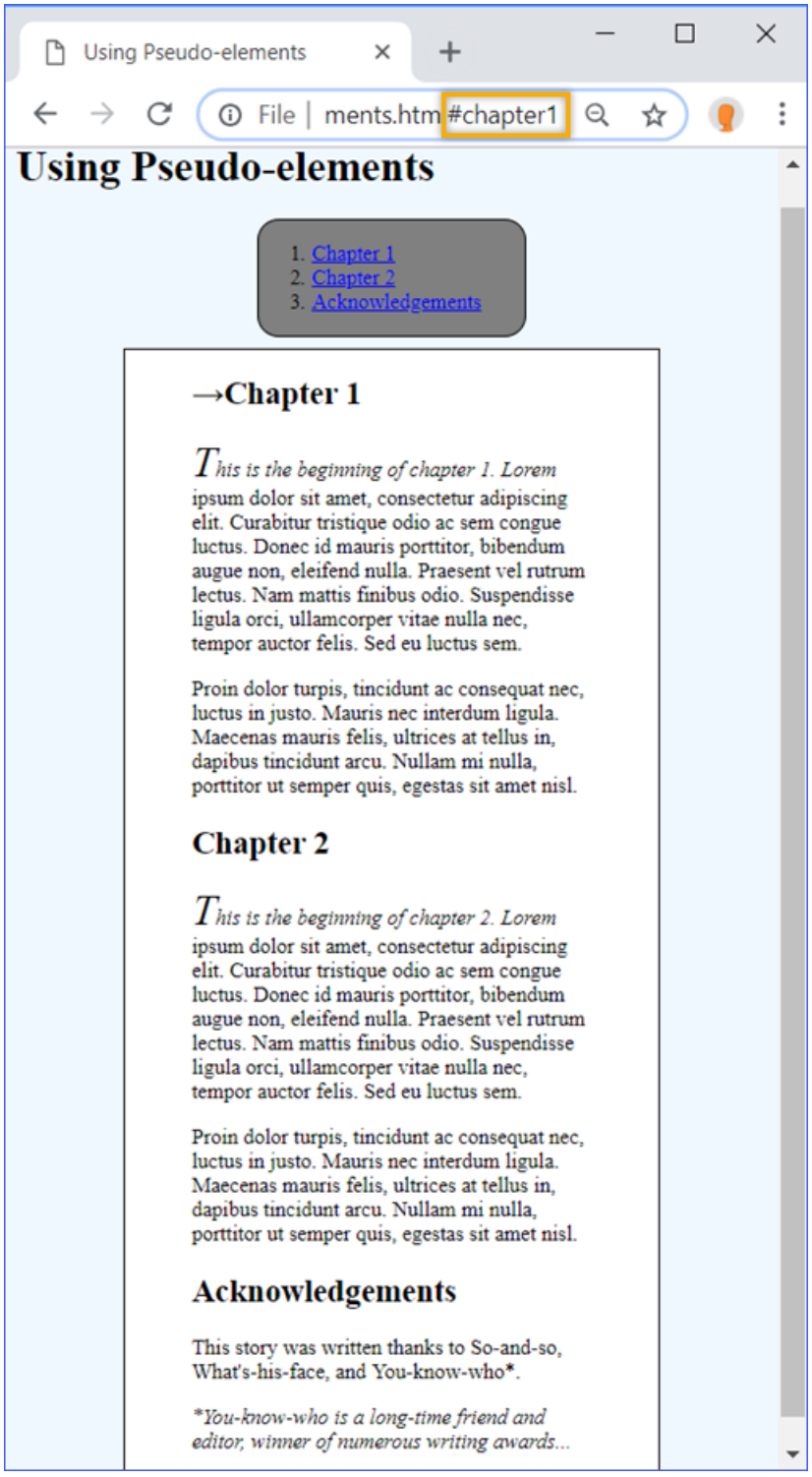
```
65.       </p>
66.     </section>
67.     <aside id="acknowledgements">
68.       <h2>Acknowledgements</h2>
69.       <p>This story was written thanks
70.         to So-and-so, What's-his-face, and
71.         <span class="note">You-know-who</span>.</p>
72.       <p class="bio">You-know-who is a long-time friend and editor,
73.         winner of numerous writing awards...</p>
74.     </aside>
75.   </main>
76.   </body>
77.   </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:

EVALUATION COPY: Not to be used in class.



Some things to note:

1. There are a lot of rules that don't use pseudo-elements. These rules are used to create the display of the demo.
2. `::before` is used to give the `h2` elements within the target an arrow before the targeted section. Note that `chapter1` is the target in the image.
3. `::after` and `::before` are also used to give an asterisk (`*`) to the `note` and `bio` classes.
4. `::first-letter` is used to make the first letter in each chapter bigger.
5. `::first-line` is used to make the first line in each chapter italic.

Evaluation Copy

EVALUATION COPY: Not to be used in class.

Exercise 10: Pseudo-classes and Pseudo-elements

25 to 40 minutes

In this exercise, you will continue to work on the Runners Home page.

1. Open `PseudoClassesAndElements/Exercises/index.html` in your editor. You will see the home page for a website called Runners Home.
2. Open `PseudoClassesAndElements/Exercises/styles.css`. This is the stylesheet from the solution to the Backgrounds exercise. Feel free to continue working from your own stylesheet.
3. Using your new knowledge of pseudo-classes and pseudo-elements add some styles to this page. The object of this exercise is to practice using the properties covered in this lesson. Also, feel free to add or change any other properties that you know.
4. When you are done, open `index.html` in your browser to see the results. You are welcome to go back to the code and continue to work.

Evaluation Copy

You can design it however you like, or you can try to:

- Make all links blue when they are normal or have been visited.
- Make the `nav` links purple when they are hovered over and red when they are clicked.
- Give `article`s a blur `box-shadow` effect when they are hovered over.
- Make the first letter in each `article` larger and bold, and make the first line italic.
- Make the `.read-more` links underlined when they are hovered over and purple when they are clicked.

Solution: PseudoClassesAndElements/Solutions/styles.css

```
1.  html {
2.    font-size: 16px;
3.  }
4.
5.  body {
6.    background-image: url(images/runners.png);
7.    background-size: cover;
8.    font: 1rem/1.3 Verdana, Geneva, Tahoma, sans-serif;
9.  }
10.
11. nav {
12.   background: url(images/runner.png) no-repeat bottom right;
13.   background-color: white;
14.   border: solid rgb(110, 78, 226) 0.1rem;
15.   border-radius: 1rem;
16.   box-shadow: 0.5rem 0.5rem 1rem rgb(110, 78, 226, .5);
17.   margin: 1rem;
18.   width: 40%;
19. }
20.
21. nav a {
22.   text-decoration: none;
23. }
24.
25. nav a:hover {
26.   color: purple;
27. }
28.
29. nav a:active {
30.   color: red;
31. }
32.
33. main {
34.   background: rgba(232, 232, 232, 0.8);
35.   padding: 2%;
36.   max-width: 1000px;
37.   min-width: 750px;
38. }
39.
40. #welcome {
41.   font-size: larger;
42. }
43.
44. #welcome h1 {
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
45.     color: rgb(8, 5, 211);
46.     text-shadow: 0.2rem 0.2rem 0.1rem rgb(211, 200, 238);
47.   }
48.
49.   article {
50.     background: white;
51.     border: 1px ridge rgb(226, 77, 40);
52.     box-sizing: border-box;
53.     margin: 1em auto;
54.     padding: 1rem;
55.     max-width: 75%;
56.     text-align: center;
57.   }
58.
59.   article h2 {
60.     font-variant: small-caps;
61.     letter-spacing: 0.1em;
62.     word-spacing: 0.2em;
63.   }
64.
65.   article p {
66.     line-height: 2;
67.     text-align: left;
68.     text-indent: 3rem;
69.   }
70.
71.   article:hover {
72.     box-shadow: 0 0 1rem 0.5rem rgba(189, 101, 189, 0.5);
73.   }
74.
75.   article p:first-of-type::first-line {
76.     font-style: italic;
77.   }
78.
79.   article p:first-of-type::first-letter {
80.     font-size: larger;
81.     font-weight: bold;
82.   }
83.
84.   a, a:visited {
85.     color: blue;
86.   }
87.
88.   .read-more {
89.     text-decoration: none;
```

Evaluation Copy

```
90.   }
91.
92.   a.read-more:hover {
93.     text-decoration: underline;
94.   }
95.
96.   a.read-more:active {
97.     color: purple;
98.   }
99.
100.  aside label {
101.    font-weight: bold;
102.  }
103.
104.  #newsletter {
105.    background: white;
106.    border: 1px dashed rgb(110, 78, 226);
107.    border-radius: 1em;
108.    margin: 0 1rem;
109.    padding: .5rem 1rem;
110.    text-align: right;
111.  }
112.
113.  footer {
114.    font-size: smaller;
115.    font-style: italic;
116.    margin: 1rem;
117.  }
```

Evaluation Copy

Code Explanation

You will need to open `PseudoClassesAndElements/Solutions/index.html` in the browser and interact with the page to see the `:hover` and `:active` effects we added.

Conclusion

In this lesson, you have learned to use CSS pseudo-classes and pseudo-elements.

EVALUATION COPY: Not to be used in class.

LESSON 9
Styling Tables with CSS

Topics Covered

☑ Formatting tables.

Introduction

In this lesson, you will learn to format tables with CSS.

9.1. A Review of HTML table Elements and Attributes

Below is a table of the `table` elements that you will commonly see in any HTML `table`:

Evaluation Copy

`table` Elements

Element	Definition
`table`	Contains the entire table.
`caption`	Specifies the caption (or title) of the table. The `caption` element is always the first child of the `table` element.
`thead`	Contains the header rows of the table. Usually contains the headers for each of the table columns.
`tbody`	Contains the main content of the table, meaning all the rows and cells that are not part of the table head (`thead`) or foot (`tfoot`).
`tfoot`	Contains the footer rows of the table. Usually contains a summary of the `tbody` content (e.g., a sum) or it may contain side notes if the data is more qualitative than quantitative.
`tr`	A table row.
`th`	A table header. `th` elements are contained within `tr` elements where they define the headers of the table columns and table rows.
`td`	A table data cell. `td` elements are contained within `tr` elements within the `tbody` and `tfoot` elements.

❖ 9.1.1. Spanning Columns and Rows

The `colspan` and `rowspan` attributes are used to make a table header cell or a table data cell span across multiple cells.

`colspan` and `rowspan`

Attribute	Tags	Use
`colspan`	`th`, `td`	Takes a positive integer greater than `0` that indicates how many columns the `th` or `td` element extends (e.g., `<td colspan="3">`). The default is `1`.
`rowspan`	`th`, `td`	Takes a positive integer that indicates how many rows the `th` or `td` element extends (e.g., `<td rowspan="3">`). The default is `1`. If its value is set to `0`, the `th` or `td` element will extend until the end of the table.

Evaluation Copy

Below you can see a basic unstyled table.

EVALUATION COPY: Not to be used in class.

Demo 9.1: StylingTables/Demos/unstyled-table.html

```
     -------Lines 1 through 7 Omitted-------
8.   <body>
9.   <h1>Unstyled Table</h1>
10.  <table>
11.    <caption>Great Books You Should Read</caption>
12.    <thead>
13.      <tr>
14.        <th id="title">Title</th>
15.        <th id="author">Author</th>
16.        <th id="genre">Genre</th>
17.      </tr>
18.    </thead>
19.    <tbody>
20.      <tr>
21.        <td>Harry Potter and the Sorcerer's Stone*</td>
22.        <td>J. K. Rowling</td>
23.        <td>Fantasy</td>
24.      </tr>
25.      <tr>
26.        <td>To Kill a Mockingbird</td>
27.        <td>Harper Lee</td>
28.        <td>Novel</td>
29.      </tr>
30.      <tr>
31.        <td>Pride and Prejudice</td>
32.        <td>Jane Austen</td>
33.        <td>Romance</td>
34.      </tr>
35.      <tr>
36.        <td>The Diary of Anne Frank</td>
37.        <td>Anne Frank</td>
38.        <td>Non-fiction</td>
39.      </tr>
40.      <tr>
41.        <td>1984</td>
42.        <td>George Orwell</td>
43.        <td>Dystopia</td>
44.      </tr>
45.      <tr>
46.        <td>The Hunger Games</td>
47.        <td>Suzanne Collins</td>
48.        <td>Dystopia</td>
49.      </tr>
50.      <tr>
```

Evaluation Copy

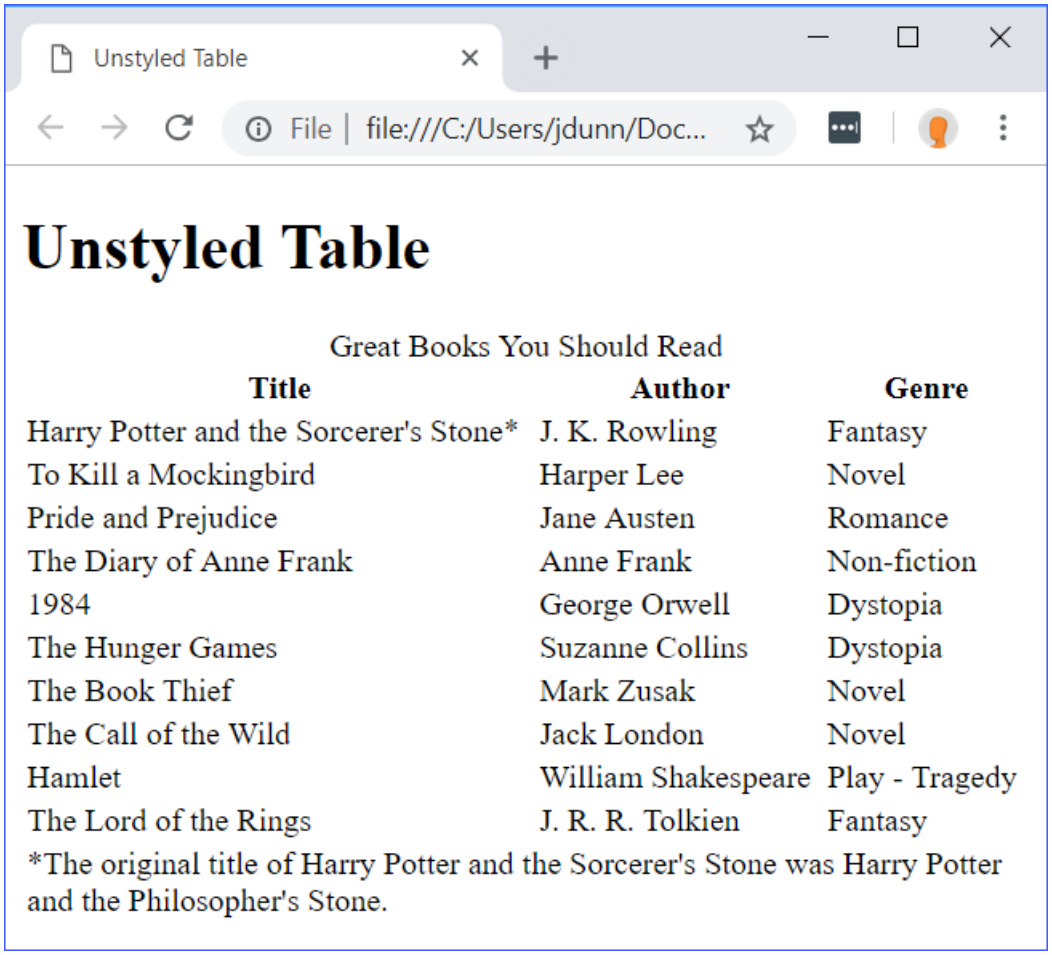
```
51.         <td>The Book Thief</td>
52.         <td>Mark Zusak</td>
53.         <td>Novel</td>
54.       </tr>
55.       <tr>
56.         <td>The Call of the Wild</td>
57.         <td>Jack London</td>
58.         <td>Novel</td>
59.       </tr>
60.       <tr>
61.         <td>Hamlet</td>
62.         <td>William Shakespeare</td>
63.         <td>Play - Tragedy</td>
64.       </tr>
65.       <tr>
66.         <td>The Lord of the Rings</td>
67.         <td>J. R. R. Tolkien</td>
68.         <td>Fantasy</td>
69.       </tr>
70.     </tbody>
71.     <tfoot>
72.       <tr>
73.         <td colspan="3">*The original title of
74.           <span>Harry Potter and
75.             the Sorcerer's Stone</span> was
76.           <span>Harry Potter and
77.             the Philosopher's Stone</span>.
78.         </td>
79.       </tr>
80.     </tfoot>
81.   </table>
82.   </body>
83.   </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:

EVALUATION COPY: Not to be used in class.



9.2. CSS Properties and Styling Tables

You already know a lot of the properties used to style tables as they are the same properties that you would use to style other elements (e.g., `background`, `padding`, `width`, `border`, etc.). We are going to look at some properties that are only used for tables:

1. `table-layout`
2. `border-collapse`

❖ 9.2.1. table-layout

The `table-layout` property is applied to the `table` element. It accepts one of two values:

1. `auto` - The default value. `auto` makes the widths of the `table`, `th`, and `td` elements adjust automatically to fit the content. `width` can still be used to make the `table` wider, but the `table` will never shrink so far that the content doesn't fit.
2. `fixed` - If a `table` is `fixed`, then the first table row determines the size of all the following rows. If they have content that goes wider than the first table row, then the content may wrap, get cut off, or extend outside of the cells depending on how the `overflow` and `whitespace` properties are set. `width` will override this tendency, and the `table` will be whatever `width` is set regardless of the content.

❖ 9.2.2. border-collapse

Evaluation Copy

The `border-collapse` property is also applied to the `table` element. It determines whether or not table cells have separate or shared borders. It takes one of two values:

1. `separate` - The default value. When cells have separate borders, the `table`, `th`, and `tr` elements all have their own four borders. The spacing between these borders is determined by the `border-spacing` property, which only works when `border-collapse` is set to `separate`.
    A. `border-spacing` takes one or two length values. If one value is specified, it is for both the vertical and horizontal spacing (e.g., `border-spacing: 0.2rem`). If two values are specified, the first one is for the vertical spacing and the second is for the horizontal spacing (e.g., `border-spacing: 0.2rem 0.1rem`).
2. `collapse` - If a `table` element has its borders collapsed, then the `table`, `th`, and `td` elements share borders with each other.

Below you can see a `table` with separate borders and a `table` with collapsed borders.

EVALUATION COPY: Not to be used in class.

Demo 9.2: StylingTables/Demos/separate-vs-collapse-styles.css

```
1.   table,
2.   th,
3.   td {
4.     border: solid black;
5.   }
6.
7.   table.separate {
8.     border-collapse: separate;
9.     border-spacing: 0.5rem;
10.  }
11.
12.  table.collapse {
13.    border-collapse: collapse;
14.  }
15.
16.  caption {
17.    font-style: italic;
18.  }
```

Evaluation Copy

Demo 9.3: StylingTables/Demos/separate-vs-collapse.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.  <h1>separate vs. collapse</h1>
12.  <table class="separate">
13.    <caption>separate</caption>
14.    <thead>
15.      <tr>
16.        <th>Column 1</th>
17.        <th>Column 2</th>
18.        <th>Column 3</th>
19.      </tr>
20.    </thead>
21.    <tbody>
22.      <tr>
23.        <td>Data 1</td>
24.        <td>Data 2</td>
25.        <td>Data 3</td>
26.      </tr>
27.      <tr>
28.        <td>Data 1</td>
29.        <td>Data 2</td>
30.        <td>Data 3</td>
31.      </tr>
32.      <tr>
33.        <td>Data 1</td>
34.        <td>Data 2</td>
35.        <td>Data 3</td>
36.      </tr>
37.    </tbody>
38.    <tfoot>
39.      <tr>
40.        <td colspan="3">Footer</td>
41.      </tr>
42.    </tfoot>
43.  </table>
44.  <table class="collapse">
45.    <caption>collapse</caption>
46.    <thead>
47.      <tr>
48.        <th>Column 1</th>
49.        <th>Column 2</th>
50.        <th>Column 3</th>
51.      </tr>
52.    </thead>
```

Evaluation Copy

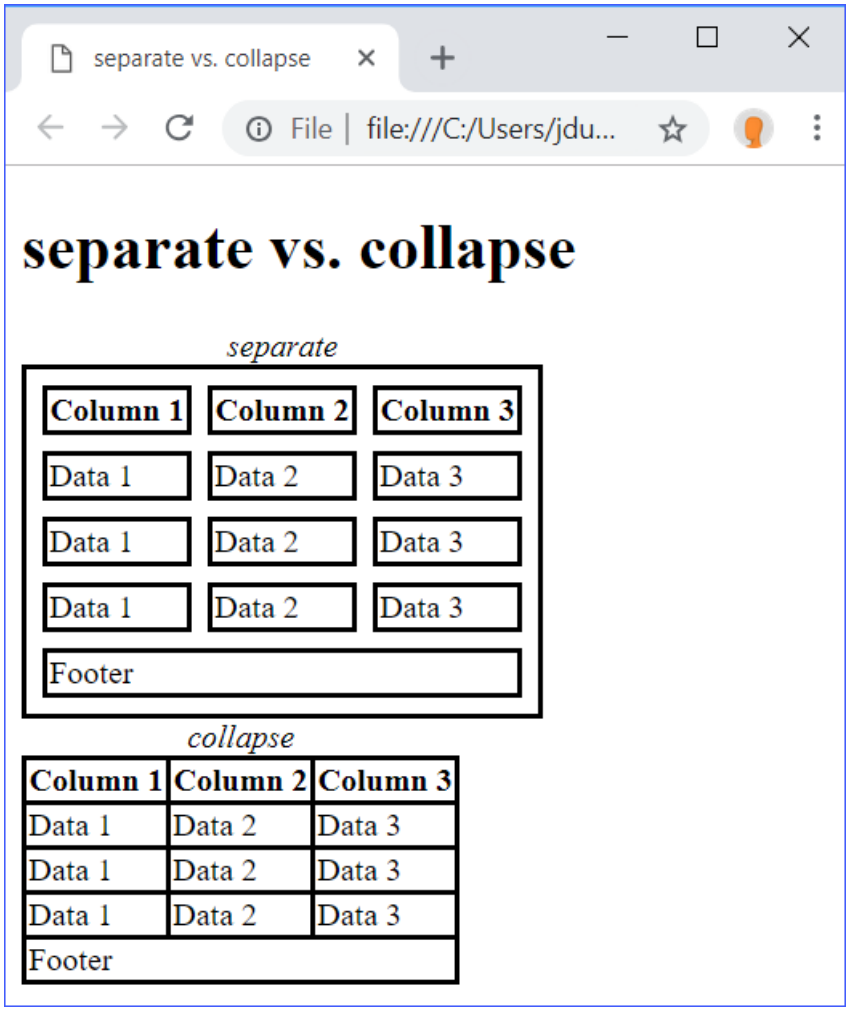
EVALUATION COPY: Not to be used in class.

```
53.     <tbody>
54.       <tr>
55.         <td>Data 1</td>
56.         <td>Data 2</td>
57.         <td>Data 3</td>
58.       </tr>
59.       <tr>
60.         <td>Data 1</td>
61.         <td>Data 2</td>
62.         <td>Data 3</td>
63.       </tr>
64.       <tr>
65.         <td>Data 1</td>
66.         <td>Data 2</td>
67.         <td>Data 3</td>
68.       </tr>
69.     </tbody>
70.     <tfoot>
71.       <tr>
72.         <td colspan="3">Footer</td>
73.       </tr>
74.     </tfoot>
75.   </table>
76.   </body>
77.   </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:



Note that the `separate table` also has `0.5rem` of `border-spacing`.

Below, you can see our fully styled Great Books table:

EVALUATION COPY: Not to be used in class.

Demo 9.4: StylingTables/Demos/great-books-table-styles.css

```
1.  body {
2.    background: aliceblue;
3.  }
4.
5.  table,
6.  th,
7.  td {
8.    border: solid black 1px;
9.    padding: 0.2rem;
10. }
11.
12. table {
13.   background: white;
14.   border-collapse: collapse;
15.   table-layout: fixed;
16.   width: 600px;
17.   margin: auto;
18. }
19.
20. #title {
21.   width: 50%;
22. }
23.
24. caption {
25.   font-size: 1.5rem;
26.   font-weight: bold;
27.   text-decoration: underline;
28. }
29.
30. thead {
31.   background: rgb(75, 75, 75);
32.   color: white;
33.   text-decoration: underline;
34. }
35.
36. span.title {
37.   font-style: italic;
38. }
39.
40. tfoot {
41.   background: rgb(75, 75, 75);
42.   color: white;
43. }
44.
```

Evaluation Copy

```
45.   tbody td:first-child {
46.     font-style: italic;
47.   }
48.
49.   tbody tr:nth-child(odd) {
50.     background-color: #ccc;
51.   }
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

Demo 9.5: StylingTables/Demos/great-books-table.html

```
    -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.  <h1>Styling a Table</h1>
12.  <table>
13.    <caption>Great Books You Should Read</caption>
14.    <thead>
15.      <tr>
16.        <th id="title">Title</th>
17.        <th id="author">Author</th>
18.        <th id="genre">Genre</th>
19.      </tr>
20.    </thead>
21.    <tbody>
22.      <tr>
23.        <td>Harry Potter and the Sorcerer's Stone*</td>
24.        <td>J. K. Rowling</td>
25.        <td>Fantasy</td>
26.      </tr>
27.      <tr>
28.        <td>To Kill a Mockingbird</td>
29.        <td>Harper Lee</td>
30.        <td>Novel</td>
31.      </tr>
32.      <tr>
33.        <td>Pride and Prejudice</td>
34.        <td>Jane Austen</td>
35.        <td>Romance</td>
36.      </tr>
37.      <tr>
38.        <td>The Diary of Anne Frank</td>
39.        <td>Anne Frank</td>
40.        <td>Non-fiction</td>
41.      </tr>
42.      <tr>
43.        <td>1984</td>
44.        <td>George Orwell</td>
45.        <td>Dystopia</td>
46.      </tr>
47.      <tr>
48.        <td>The Hunger Games</td>
49.        <td>Suzanne Collins</td>
50.        <td>Dystopia</td>
51.      </tr>
52.      <tr>
```

Evaluation Copy

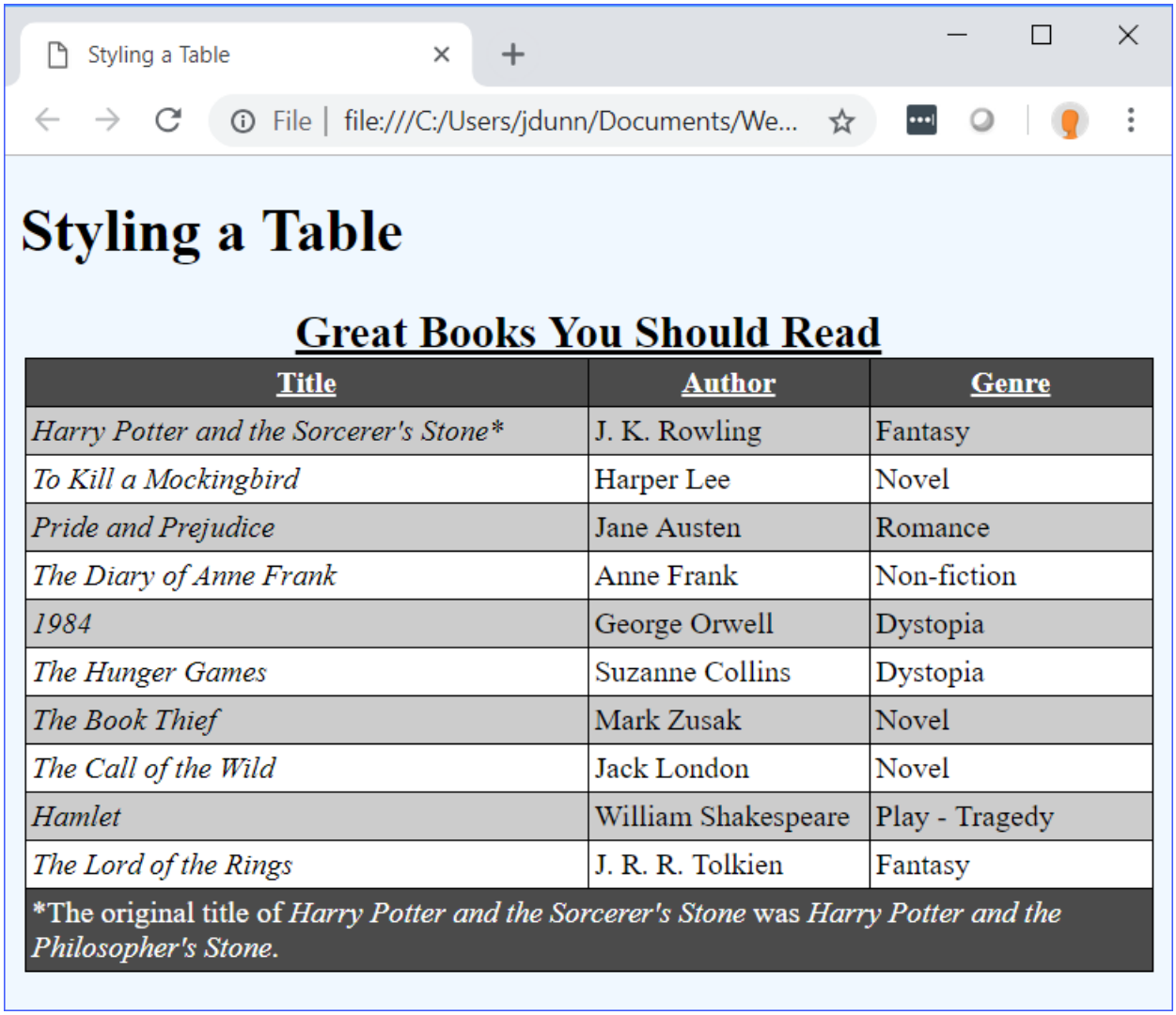
```
53.         <td>The Book Thief</td>
54.         <td>Mark Zusak</td>
55.         <td>Novel</td>
56.       </tr>
57.       <tr>
58.         <td>The Call of the Wild</td>
59.         <td>Jack London</td>
60.         <td>Novel</td>
61.       </tr>
62.       <tr>
63.         <td>Hamlet</td>
64.         <td>William Shakespeare</td>
65.         <td>Play - Tragedy</td>
66.       </tr>
67.       <tr>
68.         <td>The Lord of the Rings</td>
69.         <td>J. R. R. Tolkien</td>
70.         <td>Fantasy</td>
71.       </tr>
72.     </tbody>
73.     <tfoot>
74.       <tr>
75.         <td colspan="3">*The original title of
76.           <span class="title">Harry Potter and
77.             the Sorcerer's Stone</span> was
78.           <span class="title">Harry Potter and
79.             the Philosopher's Stone</span>.
80.         </td>
81.       </tr>
82.     </tfoot>
83.   </table>
84.   </body>
85.   </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:

EVALUATION COPY: Not to be used in class.



Some things to note:

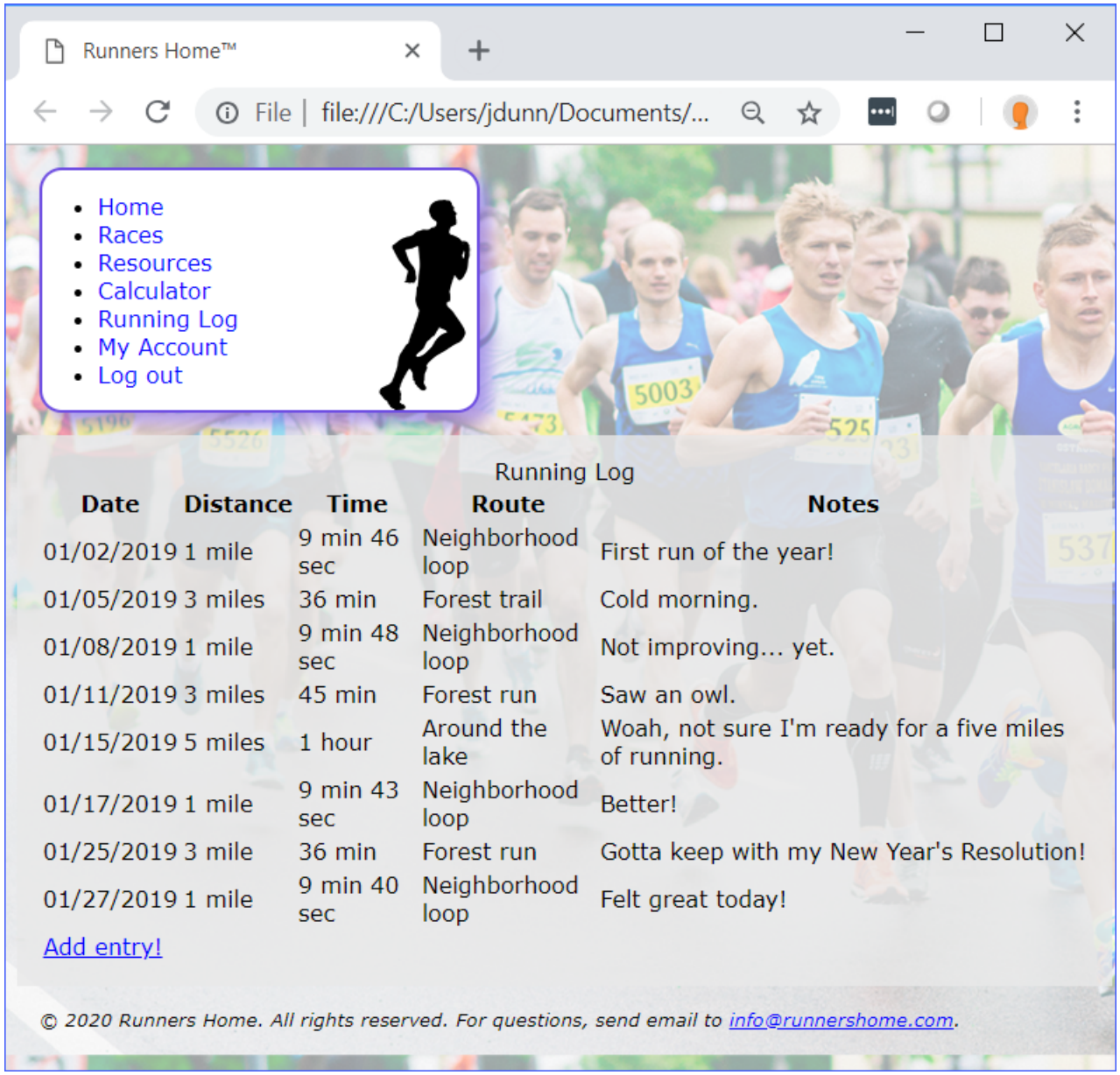
1. We collapsed the borders.
2. We made the table a fixed table.

Exercise 11: Styling a Table Exercise

25 to 40 minutes

In this exercise, you will continue to work on the Runners Home website, but you are going to edit a new page.

1. Open `StylingTables/Exercises/running-log.html` in your editor. You will see an unstyled table that looks like this:

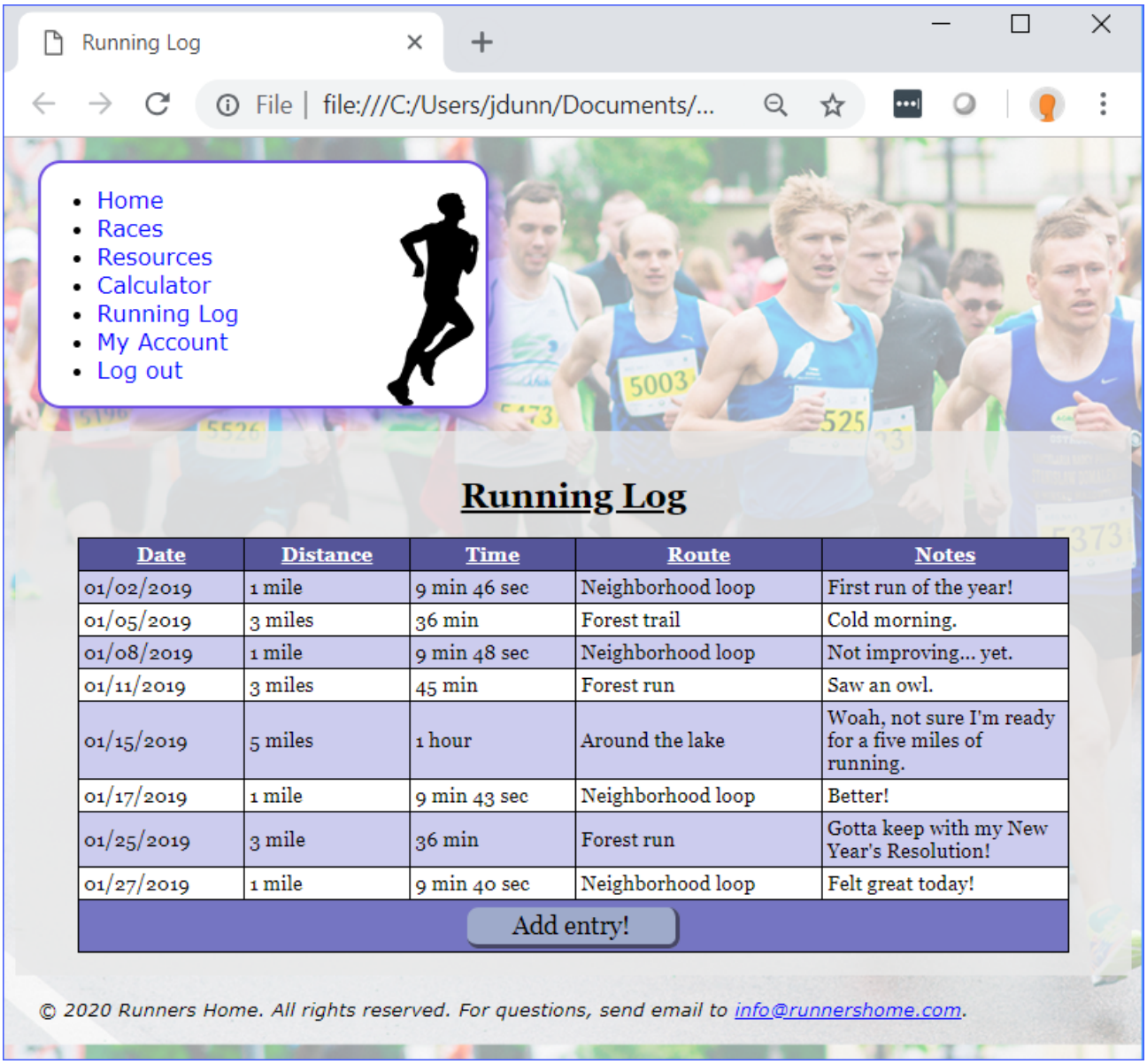


EVALUATION COPY: Not to be used in class.

2. Open `StylingTables/Exercises/styles.css` in your editor. This is the stylesheet from the solution to the Pseudo-classes and Pseudo-elements exercise. Feel free to continue working from your own stylesheet.
3. Using your new knowledge of table styles, style the unstyled Running Log table. The object of this exercise is to practice using the properties you just learned about. Also, feel free to add or change any other properties that you know.
4. When you are done, open `running-log.html` in your browser to see the results. You are welcome to go back to the code and continue to work.

Evaluation Copy

You can design it however you like, or you can try to make it look something like:



EVALUATION COPY: Not to be used in class.

Solution: StylingTables/Solutions/styles.css

```
     -------Lines 1 through 117 Omitted-------
118.
119. #running-log {
120.   background: white;
121.   border-collapse: collapse;
122.   font: 0.9rem Georgia, Times, serif;
123.   margin: auto;
124.   table-layout: fixed;
125.   width: 700px;
126. }
127.
128. table,
129. th,
130. td {
131.   border: solid black 1px;
132.   padding: 0.2rem;
133. }
134.
135. caption {
136.   font-size: 1.5rem;
137.   font-weight: bold;
138.   padding: 1rem;
139.   text-decoration: underline;
140. }
141.
142. #route,
143. #notes {
144.   width: 25%;
145. }
146.
147. #running-log thead {
148.   background: #559;
149.   color: white;
150.   text-decoration: underline;
151. }
152.
153. tbody tr:nth-child(odd) {
154.   background: #cce;
155. }
156.
157. #running-log tfoot td {
158.   background-color: rgb(117, 117, 194);
159.   padding: 0.5rem;
160.   text-align: center;
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
161. }
162.
163. #running-log tfoot a {
164.   background: rgb(158, 170, 202);
165.   border: rgb(70, 70, 100);
166.   border-radius: 0.5rem;
167.   border-style: none solid solid none;
168.   border-width: 0 0.2rem 0.2rem 0;
169.   color: black;
170.   font-size: 1.1rem;
171.   padding: 0.2rem 2rem;
172.   text-decoration: none;
173. }
174.
175. #running-log tfoot a:hover {
176.   color: rgb(97, 8, 133);
177. }
178.
179. #running-log tfoot a:active {
180.   color:#cce;
181.   border-width: 0 0.1rem 0.1rem 0;
182. }
```

Evaluation Copy

Code Explanation

Some things to note:

1. We omitted previous styles for brevity.
2. We added link pseudo-classes to the link in the `tfoot`. You will need to open `StylingTables/Solutions/running-log.html` in the browser and interact with the link to see the effect of these styles.

Conclusion

In this lesson, you have learned to style HTML tables with CSS.

EVALUATION COPY: Not to be used in class.

LESSON 10
Positioning

Topics Covered

☑ Normal Flow.

☑ The `position`, `top`, `bottom`, `left`, and `right` properties.

☑ The `z-index` property.

☑ The `float` and `clear` properties.

Introduction

In this lesson, you will learn to work with CSS properties that are used to position elements on the page.

Evaluation Copy

10.1. Normal Flow

The *flow* is the way in which elements are laid out and the *normal flow* is the way in which elements are laid out by default:

1. Elements appear on the page in the same order they appear in the code.
2. Block elements appear on separate lines in single columns.
3. Top and bottom margins collapse between block elements, so two block elements on top of each other will be separated by the `margin` of the element with the larger `margin` and not the combined `margin` of both elements.
4. Inline elements appear on the same line adjacent to each other and any text as long as there is space. If there isn't space, they will wrap down to the next line.

A demo illustrating normal flow is shown below:

Demo 10.1: Positioning/Demos/normal-flow.html

```
      -------Lines 1 through 6 Omitted-------
7.    <style>
8.      .block {
9.        background: lightblue;
10.       border: solid black;
11.       margin: 1rem;
12.       padding: 0.5rem;
13.     }
14.
15.     .inline {
16.       background: pink;
17.       border: solid black;
18.       padding: 0.1rem;
19.     }
20.   </style>
21.   <title>Normal Flow</title>
22.   </head>
23.   <body class="webucator">
24.   <h1>Normal Flow</h1>
25.   <main>
26.     <div class="block">
27.       This is a div element, a standard block element.
28.     </div>
29.     <p class="block">
30.       This is a paragraph element,
31.       another block element, that contains a
32.       <span class="inline">
33.         span element, which is a standard inline element
34.       </span>.
35.     </p>
36.     <div class="block">
37.       This is a div element that contains a paragraph element.
38.         <p class="block">
39.           This is a paragraph element that contains two
40.           <span class="inline">span</span>
41.           <span class="inline">elements</span>.
42.           It also contains a lot more content
43.           like these fake sentences:
44.           Lorem ipsum dolor
45.           sit amet, consectetur adipiscing elit.
46.           Curabitur tristique odio ac sem congue luctus.
47.         </p>
48.     </div>
49.   </main>
```

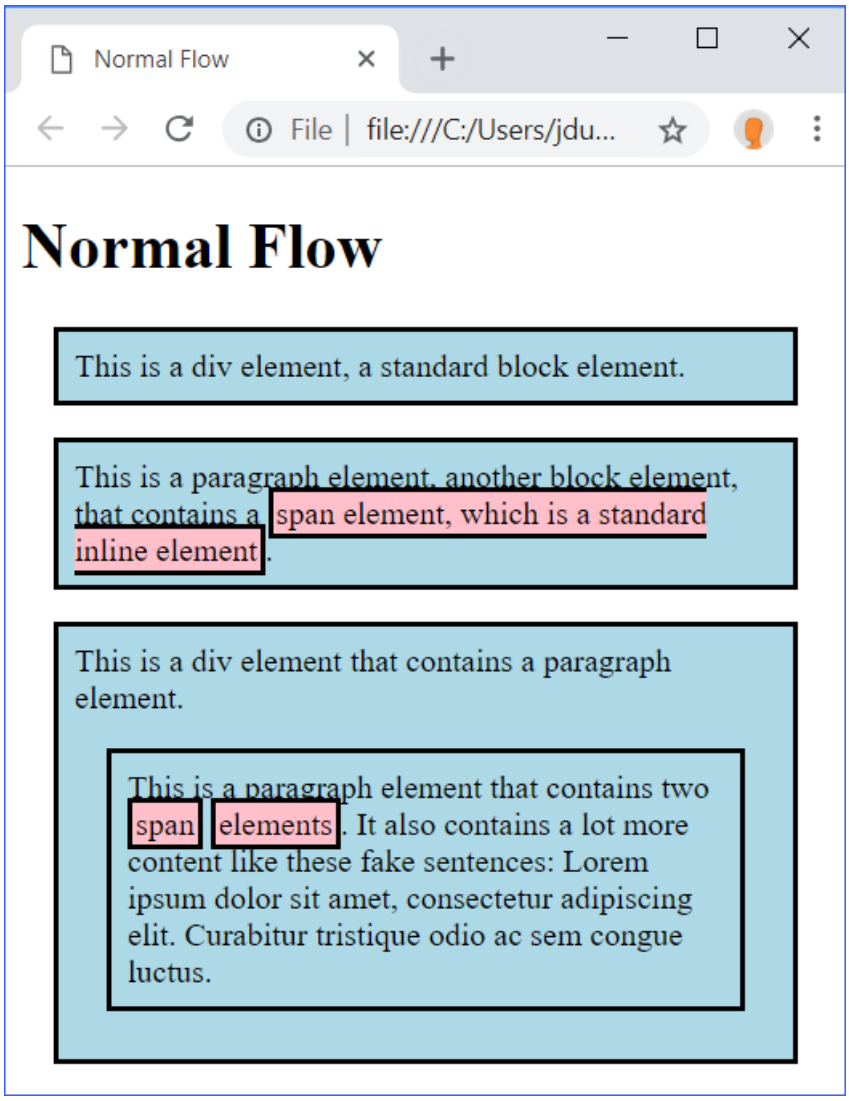
Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
50.   </body>
51.   </html>
```

Code Explanation

The code above will render the following:



With CSS, we can move elements into a separate flow. This allows us more control over the design and layout of our pages. Throughout this lesson, we will be looking at the different ways we can do this.

10.2. position

The `position` property determines how an element is positioned. It takes the following values:

1. `static` - The default value. The element is part of the normal flow.
2. `relative` - The element remains in the normal flow, but is positioned *relative* to its default position in the normal flow.
3. `absolute` - The element is removed from the normal flow and no space is left for it in the page layout, and as a result it does not affect the flow of the elements around it. It is positioned relative to its closest *positioned ancestor**. If it has no positioned ancestor, it is positioned relative to the document body.
4. `fixed` - The element is removed from the normal flow and no space is left for it in the page layout, and as a result it does not affect the flow of the elements around it. It is positioned relative to the viewport, meaning that even when the page scrolls, it will stay in the same place.

Evaluation Copy

*A *positioned element* is an element with any `position` value other than `static`.

❖ 10.2.1. Positioning with top, bottom, left, and right

`top`, `bottom`, `left`, and `right` are used to position non-`static` elements by setting offsets. They all take the following properties:

1. `auto` - the element is positioned where it would be if it were positioned statically.
2. A length value in units (e.g., `1em`) that specifies the offset distance for that respective edge.
3. A percentage value of the container element's width or height (depending on whether the property is a vertical or horizontal edge).

Some important distinctions in how these properties work with the different `position` values:

1. For absolutely positioned elements, the edges are defined by the edges of the padding of the closest positioned parent.

EVALUATION COPY: Not to be used in class.

2. For relatively positioned elements, the edges are defined by the edges of the content of the element itself.
3. For fixed positioned elements, the containing element is always the viewport and the edges are defined by the edges of the viewport.

Using top, bottom, left, and right

When using these properties to position elements, you can set either `top` or `bottom`, but not both, and you can set either `left` or `right`, but not both.

Evaluation Copy

Valid combinations are `top` and `left`, `top` and `right`, `bottom` and `left`, and `bottom` and `right`.

The following demo shows some examples of positioning:

Demo 10.2: Positioning/Demos/position-styles.css

```
1.  body {
2.    border: 1px dotted black;
3.  }
4.
5.  section {
6.    border: 1px solid black;
7.    margin: 2rem;
8.    position: relative;
9.  }
10.
11. div {
12.   background: rgba(0, 0, 255, 0.5);
13.   border: 1px dashed black;
14.   height: 25px;
15.   margin: auto;
16.   width: 75px;
17.   top: 75px;
18.   left: 75px;
19. }
20.
21. .relative {
22.   background: rgba(255, 0, 0, 0.5);
23.   position: relative;
24. }
25.
26. .absolute {
27.   background: rgba(0, 255, 0, 0.5);
28.   position: absolute;
29. }
30.
31. .fixed {
32.   background: rgba(255, 255, 0, 0.5);
33.   position: fixed;
34. }
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

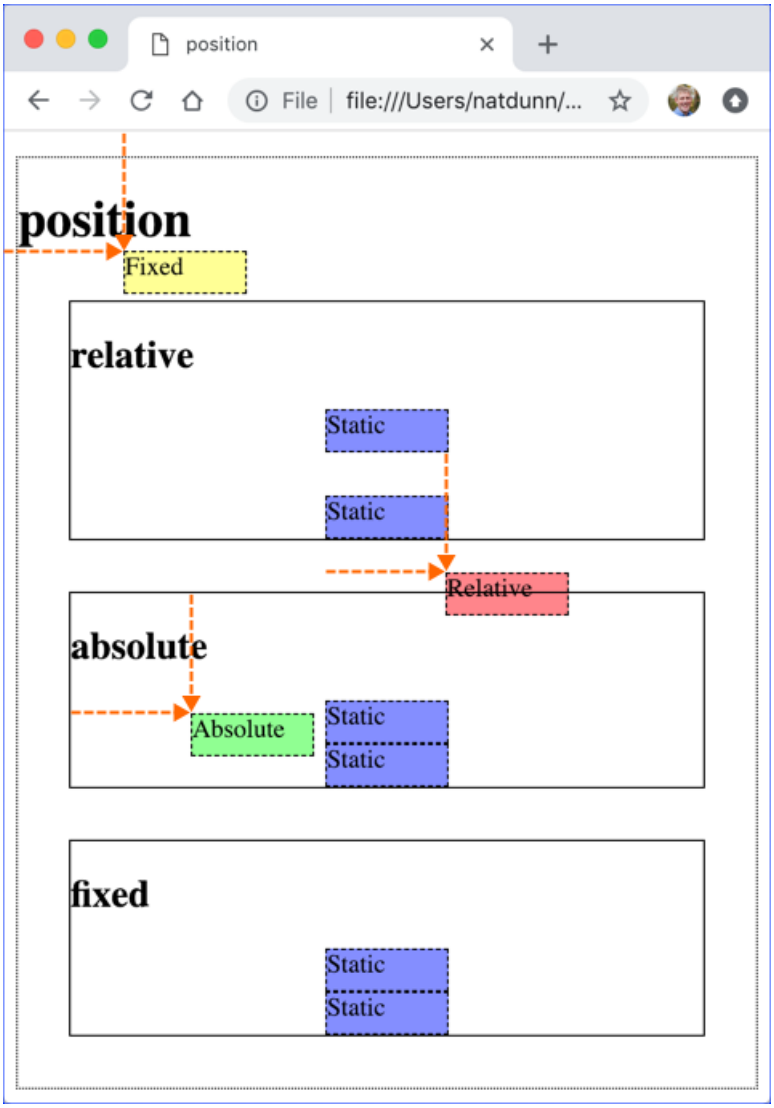
Demo 10.3: Positioning/Demos/position.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.  <h1>position</h1>
12.  <section>
13.    <h2>relative</h2>
14.    <div>Static</div>
15.    <div class="relative">Relative</div>
16.    <div>Static</div>
17.  </section>
18.  <section>
19.    <h2>absolute</h2>
20.    <div>Static</div>
21.    <div class="absolute">Absolute</div>
22.    <div>Static</div>
23.  </section>
24.  <section>
25.    <h2>fixed</h2>
26.    <div>Static</div>
27.    <div class="fixed">Fixed</div>
28.    <div>Static</div>
29.  </section>
30.  </body>
31.  </html>
```

Evaluation Copy

Code Explanation

The screenshot below shows how this code will render. The dashed arrows show the top and left offsets.



Note that the fixed `div` will always stay in the top-right corner even if the page scrolls. Also note that the absolute and fixed `div`s get removed from the normal flow, such that the subsequent `div`s don't make space for it. The relative `div` remains in the normal flow, so subsequent `div`s act as if it were where it would have been if it were static.

EVALUATION COPY: Not to be used in class.

10.3. z-index

The `z-index` property is used to specify the stacking order of positioned elements. It takes the following values:

1. `auto` - the default. Elements that are later siblings sit on top of their earlier siblings if they are positioned to overlap.
2. A positive or negative integer. The higher the integer, the higher an element is in the stacking order of its siblings.

Evaluation Copy

The following demo shows some examples of `z-index`:

Demo 10.4: Positioning/Demos/z-index-styles.css

```
1.  nav {
2.    background: lightgray;
3.    border: 1px solid black;
4.    border-radius: 0.5rem;
5.    width: 300px;
6.  }
7.
8.  section {
9.    background: rgba(211, 211, 211, 0.8);
10.   border: 1px solid black;
11.   position: relative;
12.   height: 200px;
13.   width: 300px;
14. }
15.
16. .top-section {
17.   z-index: 3;
18. }
19.
20. .middle-section {
21.   top: -3rem;
22.   z-index: 2;
23. }
24.
25. .bottom-section {
26.   top: -20rem;
27.   left: 10rem;
28.   z-index: 1;
29. }
30.
31. div {
32.   background: rgba(128, 128, 128, 0.8);
33.   border: 1px solid black;
34.   height: 100px;
35.   position: absolute;
36.   width: 100px;
37. }
38.
39. .top-div {
40.   left: 1rem;
41.   top: 2rem;
42.   z-index: 3;
43. }
44.
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
45.  .middle-div {
46.    left: 4rem;
47.    top: 3.5rem;
48.    z-index: 2;
49.  }
50.
51.  .bottom-div {
52.    left: 8rem;
53.    top: 5rem;
54.    z-index: 1;
55.  }
56.
57.  :target {
58.    /* Set z-index of targeted element to 100 */
59.    z-index: 100;
60.  }
61.
62.  /* Add text to targeted element */
63.  :target::after {
64.    content: "(z-index: 100)"
65.  }
```

Evaluation Copy

Demo 10.5: Positioning/Demos/z-index.html

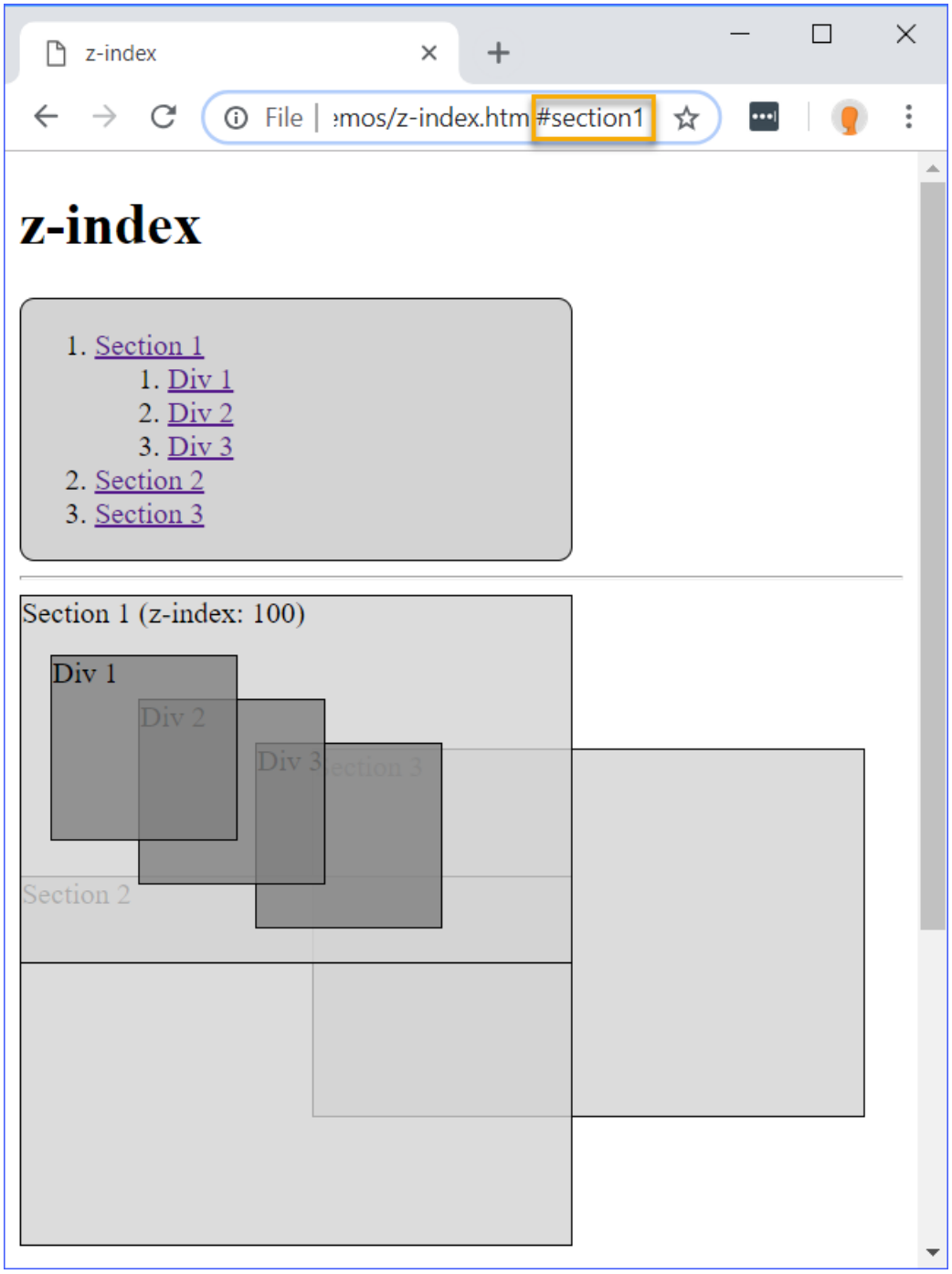
```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.  <h1>z-index</h1>
12.  <nav>
13.    <ol>
14.      <li><a href="#section1">Section 1</a>
15.        <ol>
16.          <li><a href="#div1">Div 1</a></li>
17.          <li><a href="#div2">Div 2</a></li>
18.          <li><a href="#div3">Div 3</a></li>
19.        </ol>
20.      </li>
21.      <li><a href="#section2">Section 2</a>
22.      </li>
23.      <li><a href="#section3">Section 3</a>
24.      </li>
25.    </ol>
26.  </nav>
27.  <hr>
28.  <section id="section1" class="top-section">
29.    Section 1
30.    <div id="div1" class="top-div">
31.      Div 1
32.    </div>
33.    <div id="div2" class="middle-div">
34.      Div 2
35.    </div>
36.    <div id="div3" class="bottom-div">
37.      Div 3
38.    </div>
39.  </section>
40.  <section id="section2" class="middle-section">
41.    Section 2
42.  </section>
43.  <section id="section3" class="bottom-section">
44.    Section 3
45.  </section>
46.  </body>
47.  </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:

EVALUATION COPY: Not to be used in class.



Open `Positioning/Demos/z-index.html` in the browser and click some of the links at the top. You'll notice that the targeted element always rises to the top. That's because targeted elements get a z-index of 100.

Try setting the background color of the targeted element to red in the CSS to see this change even more pronounced:

```
:target {
  /* Set z-index of targeted element to 100 */
  z-index: 100;
  background-color: red;
}
```

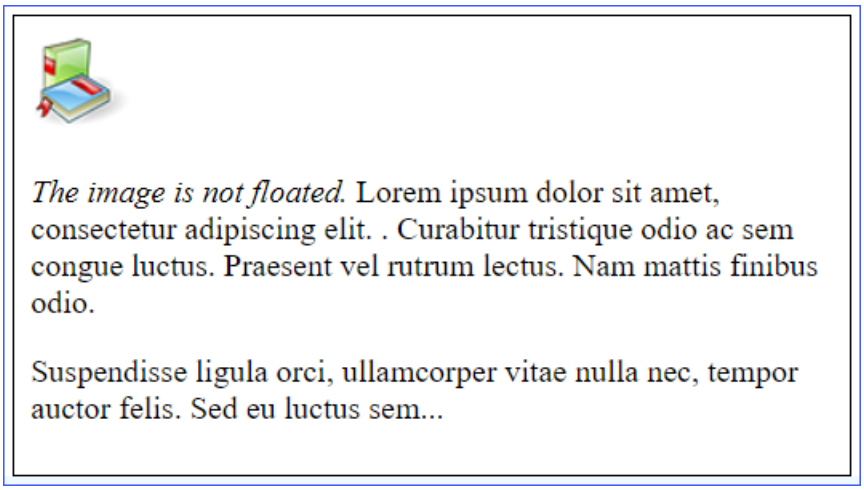
10.4. float and clear

❖ 10.4.1. float

Evaluation Copy

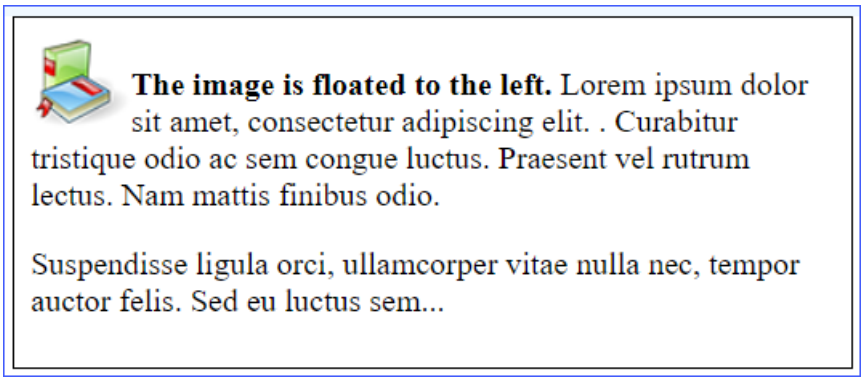
The `float` property places block elements to the left or right of their containing elements. It takes the following values:

1. `none` - The default value. The element is not floated:

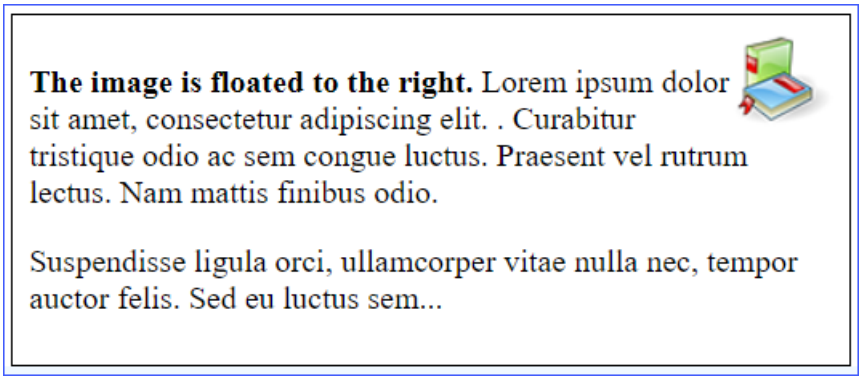


2. `left` - The element is floated to the left in its containing element:

EVALUATION COPY: Not to be used in class.



3. `right` - The element is floated to the right in its containing element:



Floated elements are taken out of normal flow, but they continue to occupy space in the page layout, unlike absolutely positioned elements.

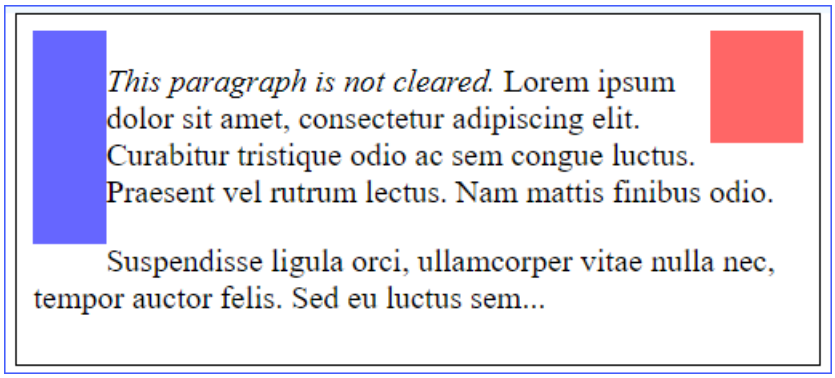
All these examples can be seen in `Positioning/Demos/float.html` and `Positioning/Demos/float-styles.css`.

Open `Positioning/Demos/float.html` in the browser and click some of the links at the top to change the `float` values.

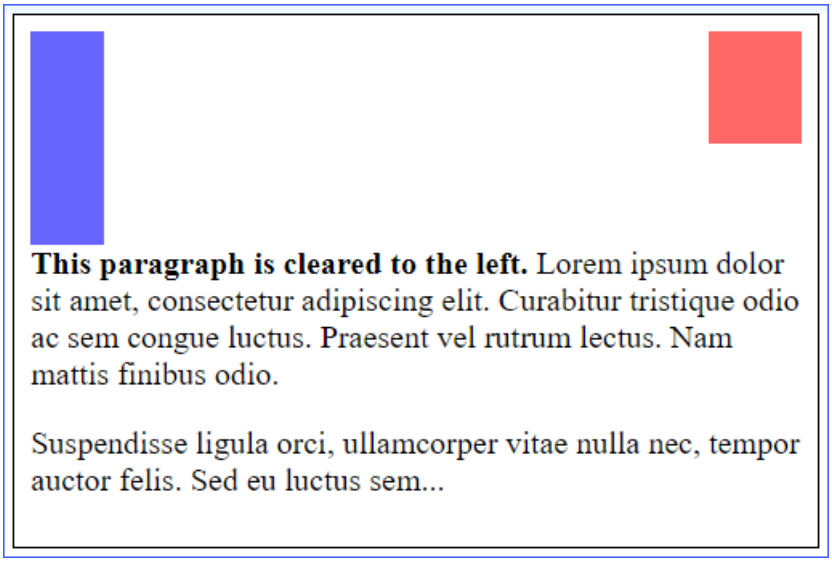
❖ 10.4.2. clear

The `clear` property is used to force elements to move below floated elements. It takes the following values:

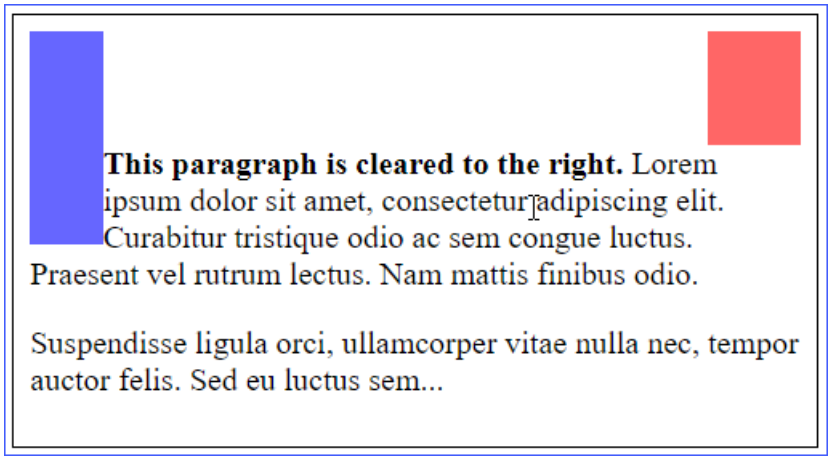
1. `none` - The default value. The element is not cleared:



2. `left` - The element is forced below all elements floated left in its containing element:

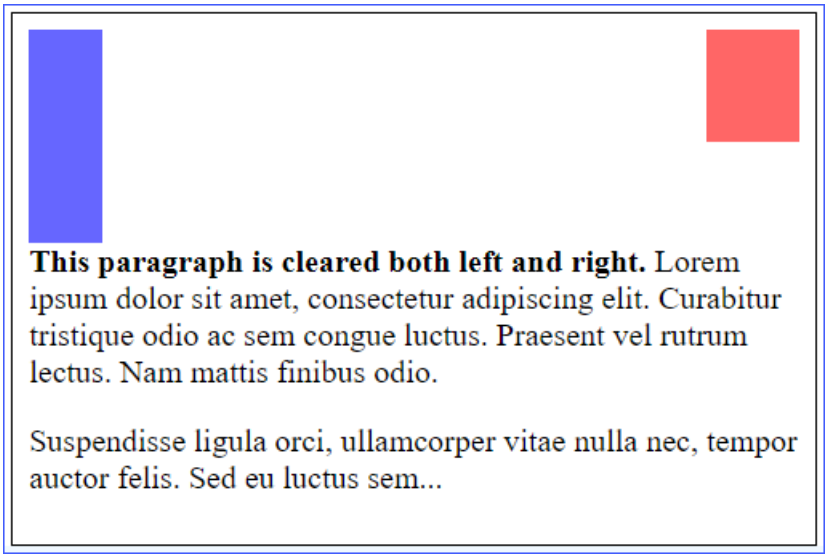


3. `right` - The element is forced below all elements floated right in its containing element:



EVALUATION COPY: Not to be used in class.

4. `both` - The element is forced below all floated elements in its containing element:



These examples can be seen in `Positioning/Demos/clear.html` and `Positioning/Demos/clear-styles.css`.

Evaluation Copy

Open `Positioning/Demos/clear.html` in the browser and click the links at the top to change the `clear` values.

Layouts with float and clear

Before the development of more sophisticated CSS layout systems, `float` and `clear` were used to create entire website layouts, but this is no longer recommended.

❖ 10.4.3. float, clear, and ::after

Sometimes the floated element is so big that when it is taken out of the flow, it extends beyond the height (which adjusts automatically for its content in normal flow) of its containing element. To fix this, it is possible to give the height a fixed value (e.g., `400px`), but this is not great for fluid pages with containers that may shrink and grow depending on the width. The solution is to use the `::after` pseudo-element in combination with the `clear` property.

The syntax is:

```
container::after {
  content: "";
  clear: both;
  display: block;
}
```

Evaluation Copy

This adds a cleared block pseudo-element at the end of the container element. We use an empty string ("") because we don't want the pseudo-element to be visible to the user. The cleared pseudo-element must stay below any floated elements, so it keeps the height of container from becoming smaller than the floated element. See the demo below for an example:

EVALUATION COPY: Not to be used in class.

Demo 10.6: Positioning/Demos/float-clear-after.html

```
      -------Lines 1 through 6 Omitted-------
7.    <style>
8.      article {
9.        border: 1px solid black;
10.       margin: 0.5rem;
11.       padding: 0.5rem;
12.     }
13.
14.     #article1 img {
15.       float: right;
16.     }
17.
18.     #article2 img {
19.       float: left;
20.     }
21.
22.     #article2::after {
23.       content: "";
24.       clear: both;
25.       display: block;
26.     }
27.   </style>
28.   <title>float, clear, and ::after</title>
29.   </head>
30.   <body class="webucator">
31.   <h1>float, clear, and ::after</h1>
32.   <main>
33.     <article id="article1" class="top-article">
34.       <img alt="Marathon" src="images/marathon.jpg">
35.       <p>Lorem ipsum dolor
36.         sit amet, consectetur adipiscing elit. .
37.         Curabitur tristique odio ac sem congue luctus.
38.         Praesent vel rutrum lectus.
39.         Nam mattis finibus odio.
40.         <p>Suspendisse ligula orci,
41.         ullamcorper vitae nulla nec, tempor auctor felis.
42.         Sed eu luctus sem...</p>
43.     </article>
44.     <article id="article2" class="middle-article">
45.       <img alt="Marathon" src="images/marathon.jpg">
46.       <p>Lorem ipsum dolor
47.         sit amet, consectetur adipiscing elit. .
48.         Curabitur tristique odio ac sem congue luctus.
49.         Praesent vel rutrum lectus.
```

Evaluation Copy

```
50.         Nam mattis finibus odio.
51.         <p>Suspendisse ligula orci,
52.         ullamcorper vitae nulla nec, tempor auctor felis.
53.         Sed eu luctus sem...</p>
54.     </article>
55.   </main>
56.   </body>
57.   </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:

EVALUATION COPY: Not to be used in class.

Some things to note:

1. The first `article` element adjusts only for the text and not for the floated image. This allows the image to extend past its container and interfere with the rest of the layout.

2. The second `article` element has a cleared pseudo-element added at the end of it, so the second `article` extends all the way to the bottom of the image.

In the following image, we added text and a border to the pseudo-element, so that you can see that it is really there:

EVALUATION COPY: Not to be used in class.

Exercise 12: Positioning

25 to 40 minutes

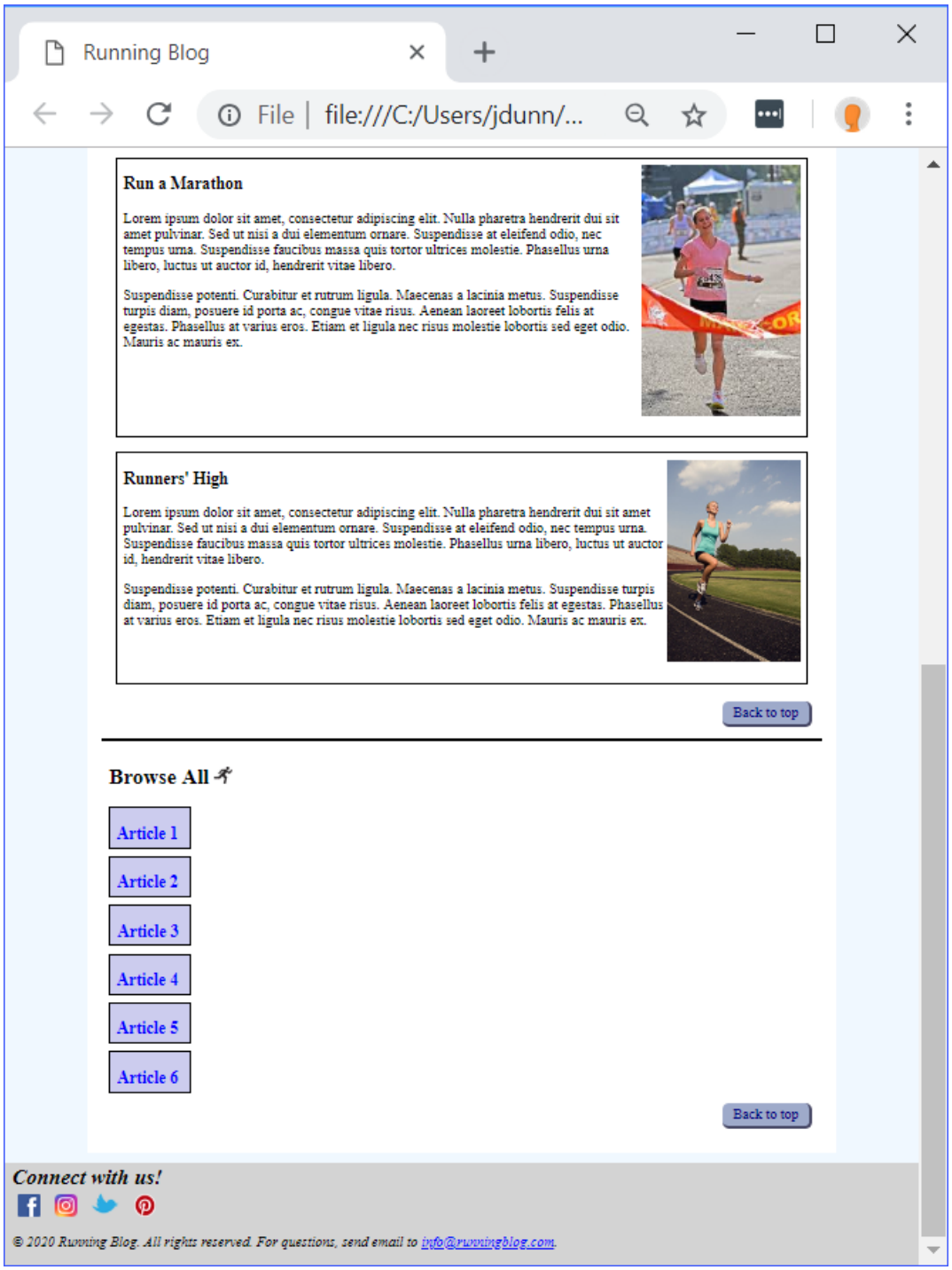
In this exercise, you will position elements in a page called Running Blog that is linked to from Runners Home (from the `read-more` links in the `articles`).

1. Open `Positioning/Exercises/running-blog.html` in your editor. Because this is a new file that you haven't seen before, take some time to go through the HTML to familiarize yourself with the structure. Also note that you have several new images available in the images folder: `runner2.png` and `runner2-tiny.png`.
2. Open `Positioning/Exercises/blog-styles.css`. This is a stylesheet we created for Running Blog. Because this is a new stylesheet that you haven't seen before, take some time to go through the styles that already exist. This will help you when it comes time to add your own styles.
3. Use your new knowledge of positioning to add some styles to this page. The object of this exercise is to practice using the properties covered in this lesson. Also, feel free to add or change any other properties that you know.
4. When you are done, open `running-blog.html` in your browser to see the results. You are welcome to go back to the code and continue to work.

Evaluation Copy

You can design it however you like, or you can try to make it look something like:

EVALUATION COPY: Not to be used in class.


Running Blog
File | file:///C:/Users/jdunn/...
Run a Marathon
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla pharetra hendrerit dui sit amet pulvinar. Sed ut nisi a dui elementum ornare. Suspendisse at eleifend odio, nec tempus urna. Suspendisse faucibus massa quis tortor ultrices molestie. Phasellus urna libero, luctus ut auctor id, hendrerit vitae libero.
Suspendisse potenti. Curabitur et rutrum ligula. Maecenas a lacinia metus. Suspendisse turpis diam, posuere id porta ac, congue vitae risus. Aenean laoreet lobortis felis at egestas. Phasellus at varius eros. Etiam et ligula nec risus molestie lobortis sed eget odio. Mauris ac mauris ex.
Runners' High
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla pharetra hendrerit dui sit amet pulvinar. Sed ut nisi a dui elementum ornare. Suspendisse at eleifend odio, nec tempus urna. Suspendisse faucibus massa quis tortor ultrices molestie. Phasellus urna libero, luctus ut auctor id, hendrerit vitae libero.
Suspendisse potenti. Curabitur et rutrum ligula. Maecenas a lacinia metus. Suspendisse turpis diam, posuere id porta ac, congue vitae risus. Aenean laoreet lobortis felis at egestas. Phasellus at varius eros. Etiam et ligula nec risus molestie lobortis sed eget odio. Mauris ac mauris ex.
Back to top
Browse All
Article 1
Article 2
Article 3
Article 4
Article 5
Article 6
Back to top
Connect with us!
© 2020 Running Blog. All rights reserved. For questions, send email to info@runningblog.com.

Solution: Positioning/Solutions/blog-styles.css

```
1.  html {
2.    font-size: 16px;
3.  }
4.
5.  body {
6.    background: aliceblue;
7.  }
8.
9.  header {
10.   margin: auto;
11.   min-width: 500px;
12.   width: 80%;
13. }
14.
15. #title {
16.   margin: auto;
17.   width: 90%;
18. }
19.
20. header h1 {
21.   margin: auto;
22.   width: 220px;
23. }
24.
25. header p {
26.   margin: auto;
27.   width: 400px;
28. }
29.
30. header nav {
31.   border-radius: 0.4rem;
32.   margin: 0.5rem auto;
33.   padding: 0.2rem 1rem;
34.   width: 140px;
35. }
36.
37. nav,
38. nav ul,
39. nav ol {
40.   margin: 0;
41. }
42.
43. a, a:visited {
44.   color: blue;
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
45.     text-decoration: none;
46.   }
47.
48.   a:hover {
49.     color: purple;
50.   }
51.
52.   a:active {
53.     color: red;
54.   }
55.
56.   main {
57.     background: white;
58.     margin: 1rem auto 8rem;
59.     min-width: 500px;
60.     padding: 1rem;
61.     width: 80%;
62.   }
63.
64.   main nav {
65.     background: lightgray;
66.     border-radius: 0.5rem;
67.     margin-bottom: 1rem;
68.     padding: 0.5rem;
69.   }
70.
71.   section {
72.     border-top: 0.2rem solid black;
73.     padding: 1%;
74.     width: 98%;
75.   }
76.
77.   article {
78.     border: 1px solid black;
79.     margin: 1rem auto;
80.     padding: 0 0.5rem;
81.     width: 96%;
82.   }
83.
84.   article:target h3 {
85.     text-decoration: underline;
86.   }
87.
88.   article img {
89.     float: right;
```

Evaluation Copy

```
90.     margin: 0.5rem 0 0.5rem 0.2rem;
91.   }
92.
93.   article p:last-child::after {
94.     content: "";
95.     display: block;
96.     clear: both;
97.   }
98.
99.   a.back-to-top {
100.    background: rgb(158, 170, 202);
101.    border: rgb(70, 70, 100);
102.    border-radius: 0.5rem;
103.    border-style: none solid solid none;
104.    border-width: 0 0.2rem 0.2rem 0;
105.    color: navy;
106.    display: block;
107.    float: right;
108.    font-size: 1rem;
109.    height: 20px;
110.    margin: 0.2rem 0.2rem 0.2rem 0.5rem;
111.    padding: 0.2rem;
112.    text-decoration: none;
113.    text-align: center;
114.    width: 90px;
115.  }
116.
117.  section::after {
118.    content: "";
119.    display: block;
120.    clear: both;
121.  }
122.
123.  a.back-to-top:active {
124.    color:#cce;
125.    border-width: 0 0.1rem 0.1rem 0;
126.  }
127.
128.  #all article {
129.    background: #cce;
130.    height: 30px;
131.    margin: 0.5rem 0;
132.    overflow: hidden;
133.    padding-bottom: 15px;
134.    width: 75px;
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
135. }
136.
137. #all article:target {
138.   background: white;
139.   height: auto;
140.   width: 80%;
141. }
142.
143. #all article:target h3 a {
144.   color: black;
145. }
146.
147. footer {
148.   background: lightgray;
149.   font-size: medium;
150.   font-style: italic;
151.   min-width: 640px;
152.   width: 100%;
153.   margin: 0 -1rem;
154.   padding: 0.2rem 1rem;
155.   position: fixed;
156.   bottom: 0;
157. }
158.
159. footer h2 {
160.   margin: 0;
161. }
162.
163. footer a {
164.   text-decoration: underline;
165.   margin: 0.4rem;
166. }
167.
168. a[href^='mailto'] {
169.   margin: 0;
170. }
```

Evaluation Copy

Code Explanation

Note that the `footer` is fixed, which is why it is in both screenshots.

Conclusion

In this lesson, you have learned to position elements with the `position`, `top`, `bottom`, `left`, `right`, `z-index`, `float`, and `clear` properties.

Evaluation Copy

EVALUATION COPY: Not to be used in class.

LESSON 11
Transforms and Transitions

Topics Covered

☑ `transition`

☑ `transform`

Introduction

The `transition` property is used to animate changes to an element's style. The `transform` property is used to move, scale, rotate, and skew elements. In this lesson, you will learn to use both.

11.1. transition

The `transition` property provides a way to animate elements between two states. Different states can be set with pseudo-classes (e.g., `:hover`, `:active`, etc.). `transition` is shorthand for `transition-property`, `transition-duration`, `transition-timing-function`, and `transition-delay`.

❖ 11.1.1. transition-property

`transition-property` defines the property or properties to animate. It takes the following values:

1. `all` - The default value. All animatable properties will transition from the first state to the second state.
2. `none` - No properties will transition.
3. A single property (e.g., `transition-property: margin`) or a comma-delimited list of properties (e.g.,`transition-property: margin, padding, color`).

❖ 11.1.2. transition-duration

`transition-duration` sets the amount of time that the transition will take. It takes a positive time value in seconds (`s`) or milliseconds (`ms`). The default is `0s`.

❖ 11.1.3. transition-timing-function

`transition-timing-function` defines how the element transitions through time, meaning its acceleration or deceleration throughout the transition. `transition-timing-function` takes the following values:

1. `ease` - The default value. The transition speeds up toward the middle of the transition and slows down at the end.
2. `linear` - The transition maintains the same speed throughout the transition.
3. `ease-in` - The transition starts slow and accelerates until the end.
4. `ease-out` - The transition start fast and decelerates until the end.
5. `ease-in-out` - The transitions starts slow, accelerates and then decelerates at the end.

Evaluation Copy

`transition-timing-function` also takes the `cubic-bezier()` [33] and `steps()` [34] function values, which give more control over the precise motion of the transition.

❖ 11.1.4. transition-delay

`transition-delay` sets the time to wait before the transition starts. It takes a positive or negative time value in seconds (`s`) or milliseconds (`ms`). If a negative value is set, the transition will start immediately from the place it would be if it had already been running for the given amount of time. The default value is `0s`, which means that the transition starts immediately from the beginning. If a positive value is set, the transition will not start for that amount time.

❖ 11.1.5. transition

The syntax for `transition` is shown below. The highlighted properties are required:

33. https://developer.mozilla.org/en-US/docs/Web/CSS/timing-function#The_cubic-bezier()_class_of_timing_functions
34. https://developer.mozilla.org/en-US/docs/Web/CSS/timing-function#The_steps()_class_of_timing_functions

EVALUATION COPY: Not to be used in class.

```
transition: transition-property
            transition-duration
            transition-timing-function
            transition-delay;
```

If multiple transitions are specified, they are separated in a comma-delimited list like in the example below:

Evaluation Copy

```
transition: margin 2s ease-in, color 4s;
```

You can see practical uses of transitions in the demo below:

Demo 11.1: TransformsTransitions/Demos/transition-styles.css

```
1.   body {
2.     background-color: aliceblue;
3.   }
4.
5.   header {
6.     background: rgb(214, 214, 214);
7.     border: black solid 1px;
8.     border-radius: 1rem;
9.     margin: auto;
10.    padding: 1rem;
11.    width: 50%;
12.  }
13.
14.  header nav a {
15.    background: rgb(158, 170, 202);
16.    border: rgb(70, 70, 100);
17.    border-radius: 0.5rem;
18.    border-style: none solid solid none;
19.    border-width: 0 0.2rem 0.2rem 0;
20.    color: navy;
21.    display: block;
22.    font-size: 1.1rem;
23.    height: 20px;
24.    margin: 0.2rem;
25.    padding: 0.2rem;
26.    text-decoration: none;
27.    width: 100px;
28.  }
29.
30.  header nav a:hover {
31.    background: whitesmoke;
32.    transition: background 1s,
33.      height 0.5s, width 0.5s;
34.    width: 120px;
35.    height: 30px;
36.  }
37.
38.  header nav a:active {
39.    color:#cce;
40.    border-width: 0 0.1rem 0.1rem 0;
41.  }
42.
43.  main {
44.    background: white;
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
45.     width: 75%;
46.     margin: auto;
47.   }
48.
49.   article {
50.     background: rgb(155, 170, 170);
51.     border: black solid 1px;
52.     border-radius: 0.5rem;
53.     padding: 0.2rem 1rem 0.5rem;
54.     margin: 0.5rem auto;
55.     overflow: hidden;
56.     height: 75px;
57.   }
58.
59.   article:target {
60.     background: white;
61.     height: 200px;
62.     transition: all 1.5s ease-out;
63.   }
```

Evaluation Copy

Demo 11.2: TransformsTransitions/Demos/transition.html

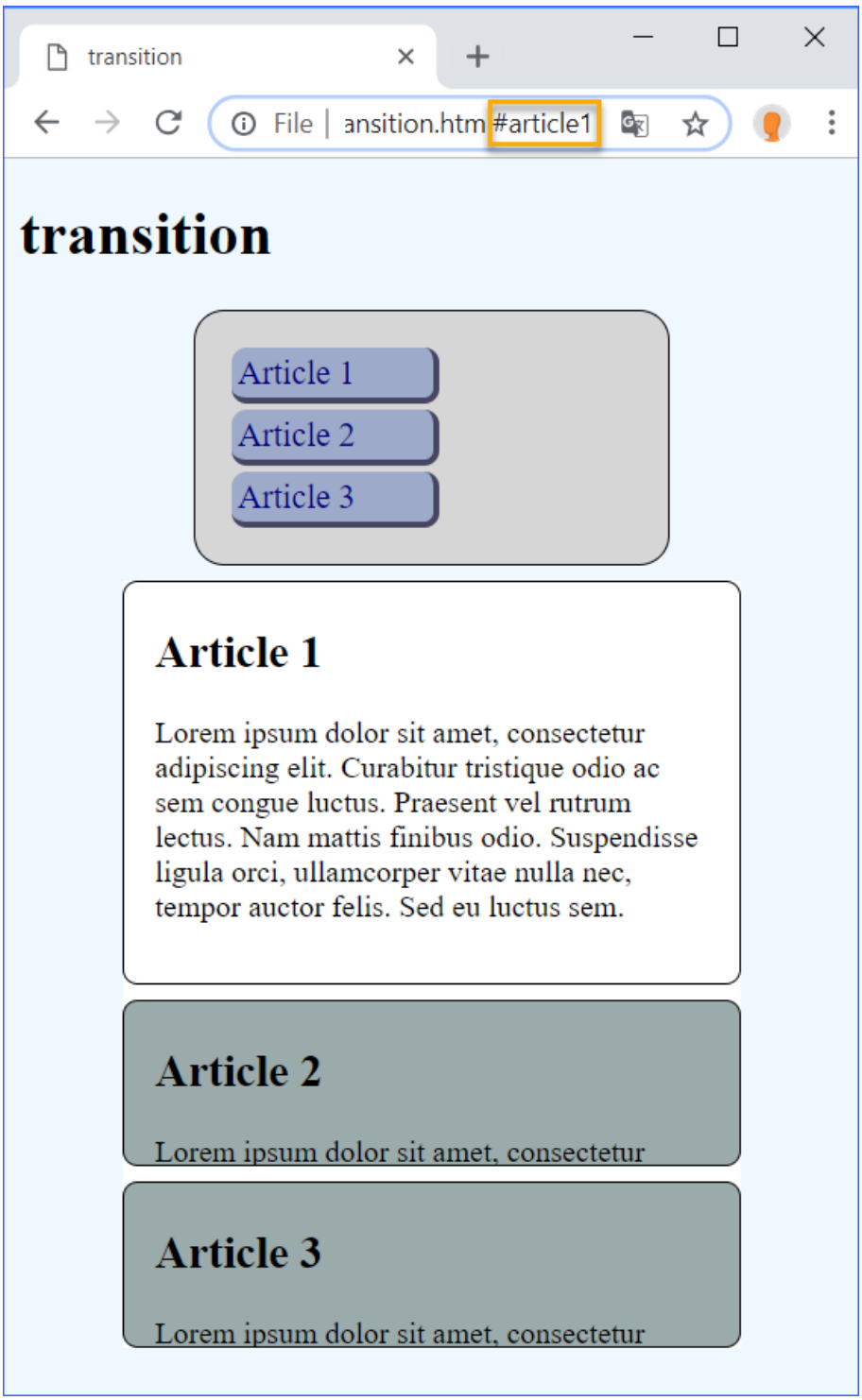
```
      -------Lines 1 through 11 Omitted-------
12.   <header>
13.     <nav>
14.       <a href="#article1">Article 1</a>
15.       <a href="#article2">Article 2</a>
16.       <a href="#article3">Article 3</a>
17.     </nav>
18.   </header>
19.   <main>
20.     <article id="article1">
21.       <h2>Article 1</h2>
22.       <p>
23.         Lorem ipsum dolor
24.         sit amet, consectetur adipiscing elit.
25.         Curabitur tristique odio ac sem congue luctus.
26.         Praesent vel rutrum lectus.
      -------Lines 27 through 29 Omitted-------
30.       </p>
31.     </article>
32.     <article id="article2">
33.       <h2>Article 2</h2>
34.       <p>
35.         Lorem ipsum dolor
36.         sit amet, consectetur adipiscing elit.
37.         Curabitur tristique odio ac sem congue luctus.
38.         Praesent vel rutrum lectus.
      -------Lines 39 through 41 Omitted-------
42.       </p>
43.     </article>
44.     <article id="article3">
45.       <h2>Article 3</h2>
46.       <p>
47.         Lorem ipsum dolor
48.         sit amet, consectetur adipiscing elit.
49.         Curabitur tristique odio ac sem congue luctus.
50.         Praesent vel rutrum lectus.
      -------Lines 51 through 53 Omitted-------
54.       </p>
55.     </article>
56.   </main>
      -------Lines 57 through 58 Omitted-------
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

Code Explanation

The code above will render the following:



Some things to note:

1. You will need to open `TransformsTransitions/Demos/transition.html` in the browser and interact with the page to see the transition effects.
2. We added transition effects to the links at the top when they are hovered over.
3. We added transition effects to the `article` elements when they are the target. `article1` is the target in the image above.

11.2. transform

The `transform` property provides ways to rotate, scale, skew, and translate block elements. It can be combined with `transition` to create cool animation effects. `transform` takes several transform function values. It can take one or more of the following functional notations in the same rule:

Evaluation Copy

1. `rotate()`
2. `scale()`
3. `skew()`
4. `translate()`

`transform` also takes the keyword `none`, which means no transforms should be applied.

matrix() and 3D Transforms

The `matrix()` function[35] is an all-in-one function that can do any transform, but its notation is exceedingly complicated, is not author-friendly, and is not as commonly used as the other functions.

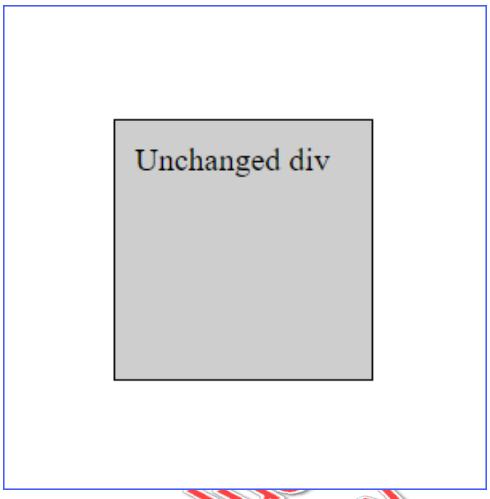
`transform` also takes 3D function values[36] used to create 3D transforms. These are not commonly used.

35. https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function/matrix
36. https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function#Syntax

EVALUATION COPY: Not to be used in class.

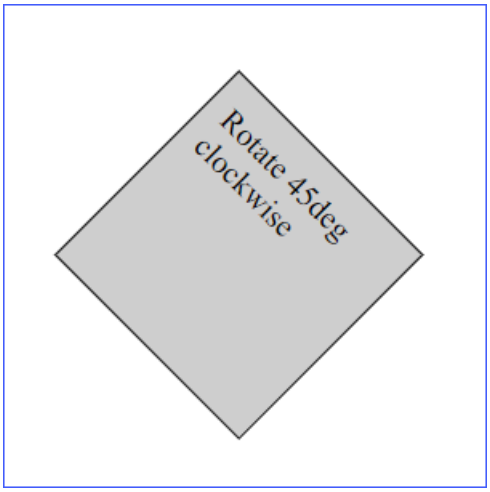
❖ 11.2.1. Rotation

The `rotate()` function takes a positive or negative angle value in degrees (`deg`), gradians (`grad`), radians (`rad`), or turns (`turn`). Positive values rotate the element clockwise and negative values rotate the element counterclockwise. You can see some examples below. We will be changing this 130px by 130px `div` element:

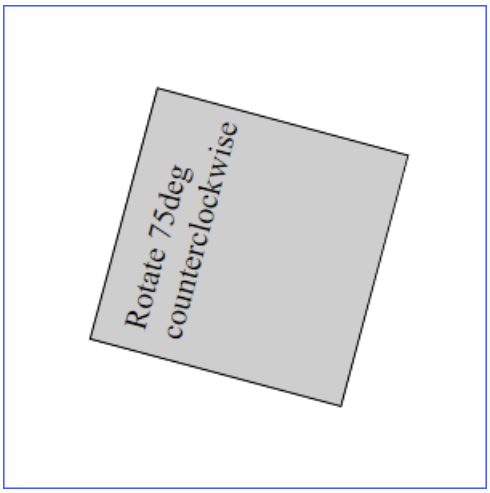


Evalu
Copy

```
transform: rotate(45deg);
```




```
transform: rotate(-75deg)
```



These examples can be found in `TransformsTransitions/Demos/transform-examples.html`.

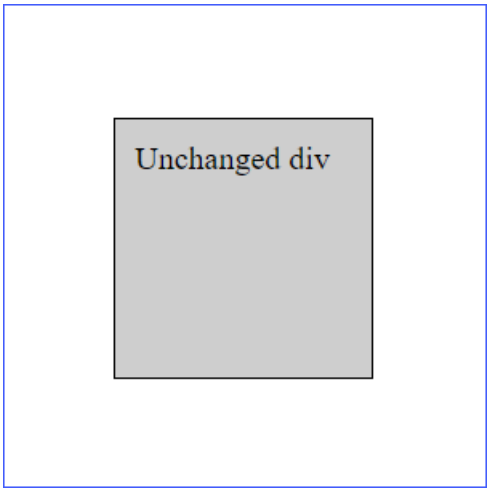
❖ 11.2.2. Scale

The `scale()` function takes one or two number values without units. If negative values are used, the element is flipped.

Evaluation Copy

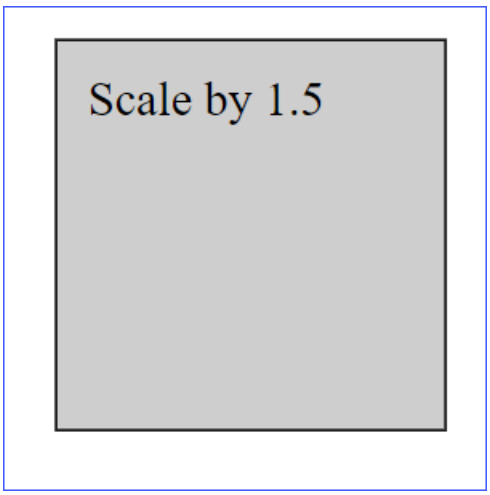
1. If one value is used, it is the scaling factor both horizontally and vertically.
2. If two values are used, the first is the scaling factor horizontally and the second is the scaling factor vertically.

You can see some examples below. We will be changing this 130px by 130px `div` element:

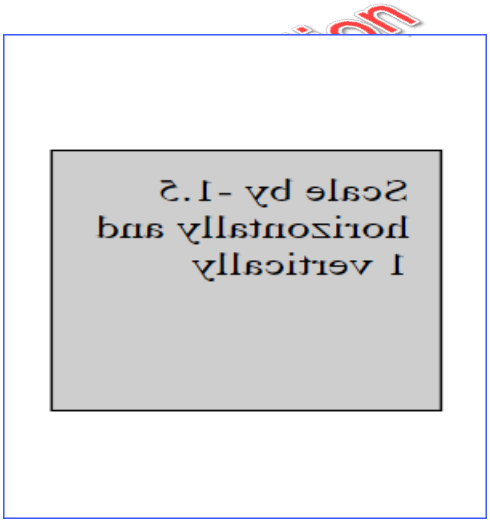


EVALUATION COPY: Not to be used in class.

```
transform: scale(1.5);
```




```
transform: scale(-1.5, 1)
```



These examples can be found in `TransformsTransitions/Demos/transform-examples.html`.

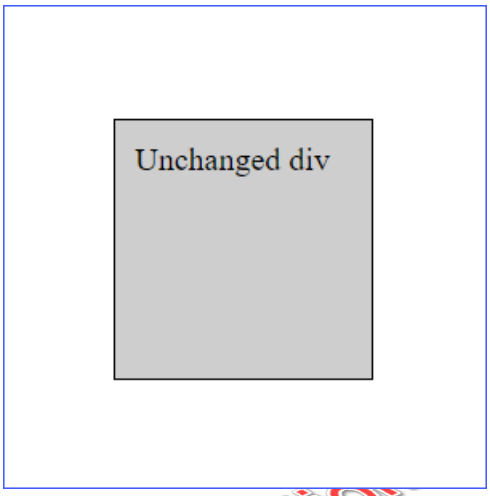
❖ 11.2.3. Skew

The `skew()` function takes one or two positive or negative angle values in degrees (`deg`), gradians (`grad`), radians (`rad`), or turns (`turn`).

1. If one value is used, it skews the element horizontally.

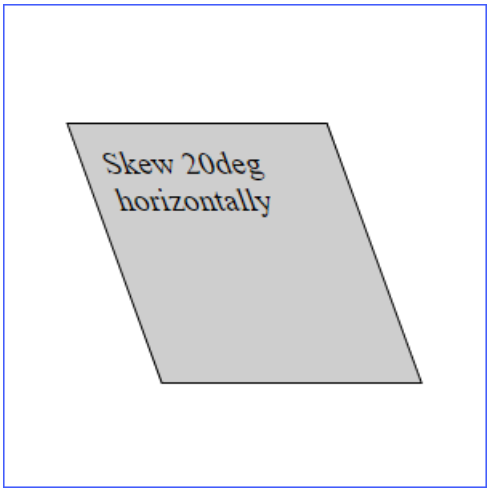
2. If two values are used, the first value skews the element horizontally, and the second value skews the element vertically.

You can see some examples below. We will be changing this 130px by 130px `div` element:



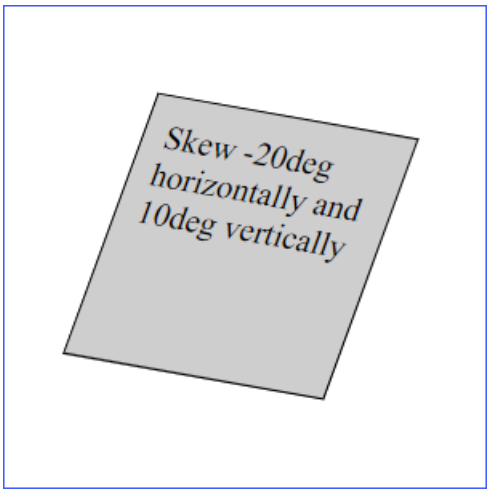

```
transform: skew(20deg);
```

Evaluation Copy




```
transform: skew(-20deg, 10deg)
```


EVALUATION COPY: Not to be used in class.



These examples can be found in `TransformsTransitions/Demos/transform-examples.html`.

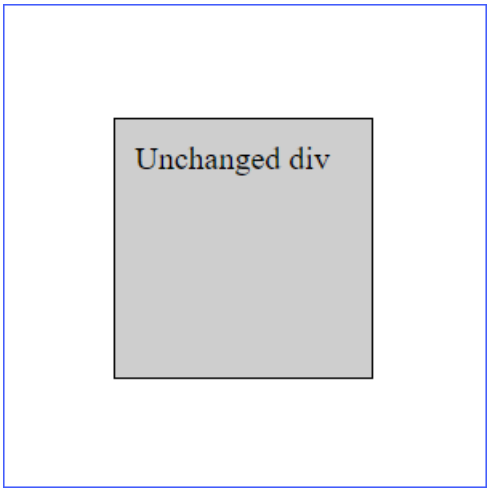
❖ 11.2.4. Translate

The `translate()` function takes one or two positive or negative length values. Positive values move the element right or down, and negative values move the element left or up.

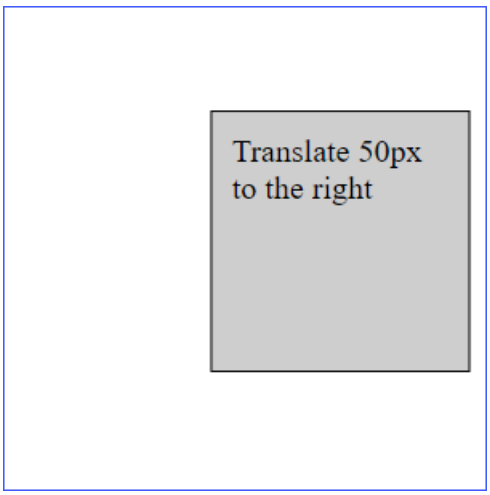
Evaluation Copy

1. If one value is used, it moves the element horizontally.
2. If two values are used, the first value moves the element horizontally, and the second value moves the element vertically.

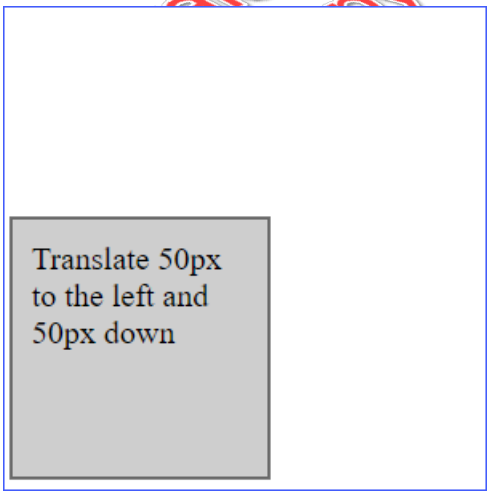
You can see some examples below. We will be changing this 130px by 130px `div` element:



```
transform: translate(50px);
```




```
transform: transition(-50px, 50px)
```



These examples can be found in `TransformsTransitions/Demos/transform-examples.html`.

You can see more uses of `transform` in the following demo. Note that you will also see more examples of `transition` and see `transform` and `transition` used together.

EVALUATION COPY: Not to be used in class.

Demo 11.3: TransformsTransitions/Demos/transform-styles.css

```
1.   body {
2.     background-color: aliceblue;
3.   }
4.
5.   article {
6.     background: rgb(155, 170, 170);
7.     border: black solid 1px;
8.     border-radius: 0.5rem;
9.     padding: 0 0.5rem 0.2rem;
10.    margin: 0.5rem;
11.    overflow: hidden;
12.    height: 50px;
13.    width: 100px;
14.  }
15.
16.  article a:visited {
17.    color: blue;
18.  }
19.
20.  article img {
21.    display: block;
22.    left: -5px;
23.    bottom: 10px;
24.    opacity: 0;
25.    position: relative;
26.    margin: -1rem;
27.  }
28.
29.  article a.read-more {
30.    display: none;
31.  }
32.
33.  article:target {
34.    background: white;
35.    height: auto;
36.    transform: rotate(-360deg);
37.    transition: background 2s,
38.      transform 0.7s ease-out;
39.    width: 300px;
40.  }
41.
42.  article:target h2 a {
43.    color: black;
44.    text-decoration: none;
```

Evaluation Copy

```
45.   }
46.
47.   article:target img {
48.     opacity: 1;
49.     left: 250px;
50.     transform: rotate(720deg);
51.     transition: opacity 3s 0.5s,
52.       transform 1s 0.8s linear,
53.       left 1s 0.8s linear;
54.   }
55.
56.   article:target img + a.read-more {
57.     background: rgba(166, 212, 252, 0.4);
58.     border-radius: 0.5rem;
59.     box-sizing: border-box;
60.     display: block;
61.     opacity: 0;
62.     position: absolute;
63.     left: 240px;
64.     bottom: 2px;
65.     width: 50px;
66.     height: 45px;
67.     padding: 0 10px;
68.     text-decoration: none;
69.     transform: scale(0.4);
70.     transition: all 1s;
71.   }
72.
73.   article:target img + a.read-more:hover {
74.     opacity: 1;
75.     transform: scale(1.1);
76.     transition: all 1s;
77.   }
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

Demo 11.4: TransformsTransitions/Demos/transform.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.  <h1>transform</h1>
12.  <main>
13.      <article id="article1">
14.        <h2><a href="#article1">Article 1</a></h2>
15.        <p>
16.          Lorem ipsum dolor
17.          sit amet, consectetur adipiscing elit.
18.          Curabitur tristique odio ac sem congue luctus.
19.          Praesent vel rutrum lectus.
     -------Lines 20 through 22 Omitted-------
23.        </p>
24.          <img alt="Books" src="images/books.png">
25.          <a href="article1.html" class="read-more">Read more!</a>
26.      </article>
27.      <article id="article2">
28.        <h2><a href="#article2">Article 2</a></h2>
29.        <p>
30.          Lorem ipsum dolor
31.          sit amet, consectetur adipiscing elit.
32.          Curabitur tristique odio ac sem congue luctus.
33.          Praesent vel rutrum lectus.
     -------Lines 34 through 36 Omitted-------
37.        </p>
38.          <img alt="Books" src="images/books.png">
39.          <a href="article2.html" class="read-more">Read more!</a>
40.      </article>
41.      <article id="article3">
42.        <h2><a href="#article3">Article 3</a></h2>
43.        <p>
44.          Lorem ipsum dolor
45.          sit amet, consectetur adipiscing elit.
46.          Curabitur tristique odio ac sem congue luctus.
47.          Praesent vel rutrum lectus.
     -------Lines 48 through 50 Omitted-------
51.        </p>
52.          <img alt="Books" src="images/books.png">
53.          <a href="article3.html" class="read-more">Read more!</a>
54.      </article>
55.      <article id="article4">
56.        <h2><a href="#article4">Article 4</a></h2>
57.        <p>
```

Evaluation Copy

```
58.          Lorem ipsum dolor
59.          sit amet, consectetur adipiscing elit.
60.          Curabitur tristique odio ac sem congue luctus.
61.          Praesent vel rutrum lectus.
     -------Lines 62 through 64 Omitted-------
65.        </p>
66.          <img alt="Books" src="images/books.png">
67.          <a href="article4.html" class="read-more">Read more!</a>
68.      </article>
69.  </main>
70.  </body>
71.  </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:

EVALUATION COPY: Not to be used in class.

Some things to note:

1. You will need to open `TransformsTransitions/Demos/transform.html` in the browser and interact with the page to see the effect of these styles.
2. We added `transform` and `transition` effects to the `article` elements when they are the target.
3. We added `transform` and `transition` effects to the `img` and `a` elements within the `article` elements.

4. Websites rarely use this many transforms and transitions; however, the demo gives you a sense of what can be accomplished with them.

Evaluation Copy

EVALUATION COPY: Not to be used in class.

Exercise 13: Transforms and Transitions

25 to 40 minutes

In this exercise, you will continue to work with the Running Blog page.

1. Open `TransformsTransitions/Exercises/running-blog.html` in your editor.
2. Open `TransformsTransitions/Exercises/blog-styles.css`. This is the stylesheet from the solution to the Positioning exercise. Feel free to continue working from your own stylesheet.
3. Using your new knowledge of transforms and transitions add some styles to this page. The object of this exercise is to practice using the properties covered in this lesson. Also, feel free to add or change any other properties that you know.
4. When you are done, open `running-blog.html` in your browser to see the results. You are welcome to go back to the code and continue to work.

Evaluation Copy

You can design it however you want, or you can try to:

1. Make the tiny runner move right, flip and come back when it is within the target.
2. Skew the `back-to-top` links when they are hovered over.
3. Transition the `width` and `color` of the `articles` in the `all section` when they are the target.

Solution: TransformsTransitions/Solutions/blog-styles.css

```
1.  html {
2.    font-size: 16px;
3.  }
4.
5.  body {
6.    background: aliceblue;
7.  }
8.
9.  header {
10.   margin: auto;
11.   min-width: 500px;
12.   width: 80%;
13. }
14.
15. #title {
16.   margin: auto;
17.   width: 90%;
18. }
19.
20. header h1 {
21.   margin: auto;
22.   width: 220px;
23. }
24.
25. header h1 img {
26.   left: 0;
27.   position: relative;
28. }
29.
30. header p {
31.   margin: auto;
32.   width: 400px;
33. }
34.
35. header nav {
36.   border-radius: 0.4rem;
37.   margin: 0.5rem auto;
38.   padding: 0.2rem 1rem;
39.   width: 140px;
40. }
41.
42. nav,
43. nav ul,
44. nav ol {
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
45.     margin: 0;
46.   }
47.
48.   a, a:visited {
49.     color: blue;
50.     text-decoration: none;
51.   }
52.
53.   a:hover {
54.     color: purple;
55.   }
56.
57.   a:active {
58.     color: red;
59.   }
60.
61.   main {
62.     background: white;
63.     margin: 1rem auto 8rem;
64.     min-width: 500px;
65.     padding: 1rem;
66.     width: 80%;
67.   }
68.
69.   main nav {
70.     background: lightgray;
71.     border-radius: 0.5rem;
72.     margin-bottom: 1rem;
73.     padding: 0.5rem;
74.   }
75.
76.   section {
77.     border-top: 0.2rem solid black;
78.     padding: 1%;
79.     width: 98%;
80.   }
81.
82.   section h2 img {
83.     left: 0;
84.     position: relative;
85.   }
86.
87.   div:target h1 img,
88.   section:target h2 img {
89.     left: 100px;
```

Evaluation Copy

```
90.     position: relative;
91.     transform: scale(-1, 1) translate(100px);
92.     transition: left 1s, transform 1s 1s linear;
93.   }
94.
95.   article {
96.     border: 1px solid black;
97.     margin: 1rem auto;
98.     padding: 0 0.5rem;
99.     width: 96%;
100.  }
101.
102.  article:target h3 {
103.    text-decoration: underline;
104.  }
105.
106.  article img {
107.    float: right;
108.    margin: 0.5rem 0 0.5rem 0.2rem;
109.  }
110.
111.  article p:last-child::after {
112.    content: "";
113.    display: block;
114.    clear: both;
115.  }
116.
117.  a.back-to-top {
118.    background: rgb(158, 170, 202);
119.    border: rgb(70, 70, 100);
120.    border-radius: 0.5rem;
121.    border-style: none solid solid none;
122.    border-width: 0 0.2rem 0.2rem 0;
123.    color: navy;
124.    display: block;
125.    float: right;
126.    font-size: 1rem;
127.    height: 20px;
128.    margin: 0.2rem 0.2rem 0.2rem 0.5rem;
129.    padding: 0.2rem;
130.    text-decoration: none;
131.    text-align: center;
132.    width: 90px;
133.  }
134.
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
135. section::after {
136.   content: "";
137.   display: block;
138.   clear: both;
139. }
140.
141. a.back-to-top:hover {
142.   transform: skew(-10deg);
143.   transition: 0.2s;
144. }
145.
146. a.back-to-top:active {
147.   color:#cce;
148.   border-width: 0 0.1rem 0.1rem 0;
149. }
150.
151. #all article {
152.   background: #cce;
153.   height: 30px;
154.   margin: 0.5rem 0;
155.   overflow: hidden;
156.   padding-bottom: 15px;
157.   width: 75px;
158. }
159.
160. #all article:target {
161.   background: white;
162.   height: auto;
163.   width: 80%;
164.   transition: all 0.5s linear;
165. }
166.
167. #all article:target h3 a {
168.   color: black;
169. }
170.
171. footer {
172.   background: lightgray;
173.   font-size: medium;
174.   font-style: italic;
175.   min-width: 640px;
176.   width: 100%;
177.   margin: 0 -1rem;
178.   padding: 0.2rem 1rem;
179.   position: fixed;
```

Evaluation Copy

```
180.   bottom: 0;
181. }
182.
183. footer h2 {
184.   margin: 0;
185. }
186.
187. footer a {
188.   text-decoration: underline;
189.   margin: 0.4rem;
190. }
191.
192. footer a[href^='mailto'] {
193.   margin: 0;
194. }
```

Evaluation Copy

Code Explanation

You will need to open `TransformsTransitions/Solutions/running-blog.html` in the browser and interact with the page to see the transform and transition effects we added.

Conclusion

In this lesson, you have learned how to use transitions to animate changes to an element's style and transforms to move, scale, rotate, and skew elements.

EVALUATION COPY: Not to be used in class.

LESSON 12
Layouts

Topics Covered

☑ CSS Flexbox Layout.

☑ CSS Grid Layout.

☑ CSS Multi-column Layout.

Introduction

CSS provides multiple layout systems that can be mixed and matched to give incredible control over the layout of your pages. We will be briefly looking at three of them:

1. Flexible Box Layout Module
2. Grid Layout
3. Multi-column Layout

Evaluation Copy

12.1. Introduction to Flexible Box Layout Module

Layouts and Browser Compatibility

The layouts that we will be looking at in this lesson are newer technologies in CSS, and therefore are not fully supported in all browsers.

Make sure to take advantage of caniuse.com[37] and Mozilla[38] to check which browsers support these layouts, which properties they support, and what tricks there are to get around these differences.

37. https://caniuse.com/
38. https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Supporting_Older_Browsers

The CSS Flexible Box Layout Module (or "Flexbox", as it's commonly known) offers a handy way to control the horizontal and vertical arrangement of elements on a page. As the W3C description[39] states:

> In the flex layout model, the children of a flex container can be laid out in any direction, and can "flex" their sizes, either growing to fill unused space or shrinking to avoid overflowing the parent. Both horizontal and vertical alignment of the children can be easily manipulated.

❖ 12.1.1. Some Flexbox Properties

Below are some common Flexbox properties:

Flexbox Properties

Property	Explanation
`display`	To create flexible boxes, the `display` of the parent element of the elements you wish to turn into flexible boxes must be `flex`.
`flex-direction`	`flex-direction` sets the direction of the flex container to be either horizontal or vertical. If it is horizontal the child flexible boxes are side by side. If it is vertical, the child flexible boxes are placed on top of each other.
`flex-wrap`	`flex-wrap` sets whether or not the flexible boxes will wrap to a new line if they overflow their container.
`flex-flow`	`flex-flow` is shorthand for `flex-direction` and `flex-wrap`.
`flex`	`flex` is shorthand for the `flex-grow`, `flex-shrink`, and `flex-basis` properties and is used to set how flexible boxes take up the space of their container.

Evaluation Copy

Here are some further explanations with examples:

flex-flow

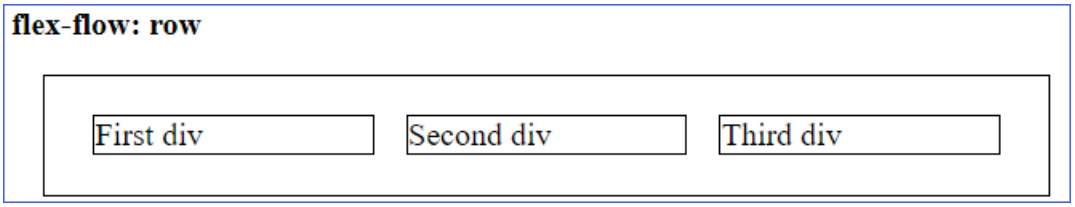
`flex-flow` takes one value for either `flex-direction` or `flex-wrap`, or two values for both together (order doesn't matter).

Common `flex-direction` values are:

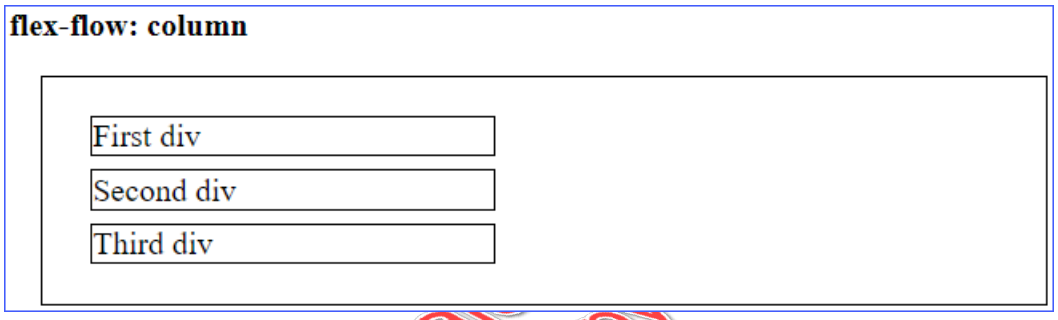
39. https://www.w3.org/TR/css-flexbox-1/#abstract

EVALUATION COPY: Not to be used in class.

1. `row` - child flexible boxes are side by side.



2. `column` - child flexible boxes are stacked on top of each other.

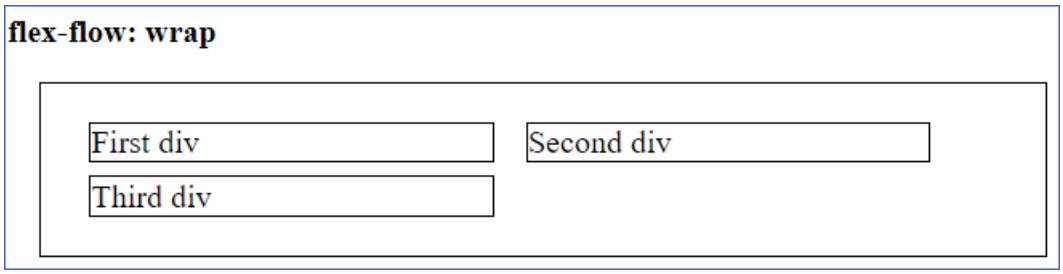


Eval Copy

`flex-direction` [40] also has takes `row-reverse` and `column-reverse` values to display the elements in reverse order.

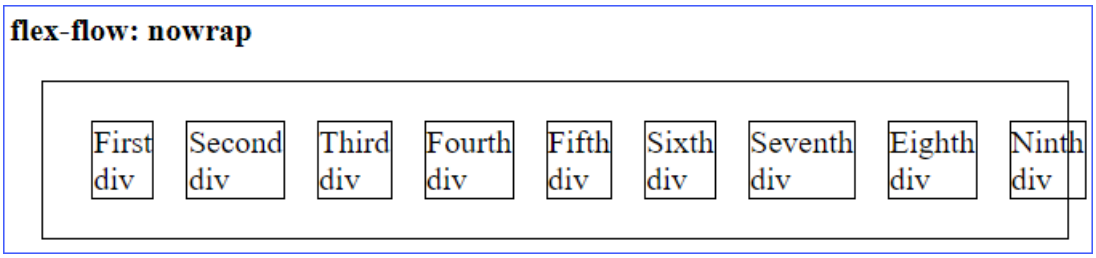
Common `flex-wrap` values are:

1. `wrap` - Flexible boxes wrap if there is not enough space.



2. `nowrap` - Flexible boxes extend outside of their container if there is not enough space.

40. https://developer.mozilla.org/en-US/docs/Web/CSS/flex-direction



`flex-wrap` [41] also has takes a `wrap-reverse` value to display the elements in reverse order.

These examples can be found in `Layouts/Demos/flex-flow-examples.html`.

flex

Evaluation Copy

`flex` is shorthand for the `flex-grow`, `flex-shrink`, and `flex-basis` properties, but generally just takes one of the following values:

1. `initial` - The default value. The flexible box is sized by its `width` and `height` properties and shrinks to its minimum size to fit in its container.
2. `auto` - The flexible box is sized by its `width` and `height` properties, but grows to take up any extra space available in its container.
3. `none` - The flexible box is inflexible. It is sized by its `width` and `height` properties, but does not shrink or grow in relation to its container.
4. A positive, unitless number - The flexible box takes up the specified proportion of its container. For example, if there are two boxes, one with `flex` set to `1` and one with `flex` set to `2`, then the box with `flex: 2` will take up twice as much space as the other one.

Below you can see a basic Flexbox demo:

41. https://developer.mozilla.org/en-US/docs/Web/CSS/flex-wrap

EVALUATION COPY: Not to be used in class.

Demo 12.1: Layouts/Demos/flexbox-styles.css

```
1.   body {
2.     background: aliceblue;
3.   }
4.
5.   main {
6.     background: white;
7.     width: 98%;
8.     margin: auto;
9.   }
10.
11.  section {
12.    border: 1px solid black;
13.    display: flex;
14.    flex-flow: row wrap;
15.    margin: 0.5rem auto;
16.  }
17.
18.  section article {
19.    border: 1px solid black;
20.    border-radius: 0.5rem;
21.    flex: 1;
22.    padding: 0.5rem;
23.    margin: 0.5rem;
24.    min-width: 200px;
25.  }
26.
27.  h2 {
28.    width: 100%;
29.    margin-left: 0.5rem;
30.  }
31.
32.  section article:target {
33.    flex: 3;
34.  }
35.
36.  section article:target h3 {
37.    text-decoration: underline;
38.  }
```

Evaluation Copy

Demo 12.2: Layouts/Demos/flexbox.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.  <h1>Basic Flexbox</h1>
12.  <nav>
13.    <ol>
14.      <li><a href="#article1">Article 1</a></li>
15.      <li><a href="#article2">Article 2</a></li>
16.      <li><a href="#article3">Article 3</a></li>
17.      <li><a href="#article4">Article 4</a></li>
18.      <li><a href="#article5">Article 5</a></li>
19.      <li><a href="#article6">Article 6</a></li>
20.    </ol>
21.  </nav>
22.  <main>
23.    <section id="articles">
24.      <h2>Articles</h2>
25.      <article id="article1">
26.        <h3>Article 1</h3>
27.        <p>Lorem ipsum dolor
28.          sit amet, consectetur adipiscing elit.
29.          Curabitur tristique odio ac sem congue luctus.
30.          Praesent vel rutrum lectus.
     -------Lines 31 through 34 Omitted-------
35.      </article>
36.      <article id="article2">
37.        <h3>Article 2</h3>
38.        <p>Lorem ipsum dolor
39.          sit amet, consectetur adipiscing elit.
40.          Curabitur tristique odio ac sem congue luctus.
41.          Praesent vel rutrum lectus.
     -------Lines 42 through 45 Omitted-------
46.      </article>
47.      <article id="article3">
48.        <h3>Article 3</h3>
49.        <p>Lorem ipsum dolor
50.          sit amet, consectetur adipiscing elit.
51.          Curabitur tristique odio ac sem congue luctus.
52.          Praesent vel rutrum lectus.
     -------Lines 53 through 56 Omitted-------
57.      </article>
58.      <article id="article4">
59.        <h3>Article 4</h3>
60.        <p>Lorem ipsum dolor
```

Evaluation Copy

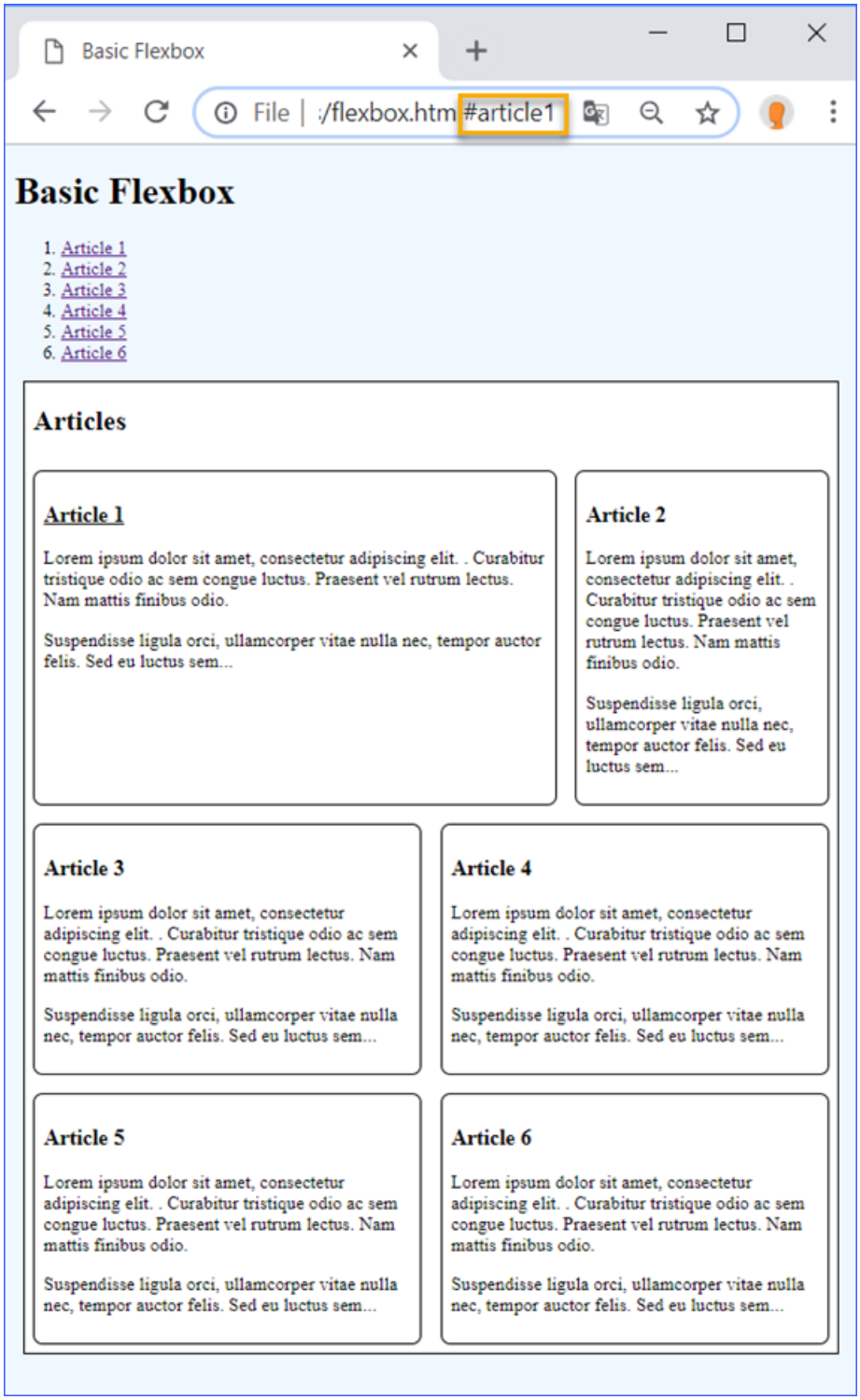
EVALUATION COPY: Not to be used in class.

```
61.           sit amet, consectetur adipiscing elit.
62.           Curabitur tristique odio ac sem congue luctus.
63.           Praesent vel rutrum lectus.
      -------Lines 64 through 67 Omitted-------
68.       </article>
69.       <article id="article5">
70.         <h3>Article 5</h3>
71.         <p>Lorem ipsum dolor
72.           sit amet, consectetur adipiscing elit.
73.           Curabitur tristique odio ac sem congue luctus.
74.           Praesent vel rutrum lectus.
      -------Lines 75 through 78 Omitted-------
79.       </article>
80.       <article id="article6">
81.         <h3>Article 6</h3>
82.         <p>Lorem ipsum dolor
83.           sit amet, consectetur adipiscing elit.
84.           Curabitur tristique odio ac sem congue luctus.
85.           Praesent vel rutrum lectus.
      -------Lines 86 through 89 Omitted-------
90.       </article>
91.     </section>
92.   </main>
93.   </body>
94.   </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:



Some things to note:

EVALUATION COPY: Not to be used in class.

1. You will need to open `Layouts/Demos/flexbox.html` in the browser and click the article links to see the effect.
2. The `articles` are the flexible boxes. They are set to be in rows and to wrap. They are set to have a `flex` value of `1` by default.
3. The target `article` is given a `flex` value of `3`. `article1` is the target in this image.

12.2. Introduction to Grid Layout

Evaluation Copy

Grid offers a convenient, powerful method for laying out elements on a page. As the W3C specification states[42]:

> This CSS module defines a two-dimensional grid-based layout system, optimized for user interface design. In the grid layout model, the children of a grid container can be positioned into arbitrary slots in a predefined flexible or fixed-size layout grid.

❖ 12.2.1. Some Grid Properties

Below are some common Grid properties:

42. https://www.w3.org/TR/css-grid-1/#abstract

Grid Properties

Property	Explanation
`display`	To create a grid, the `display` of the parent element of the elements you wish to layout in a grid must be `grid`.
`grid-template-columns`	`grid-template-columns` sets the size and number of columns in the grid container.
`grid-template-rows`	`grid-template-rows` sets the size and number of rows in the grid container.
`grid-auto-columns`	If `grid-template-rows` is used to create rows, then columns are generated implicitly (automatically). `grid-auto-columns` sets the size of these implicitly-generated columns if they are not explicitly defined by the `grid-template-columns` property.
`grid-auto-rows`	If `grid-template-columns` is used to create columns, then rows are generated implicitly (automatically). `grid-auto-rows` sets the size of these implicitly-generated rows if they are not explicitly defined by the `grid-template-rows` property.
`gap`	`gap` sets the space between rows and columns.
`grid-column`	`grid-column` sets an element's horizontal position and size by specifying where it starts and ends along the columns in the grid container.
`grid-row`	`grid-row` sets an element's vertical position and size by specifying where it starts and ends along the rows in the grid container.

Evaluation Copy

Here are some further explanations with examples:

grid-template-columns and grid-template-rows

Some common `grid-template-columns` and `grid-template-rows` values are:

1. A non-negative length values in units (e.g., `200px`).
2. A percentage value relative to the size of the container grid (e.g., `30%`).
3. A non-negative flex value in `fr` units (e.g., `3fr`) - Similar to the unitless values for Flexbox's `flex` property. It specifies the proportion of space that the columns or rows take up.
4. `minmax(min, max)` [43] - Defines the size of the column or row ranging from the `min` to the `max` (e.g., `minmax(100px, 1fr)`).

43. https://developer.mozilla.org/en-US/docs/Web/CSS/minmax

EVALUATION COPY: Not to be used in class.

5. `repeat(number, size)` [44] - Creates the `number` of columns or rows sized by the `size` argument (e.g., `repeat(5, 1fr)`).
6. `auto` - The element takes up the most space it can and doesn't shrink smaller than what is specified in `min-width` and `min-height` properties.

The syntax of `grid-template-columns` and `grid-template-rows` is shown below:

```
grid-template-columns: column1, column2, column3, etc.

grid-template-rows: row1, row2, row3, etc.
```

The following example creates a grid with a `40%` column, three `1fr` columns, and a `1fr` column that won't shrink below `150px`:

Evaluation Copy

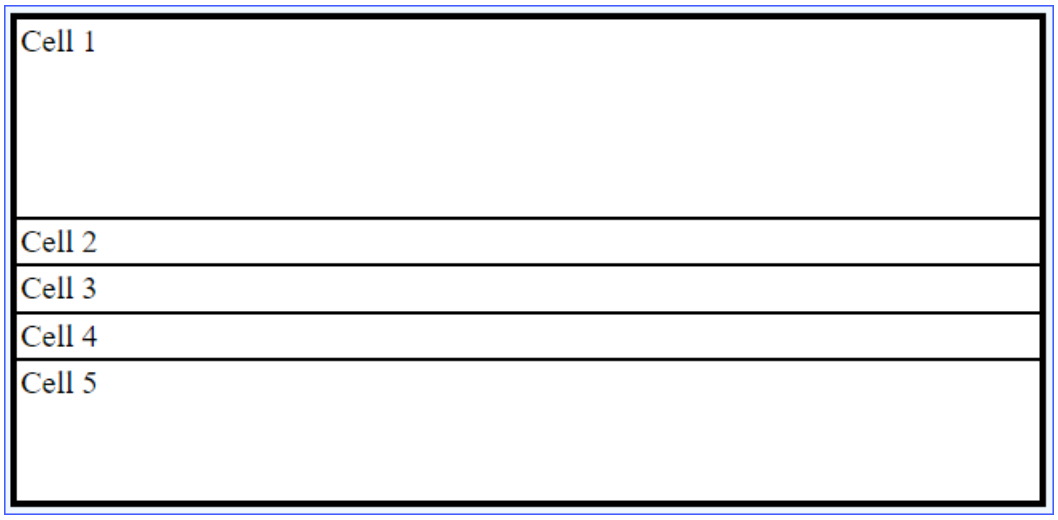
```
grid-template-columns: 40% repeat(3, 1fr) minmax(150px, 1fr);
```

Cell 1	Cell 2	Cell 3	Cell 4	Cell 5
Cell 6	Cell 7	Cell 8	Cell 9	Cell 10
Cell 11	Cell 12	Cell 13	Cell 14	Cell 15
Cell 16	Cell 17	Cell 18	Cell 19	Cell 20

The following example creates a grid with a `100px` row, three `1fr` rows, and a `3fr` row.

```
grid-template-rows: 100px repeat(3, 1fr) 3fr;
```

44. https://developer.mozilla.org/en-US/docs/Web/CSS/repeat



These examples can be found in `Layouts/Demos/grid-examples.html`.

grid-auto-columns and grid-auto-rows

Evaluation Copy

`grid-auto-columns` and `grid-auto-rows` are very similar to `grid-template-columns` and `grid-template-rows`, but since they are targeting the implicitly-generated columns and rows, they only set size and not number.

Some common `grid-auto-columns` and `grid-auto-rows` values are:

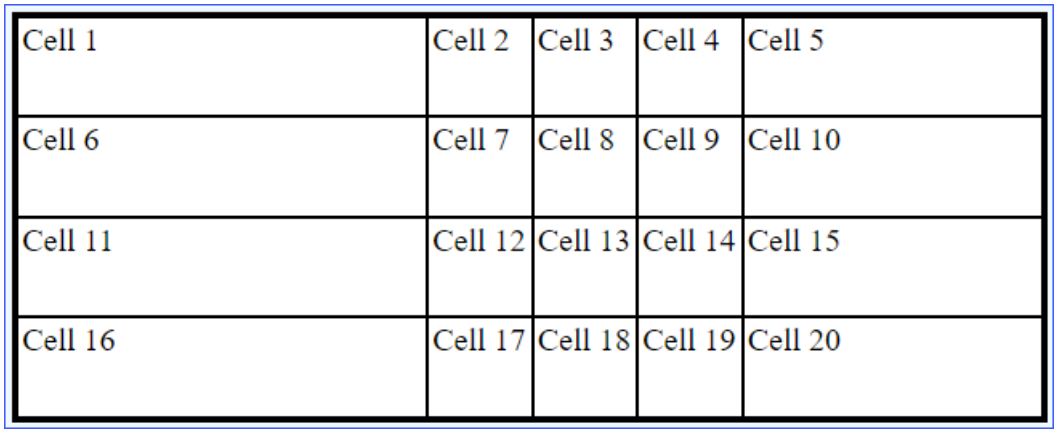
1. A non-negative length values in units (e.g., `200px`).
2. A percentage value relative to the size of the container grid (e.g., `30%`).
3. A non-negative flex value in `fr` units (e.g., `3fr`) - Similar to the unitless values for Flexbox's `flex` property. It specifies the proportion of space that the columns or rows take up.
4. `minmax(min, max)` [45] - Defines the size of the column or row ranging from the `min` to the `max` (e.g., `minmax(100px, 1fr)`).
5. `auto` - The element takes up the most space it can and doesn't shrink smaller than what is specified in `min-width` and `min-height` properties.

The following example sizes the implicit rows from our `grid-template-columns` example:

45. https://developer.mozilla.org/en-US/docs/Web/CSS/minmax

EVALUATION COPY: Not to be used in class.

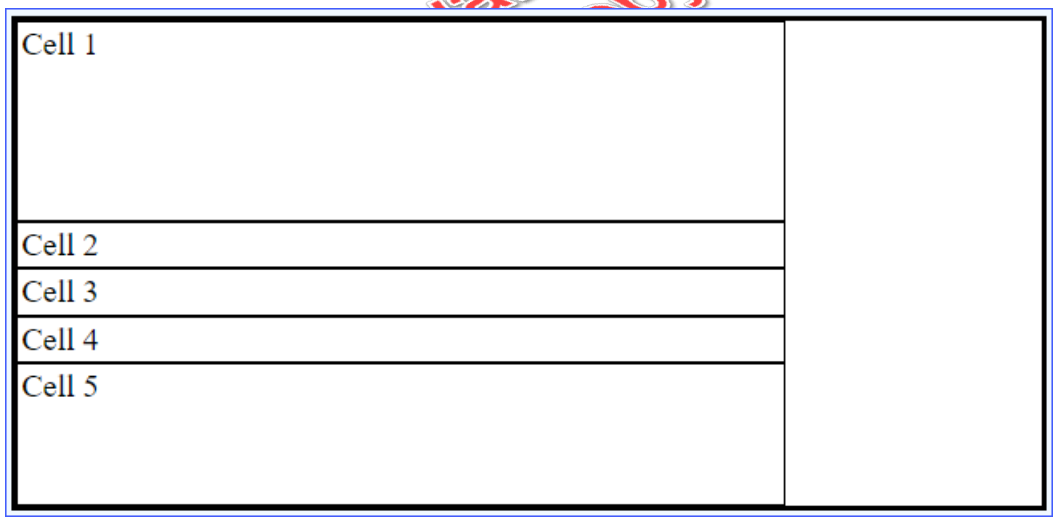
```
grid-template-columns: 40% repeat(3, 1fr) minmax(150px, 1fr);
grid-auto-rows: 50px;
```



Cell 1	Cell 2	Cell 3	Cell 4	Cell 5
Cell 6	Cell 7	Cell 8	Cell 9	Cell 10
Cell 11	Cell 12	Cell 13	Cell 14	Cell 15
Cell 16	Cell 17	Cell 18	Cell 19	Cell 20

The following example sizes the implicit columns from our `grid-template-rows` example:

```
grid-template-rows: 100px repeat(3, 1fr) 3fr;
grid-auto-columns: 75%;
```



Cell 1	
Cell 2	
Cell 3	
Cell 4	
Cell 5	

These examples can be found in `Layouts/Demos/grid-examples.html`.

gap

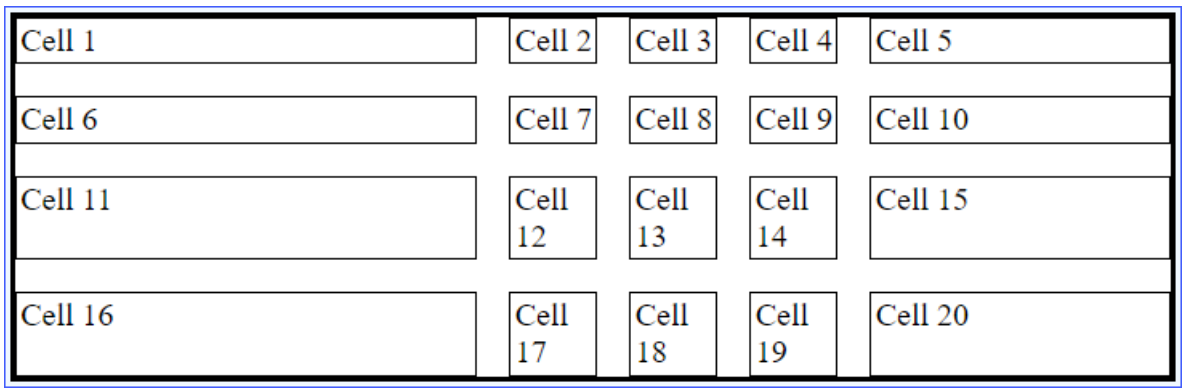
`gap` specifies the space in between rows and columns. It takes one or two length or percentage values.

1. If one value is specified, it applies to the gap between rows and columns (e.g., `gap: 1rem`).
2. If two values are specified, the first applies to the gap between rows and the second to the gap between columns (e.g., `gap: 1rem, 0.5rem`).

Alternatively, the longhand `row-gap` [46] and `column-gap` [47] properties can be used.

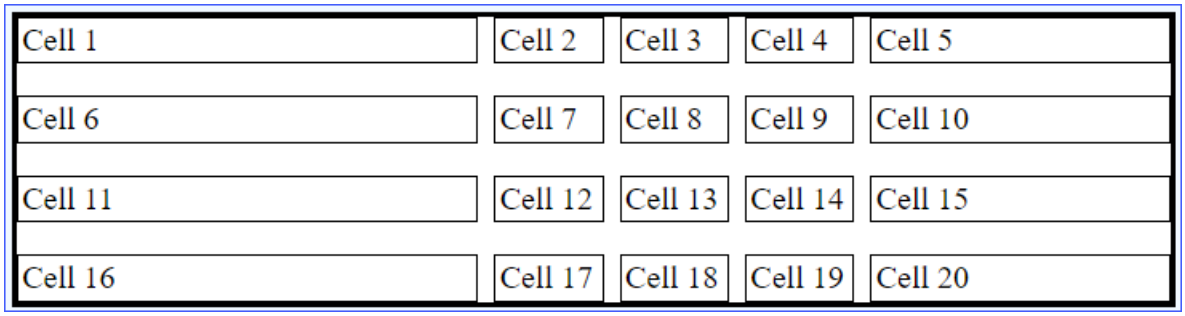
The following example adds a `1rem` gap between rows and columns:

```
gap: 1rem;
grid-template-columns: 40% repeat(3, 1fr) minmax(150px, 1fr);
```



The following example adds a `1rem` gap between rows and a `0.5rem` gap between columns:

```
gap: 1rem 0.5rem;
grid-template-columns: 40% repeat(3, 1fr) minmax(150px, 1fr);
```



46. https://developer.mozilla.org/en-US/docs/Web/CSS/row-gap
47. https://developer.mozilla.org/en-US/docs/Web/CSS/column-gap

EVALUATION COPY: Not to be used in class.

grid-column and grid-row

All the properties we have looked at so far are applied to the container grid element to create the grid. `grid-column` and `grid-row` are applied to the child elements and are used to size and position them within the grid. Both `grid-column` and `grid-row` take one or two values separated by a forward slash (`/`).

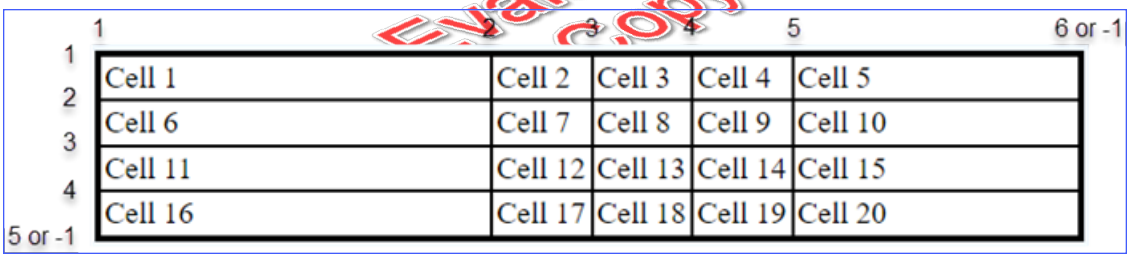
1. The first value (or only value, if only one is specified) sets the starting point of the element.
2. The second value sets the ending point of the element.

These values use grid lines as references. Grid lines are numbered lines between the columns and rows.

1. Column line 1 is on the left-hand side of the first column, column line 2 is on the right-hand side of the first column and on the left-hand side of the second column, etc.
2. Row line 1 is on the top of the first row, row line 2 is on the bottom of the first row and on top of the second row, etc.
3. The last row or column line can be referred to as its positive number using the method above, or it can be referred to as -1.

See the image below to see the grid lines labeled on a basic grid:

Evaluation Copy

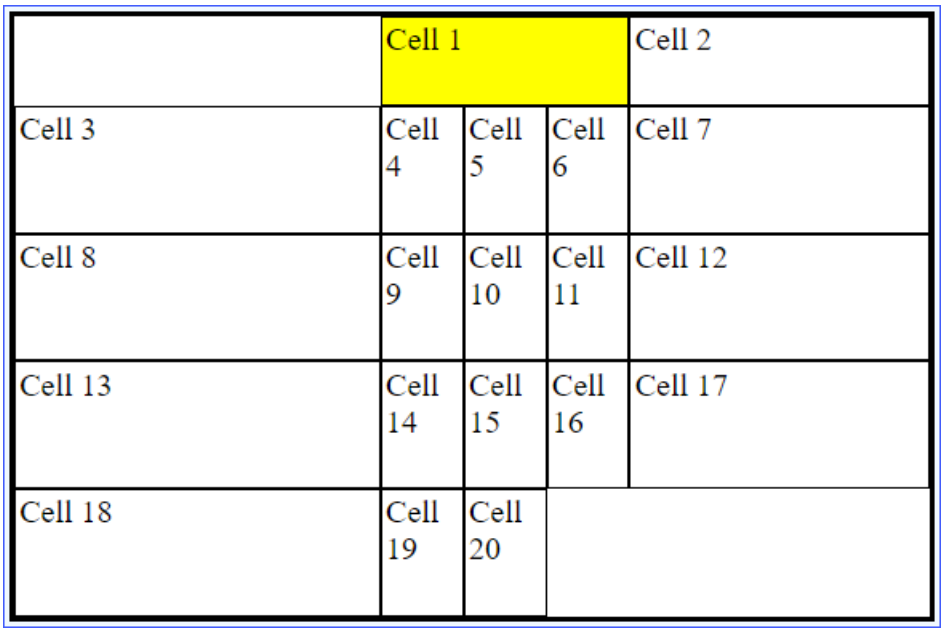


Some common `grid-column` and `grid-row` values are:

1. A unitless integer referring to grid lines (e.g., `grid-column: 1 / 3` sets the element to start at column line 1 and end at column line 3).
2. The `span` keyword and unitless positive number specifying the number of columns or rows to span (e.g., `grid-row: span 2 / -1` sets the element to span two rows and to end at row line -1).

The following example gives Cell 1 a span of three and and makes it end at column line 5:

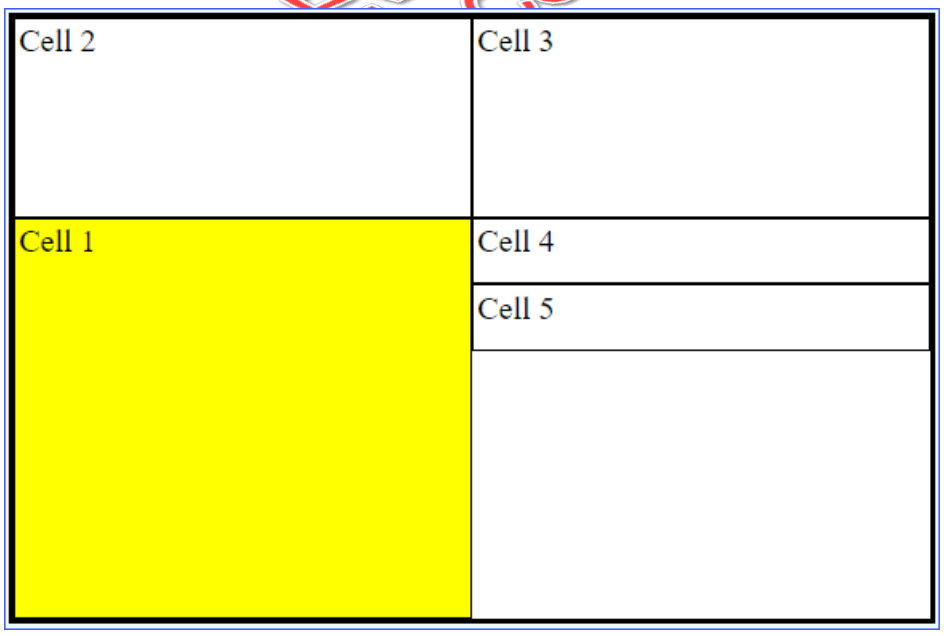
```
grid-column: span 3 / 5;
```



The following example positions Cell 1 starting at row line 2 and ending the last row line (-1):

```
grid-row: 2 / -1;
```

Evaluation Copy



These examples can be found in `Layouts/Demos/grid-examples.html`.

EVALUATION COPY: Not to be used in class.

Below you can see a basic Grid Layout demo:

Demo 12.3: Layouts/Demos/grid-styles.css

```
1.   body {
2.     background: aliceblue;
3.   }
4.
5.   main {
6.     background: white;
7.     display: grid;
8.     gap: 1rem 0.5rem;
9.     grid-template-columns: minmax(150px, 1fr) repeat(3, 1fr);
10.    grid-template-rows: repeat(3, 200px);
11.    margin: auto;
12.    min-width: 600px;
13.    padding: 0.5rem;
14.    width: 80%;
15.  }
16.
17.  main nav {
18.    background: rgb(243, 241, 241);
19.    border: 1px solid black;
20.    border-radius: 0.5rem;
21.    font-size: 1.2rem;
22.    grid-column: 1 / 2;
23.    grid-row: 1 / -1;
24.  }
25.
26.  article {
27.    border: 1px solid black;
28.    overflow: hidden;
29.    padding: 0.5rem;
30.    min-width: 100px;
31.  }
32.
33.  article:target {
34.    border: solid black;
35.    grid-column: 2 / -1;
36.    grid-row: 1;
37.  }
38.
39.  article:target h3 {
40.    text-decoration: underline;
41.  }
```

Evaluation Copy

Demo 12.4: Layouts/Demos/grid.html

```
      -------Lines 1 through 9 Omitted-------
10.   <body class="webucator">
11.   <h1>Basic Grid Layout</h1>
12.   <main>
13.     <nav>
14.       <ol>
15.         <li><a href="#article1">Article 1</a></li>
16.         <li><a href="#article2">Article 2</a></li>
17.         <li><a href="#article3">Article 3</a></li>
18.         <li><a href="#article4">Article 4</a></li>
19.         <li><a href="#article5">Article 5</a></li>
20.         <li><a href="#article6">Article 6</a></li>
21.         <li><a href="#article7">Article 7</a></li>
22.       </ol>
23.     </nav>
24.     <article id="article1">
25.       <h3>Article 1</h3>
26.       <p>Lorem ipsum dolor
27.         sit amet, consectetur adipiscing elit.
28.         Curabitur tristique odio ac sem congue luctus.
29.         Praesent vel rutrum lectus.
      -------Lines 30 through 33 Omitted-------
34.     </article>
35.     <article id="article2">
36.       <h3>Article 2</h3>
37.       <p>Lorem ipsum dolor
38.         sit amet, consectetur adipiscing elit.
39.         Curabitur tristique odio ac sem congue luctus.
40.         Praesent vel rutrum lectus.
      -------Lines 41 through 44 Omitted-------
45.     </article>
46.     <article id="article3">
47.       <h3>Article 3</h3>
48.       <p>Lorem ipsum dolor
49.         sit amet, consectetur adipiscing elit.
50.         Curabitur tristique odio ac sem congue luctus.
51.         Praesent vel rutrum lectus.
      -------Lines 52 through 55 Omitted-------
56.     </article>
57.     <article id="article4">
58.       <h3>Article 4</h3>
59.       <p>Lorem ipsum dolor
60.         sit amet, consectetur adipiscing elit.
```

Evaluation Copy

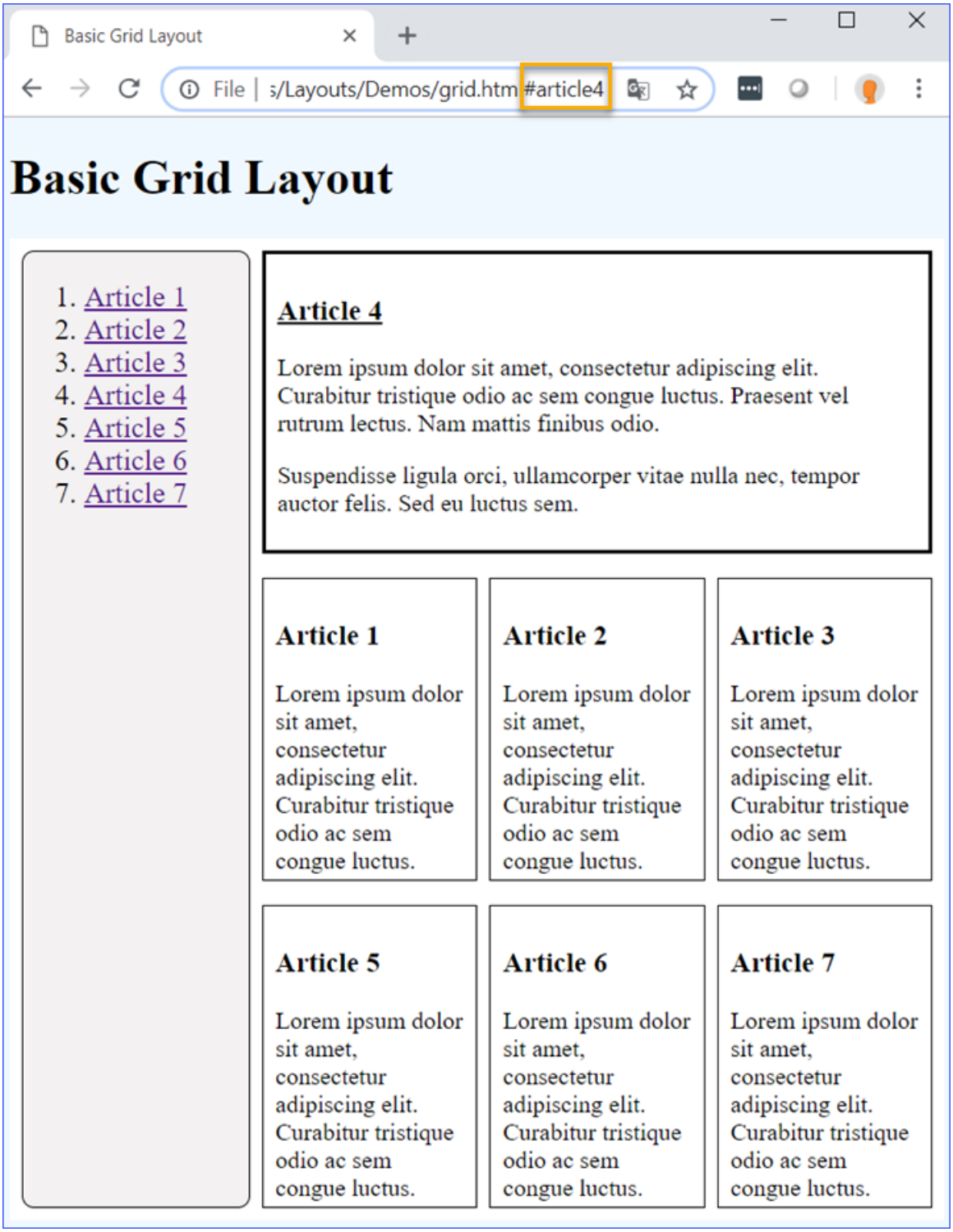
EVALUATION COPY: Not to be used in class.

```
61.         Curabitur tristique odio ac sem congue luctus.
62.         Praesent vel rutrum lectus.
      -------Lines 63 through 66 Omitted-------
67.     </article>
68.     <article id="article5">
69.       <h3>Article 5</h3>
70.       <p>Lorem ipsum dolor
71.         sit amet, consectetur adipiscing elit.
72.         Curabitur tristique odio ac sem congue luctus.
73.         Praesent vel rutrum lectus.
      -------Lines 74 through 77 Omitted-------
78.     </article>
79.     <article id="article6">
80.       <h3>Article 6</h3>
81.       <p>Lorem ipsum dolor
82.         sit amet, consectetur adipiscing elit.
83.         Curabitur tristique odio ac sem congue luctus.
84.         Praesent vel rutrum lectus.
      -------Lines 85 through 88 Omitted-------
89.     </article>
90.     <article id="article7">
91.       <h3>Article 7</h3>
92.       <p>Lorem ipsum dolor
93.         sit amet, consectetur adipiscing elit.
94.         Curabitur tristique odio ac sem congue luctus.
95.         Praesent vel rutrum lectus.
      -------Lines 96 through 99 Omitted-------
100.    </article>
101.  </main>
102.  </body>
103.  </html>
```

Evaluation Copy

Code Explanation

The code above will render the following:



Some things to note:

EVALUATION COPY: Not to be used in class.

1. The `main` element is the grid container.
2. The target `article` is positioned at the top. `article4` is the target in this image.

12.3. Multi-column Layout

Multi-column offers a way to organize content into columns. We can control the size and number of columns, the gap between columns, and an optional rule between columns. Listed below are the relevant properties.

Multi-column Properties

Property	Explanation	Example/Possible Values
`columns`	Sets the width and count (number) of columns.	`columns: 10rem 2`
`column-fill`	Sets how columns are populated with content.	`auto` (fill columns sequentially) or `balance` (balance content as equally as possible).
`column-gap`	Sets the size of the gap between columns.	A length value in units or `normal`, which is `1em` for Multi-column.
`column-rule`	Sets the width, style, and color of the rule (a line that separates columns).	`column: 0.2rem dashed red`
`break-inside`	Sets whether or not an element can break inside itself.	`auto` (can break) or `avoid` (avoids breaking).

Evaluation Copy

Unlike Flexbox and Grid, Multi-column is not meant to be used to create the entire layout of your web page. It is best to use it for collapsing small text elements like lists, or for creating masonry layouts. We do both in the following demo:

Demo 12.5: Layouts/Demos/multi-column-styles.css

```
1.  body {
2.    background: aliceblue;
3.  }
4.
5.  main {
6.    background: white;
7.    margin: auto;
8.    min-width: 600px;
9.    width: 90%;
10. }
11.
12. h2 {
13.   margin: 0.2rem auto;
14.   padding: 0.2rem;
15.   width: 256px;
16. }
17.
18. nav {
19.   border: solid black;
20.   border-radius: 0.5rem;
21.   margin: auto;
22.   min-width: 550px;
23.   width: 90%;
24. }
25.
26. nav ol {
27.   columns: 4;
28.   column-gap: 2rem;
29.   margin: 0.2rem auto;
30.   width: 90%;
31. }
32.
33. div {
34.   background: rgb(235, 233, 233);
35.   box-sizing: border-box;
36.   break-inside: avoid;
37.   margin: 0 auto 1rem;
38.   padding: 0;
39. }
40.
41. div a {
42.   display: block;
43.   width: 100%;
44. }
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
45.
46.   div img {
47.     width: 100%;
48.     transition: transform 0.5s;
49.   }
50.
51.   div img:hover {
52.     transform: scale(0.95);
53.     transition: transform 0.5s;
54.   }
55.
56.   #gallery {
57.     columns: 200px;
58.     column-fill: balance;
59.     column-gap: 0.5rem;
60.     margin: 1rem auto;
61.     width: 90%;
62.   }
63.
64.   p.caption {
65.     font-size: 1.2rem;
66.     font-style: italic;
67.     margin: 0 1em;
68.     padding: 0 1em;
69.   }
```

Evaluation Copy

Demo 12.6: Layouts/Demos/multi-column.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.  <h1>Multi-column</h1>
12.  <main>
13.    <h2>Check out our cute dogs!</h2>
14.    <nav>
15.      <ol>
16.        <li><a href="#dog1">Cute Dog 1</a></li>
17.        <li><a href="#dog2">Cute Dog 2</a></li>
18.        <li><a href="#dog3">Cute Dog 3</a></li>
19.        <li><a href="#dog4">Cute Dog 4</a></li>
     -------Lines 20 through 34 Omitted-------
35.        <li><a href="#dog20">Cute Dog 20</a></li>
36.      </ol>
37.    </nav>
38.    <section id="gallery">
39.      <div id="dog1">
40.        <a href="#dog1">
41.          <img src="images/dog1.jpg" alt="Cute Dog 1"></a>
42.        <p class="caption">
43.          Here is one cute dog!</p>
44.      </div>
45.      <div id="dog2">
46.        <a href="#dog2">
47.          <img src="images/dog2.jpg" alt="Cute Dog 2"></a>
48.        <p class="caption">
49.          And here's another one!</p>
50.      </div>
51.      <div id="dog3">
52.        <a href="#dog3">
53.          <img src="images/dog3.jpg" alt="Cute Dog 3"></a>
54.        <p class="caption">
55.          So cute!</p>
56.      </div>
57.      <div id="dog4">
     -------Lines 58 through 153 Omitted-------
154.     <div id="dog20">
155.       <a href="#dog20">
156.         <img src="images/dog20.jpg" alt="Cute Dog 20"></a>
157.       <p class="caption">
158.         Mommy and baby. Adorable!</p>
159.     </div>
160.   </section>
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
161.  </main>
162.  </body>
163.  </html>
```

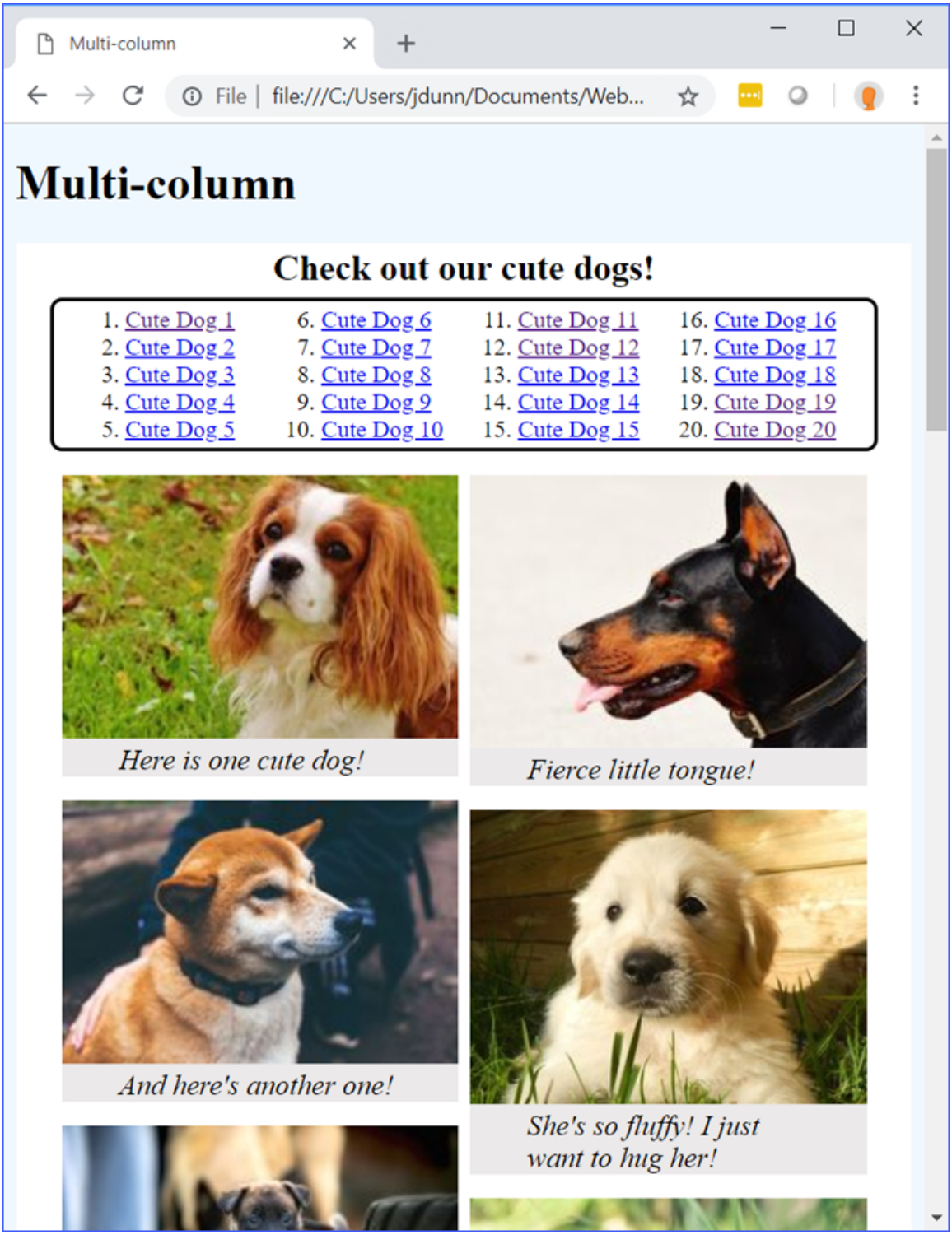
Evaluation Copy

Code Explanation

The above code will render the following: [48]

48. The images used in this section are in the public domain (`https://commons.wikimedia.org/wiki/Public_domain`):

- `https://pixnio.com/fauna-animals/dogs/animal-cute-dog-canine-pet`
- `https://pixnio.com/fauna-animals/dogs/animal-cute-dog-pet-canine`
- `https://pixnio.com/fauna-animals/dogs/animal-dog-breed-canine-cute-field-fur-grass`
- `https://pixnio.com/fauna-animals/dogs/animal-dog-doberman-breed-necklace`
- `https://pixnio.com/fauna-animals/dogs/animal-dog-leash-brown-canine-portrait-colorful-dog-cute-puppy`
- `https://pixnio.com/fauna-animals/dogs/animal-pet-dog-puppy-canine`
- `https://pixnio.com/fauna-animals/dogs/cute-dog-canine-green-grass-young`
- `https://pixnio.com/fauna-animals/dogs/dog-grass-pet-looking-fur-wood-shadow`
- `https://pixnio.com/fauna-animals/dogs/dog-grass-pet-puppy-animal`
- `https://pixnio.com/fauna-animals/dogs/dogs-grass-pets-wood-shade-backyard`
- `https://pixnio.com/fauna-animals/dogs/dogs-pet-animals-wood-grass-nature`
- `https://pixnio.com/fauna-animals/dogs/husky-sled-dog-canine-fur-pet-dog-cute-portrait`
- `https://pixnio.com/fauna-animals/dogs/pet-cute-dog-puppy-sleep-canine-animal-bed-bedroom`
- `https://pixnio.com/fauna-animals/dogs/pug-puppy-dog-canine-purebred`
- `https://pixnio.com/fauna-animals/dogs/purebred-canine-pet-cute-dog-fur-grass-portrait`
- `https://pixnio.com/fauna-animals/dogs/vertebrate-canine-dog-grass-pet`
- `https://pixnio.com/fauna-animals/dogs/puppy-purebred-dog-canine`
- `https://pixnio.com/fauna-animals/dogs/dog-field-grass-pet-animal-jump`
- `https://pixnio.com/fauna-animals/dogs/grey-purebred-dog-pet-canine-terrier-shepherd-dog-cute`
- `https://pixnio.com/fauna-animals/dogs/hunting-dog-dog-animal-cute-canine-puppy-portrait-pedigree`



Some things to note:

EVALUATION COPY: Not to be used in class.

1. The top menu list will always have four columns, and this way it conserves vertical space.
2. The `gallery section` will have as many `200px` columns as will fit. Making the window bigger creates room for more columns.

Evaluation Copy

Exercise 14: Layouts

25 to 40 minutes

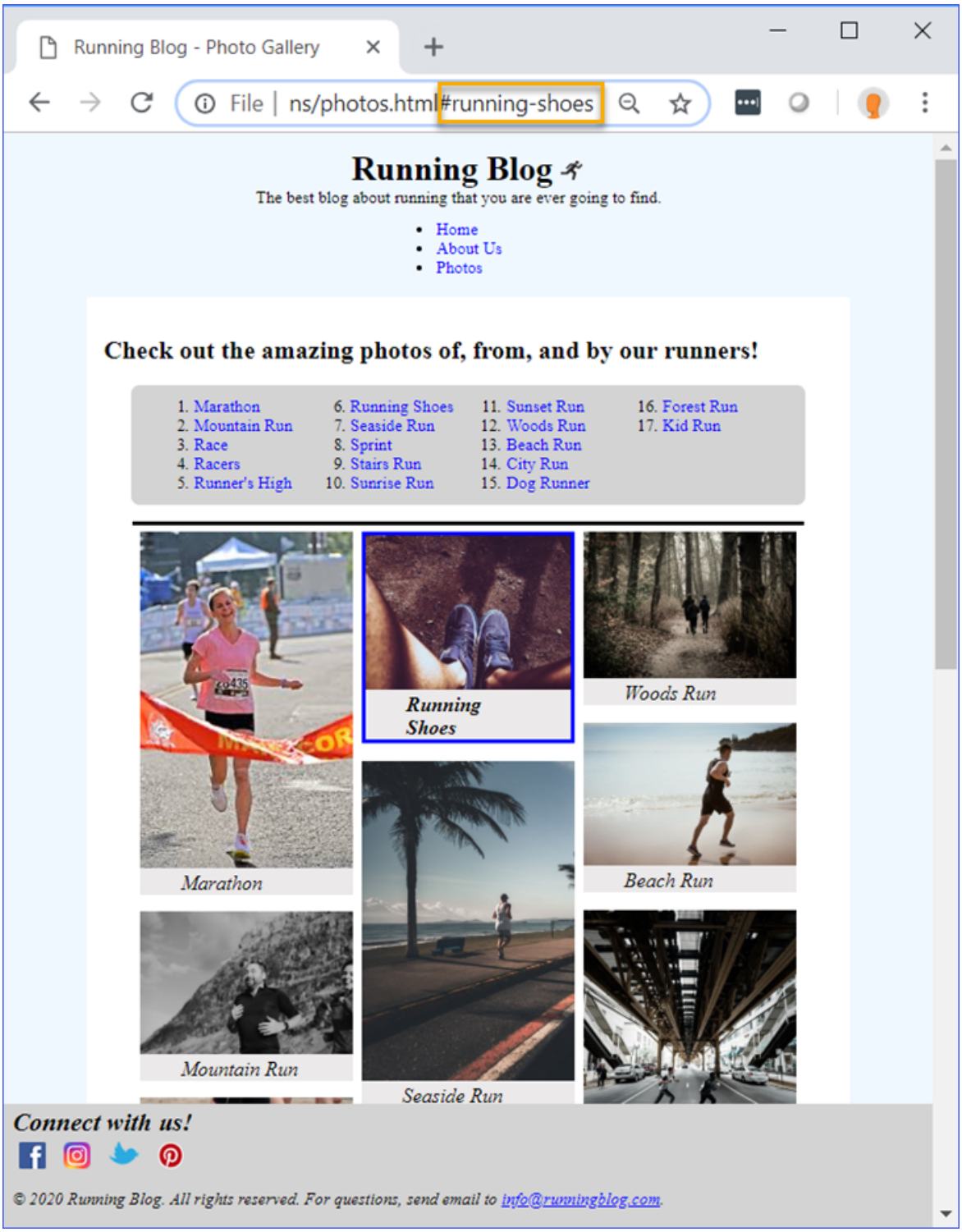
In this exercise, you will continue to work with a new page that is part of the Running Blog.

1. Open `Layouts/Exercises/photos.html` in your editor.
2. Open `Layouts/Exercises/blog-styles.css`. This is the stylesheet from the solution to the Positioning exercise. Feel free to continue working from your own stylesheet.
3. Using your new knowledge of layouts add some styles to this page. The object of this exercise is to practice using the properties covered in this lesson, but note that for this page it may not make sense to use all the layouts. You may end up just using one or two. This is perfectly fine. Also, feel free to add or change any other properties that you know.
4. When you are done, open `photos.html` in your browser to see the results. You are welcome to go back to the code and continue to work.

Evaluation Copy

You can design it however you like, or you can try to make it look something like:

EVALUATION COPY: Not to be used in class.


Running Blog - Photo Gallery
File | ns/photos.html#running-shoes
Running Blog
The best blog about running that you are ever going to find.
Home
About Us
Photos
Check out the amazing photos of, from, and by our runners!
1. Marathon
2. Mountain Run
3. Race
4. Racers
5. Runner's High
6. Running Shoes
7. Seaside Run
8. Sprint
9. Stairs Run
10. Sunrise Run
11. Sunset Run
12. Woods Run
13. Beach Run
14. City Run
15. Dog Runner
16. Forest Run
17. Kid Run
Marathon
Running Shoes
Woods Run
Beach Run
Mountain Run
Seaside Run
Connect with us!
© 2020 Running Blog. All rights reserved. For questions, send email to info@runningblog.com.

Solution: Layouts/Solutions/blog-styles.css

```
     -------Lines 1 through 194 Omitted-------
195.
196. main#photos {
197.   min-width: 620px;
198. }
199.
200. #photos nav {
201.   margin: auto;
202.   min-width: 550px;
203.   width: 90%;
204. }
205.
206. #photos nav ol {
207.   columns: 4;
208.   column-gap: 2rem;
209.   margin: 0.2rem auto;
210.   width: 90%;
211. }
212.
213. #gallery {
214.   columns: 200px;
215.   column-fill: balance;
216.   column-gap: 0.5rem;
217.   margin: 1rem auto;
218.   width: 90%;
219. }
220.
221. #gallery div {
222.   background: rgb(235, 233, 233);
223.   box-sizing: border-box;
224.   break-inside: avoid;
225.   margin: 0 auto 1rem;
226.   padding: 0;
227. }
228.
229. #gallery div a {
230.   display: block;
231.   width: 100%;
232. }
233.
234. #gallery div img {
235.   width: 100%;
236.   transition: transform 0.5s;
237. }
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
238.
239. #gallery div img:hover {
240.   transform: scale(0.95);
241.   transition: transform 0.5s;
242. }
243.
244. p.caption {
245.   font-size: 1.2rem;
246.   font-style: italic;
247.   margin: 0 1em;
248.   padding: 0 1em;
249. }
250.
251. #gallery div:target {
252.   border: blue solid;
253. }
254.
255. #gallery div:target p.caption {
256.   font-weight: bold;
257. }
```

Evaluation Copy

Code Explanation

Some things to note:

1. We omitted lines in Layouts/Solutions/blog-styles.css for brevity.
2. Note that the `running-shoes div` is the target in the screenshot.
3. We only used the Multi-column layout, but that does not mean you have to or that you can't also use the other layouts.

Conclusion

In this lesson, you have been introduced to the Flexbox, Grid, and Multi-column layouts.

EVALUATION COPY: Not to be used in class.

LESSON 13

CSS Lists as Hierarchical Navigation

Topics Covered

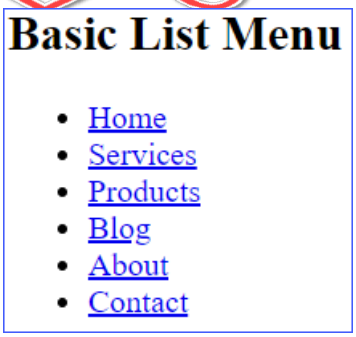
☑ Horizontal and vertical navigation bars.

☑ Nested lists for multi-level navigation.

Introduction

As you have seen throughout this course in our demos and exercises, lists of links are often used as navigation bars like the one shown below:

Evaluation Copy

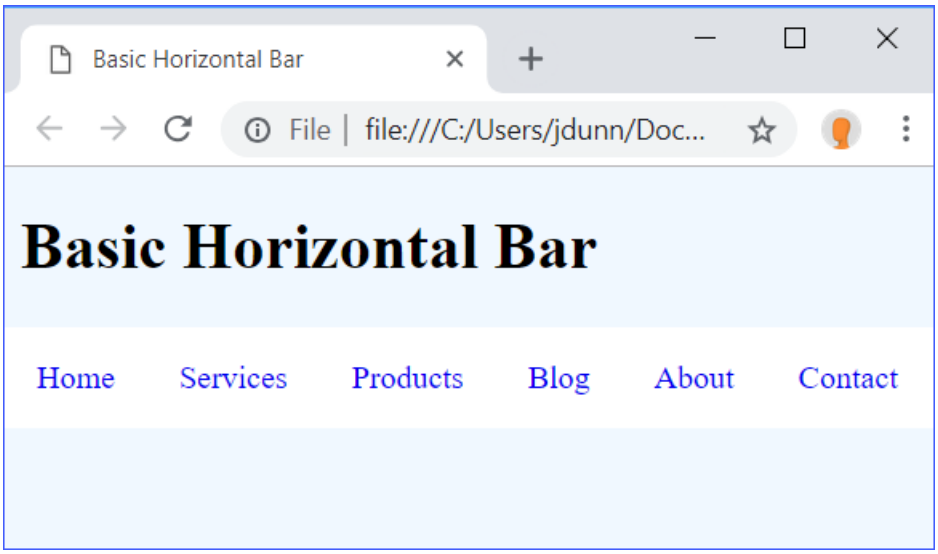


In this lesson, through a series of exercises, you will learn how to style these lists to turn them into easy-to-use and dynamic navigation bars.

Exercise 15: Basic Horizontal Navigation Bars

10 to 20 minutes

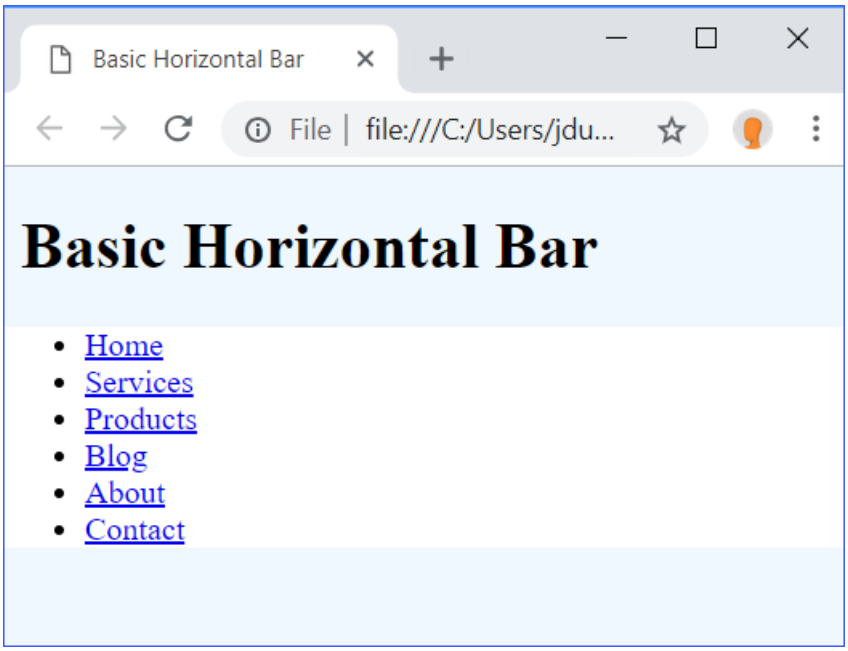
Let's look at how we could turn the basic list menu from the introduction into a stylish horizontal navigation bar, like this:



Follow these steps:

1. Open `CssListMenus/Exercises/basic-horizontal-bar-styles.css` and `CssListMenus/Exercises/basic-horizontal-bar.html` in your editor.
2. Style the `body` and `nav`:
    A. Add an `aliceblue` background to the `body`.
    B. Add a `white` background to the `nav` and left and right `margin` of `-0.5rem` (to cancel out the default `margin` added to the `body` element and make the `nav` touch both side of the viewport.

EVALUATION COPY: Not to be used in class.

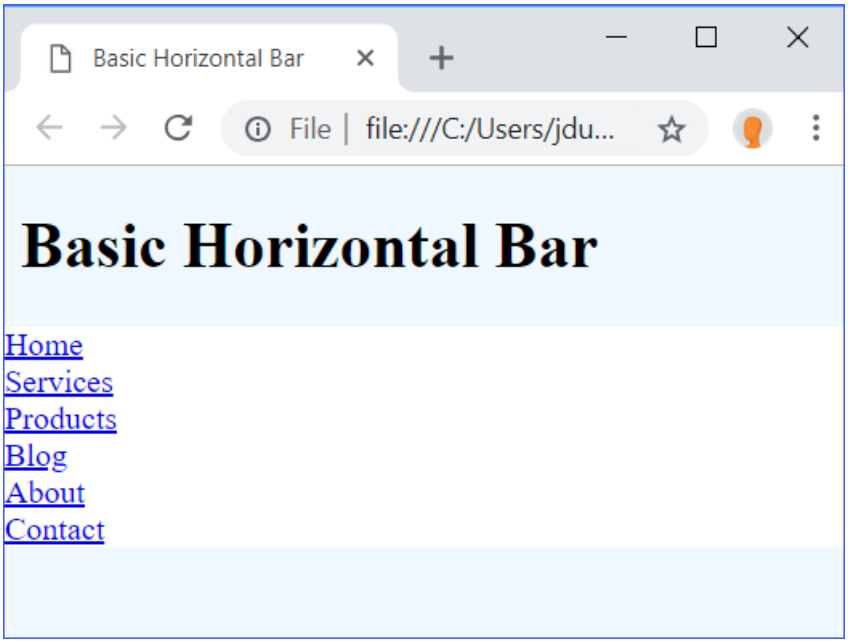


3. Get rid of default `ul` styles:

    A. Remove bullets by setting `list-style`[49] to `none`.

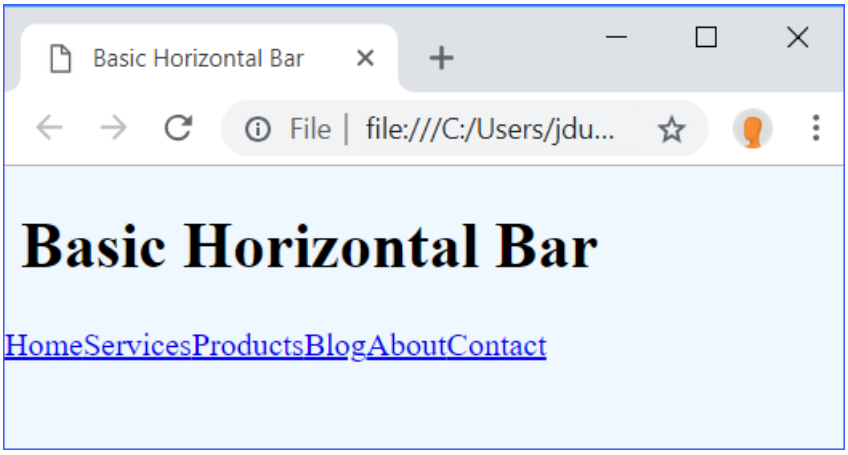
    B. Set `margin` and `padding` to `0`.

Evaluation Copy



49. https://developer.mozilla.org/en-US/docs/Web/CSS/list-style

4. Float the `li` elements left to line them up horizontally.



5. Style the links:

   A. Set `display` to `block`.

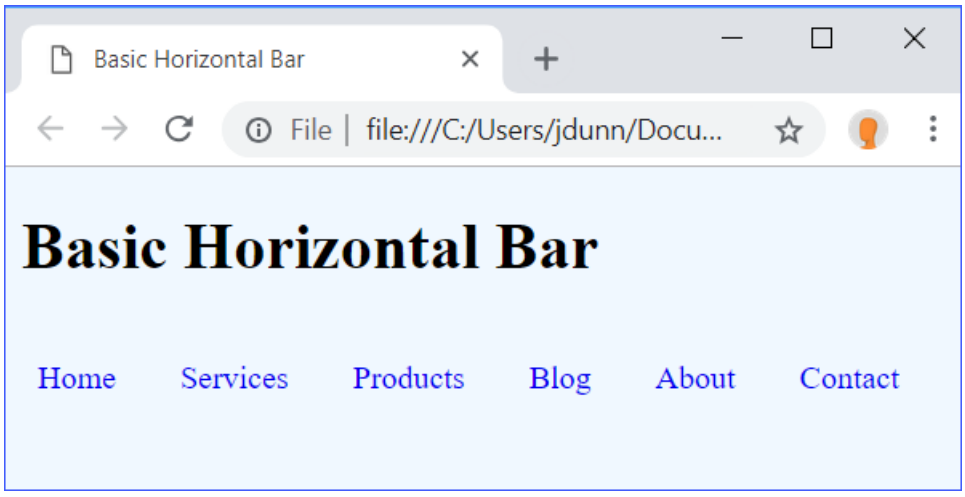
   B. Set `padding` to `1rem`.

   C. Set `text-decoration` to `none`.

   D. Change the `background` when the links are hovered over.

   E. Change the `color` when the links are clicked.

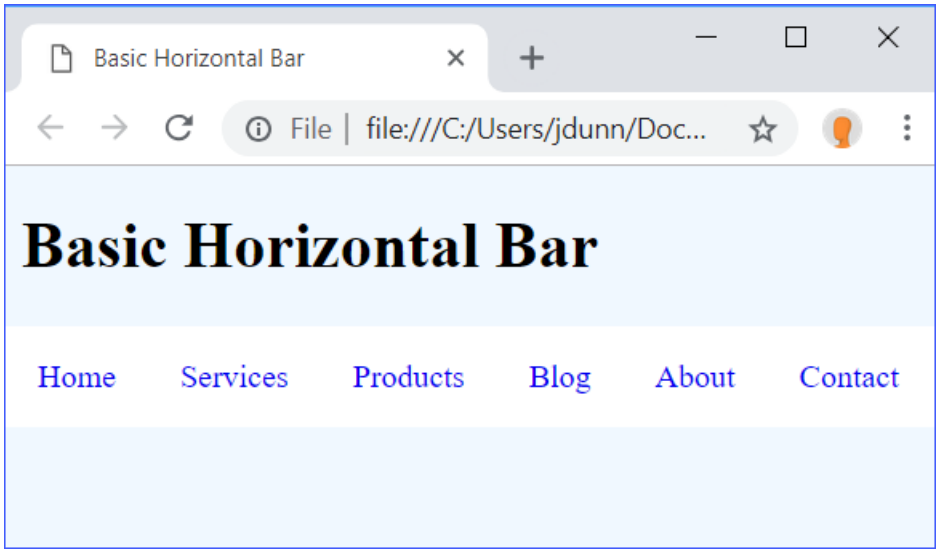
Evaluation Copy



EVALUATION COPY: Not to be used in class.

6. Notice that the white background seems to have disappeared. This is because we took the `li` elements out of normal flow. To get it back, use the `::after` pseudo-element to add a cleared block pseudo-element at the end of the `nav` element:

```
nav::after {
  content: "";
  display: block;
  clear: both;
}
```



You can see the final code below:

Solution: CssListMenus/Solutions/basic-horizontal-bar-styles.css

```
1.  body {
2.    background: aliceblue;
3.  }
4.
5.  nav {
6.    background: white;
7.    margin: 0 -0.5rem;
8.  }
9.
10. nav::after {
11.   content: "";
12.   display: block;
13.   clear: both;
14. }
15.
16. nav ul {
17.   list-style: none;
18.   margin: 0;
19.   padding: 0;
20. }
21.
22. nav li {
23.   float: left;
24. }
25.
26. nav a {
27.   display: block;
28.   padding: 1rem;
29.   text-decoration: none;
30. }
31.
32. nav a:hover {
33.   background: rgb(240, 240, 240);
34. }
35.
36. nav a:active {
37.   color: #cce;
38. }
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

Solution: CssListMenus/Solutions/basic-horizontal-bar.html

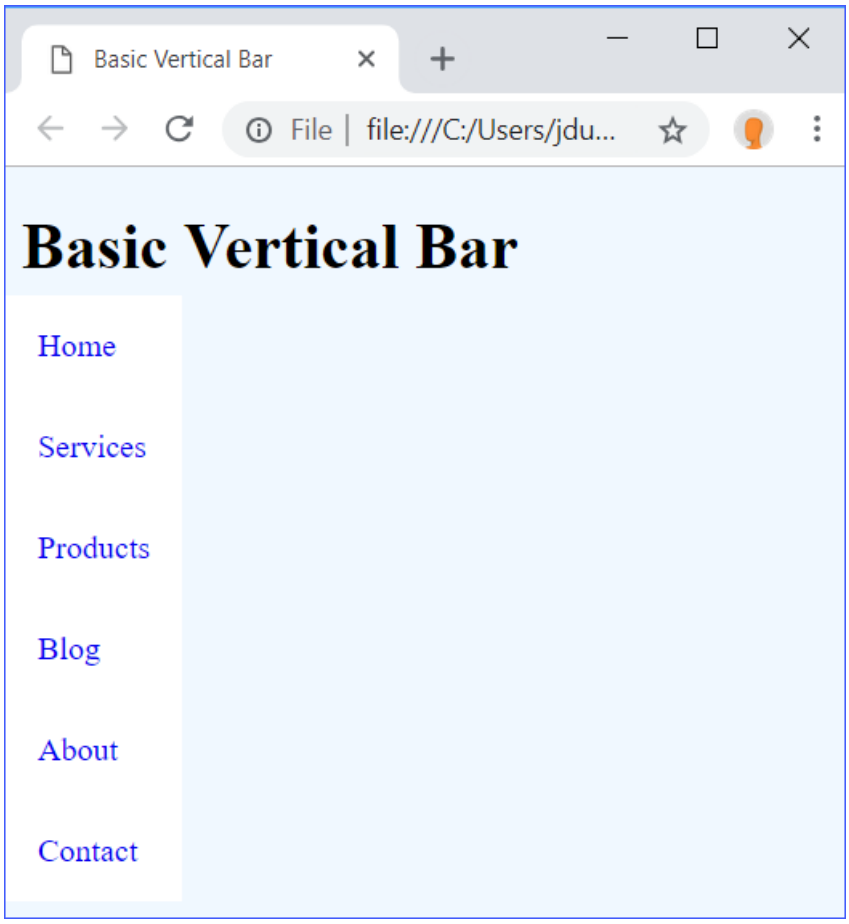
```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.  <h1>Basic Horizontal Bar</h1>
12.  <nav>
13.    <ul>
14.      <li><a href="index.html">Home</a></li>
15.      <li><a href="services.html">Services</a></li>
16.      <li><a href="products.html">Products</a></li>
17.      <li><a href="blog.html">Blog</a></li>
18.      <li><a href="about.html">About</a></li>
19.      <li><a href="contact.html">Contact</a></li>
20.    </ul>
21.  </nav>
22.  </body>
23.  </html>
```

Evaluation Copy

Exercise 16: Basic Vertical Navigation Bars

10 to 20 minutes

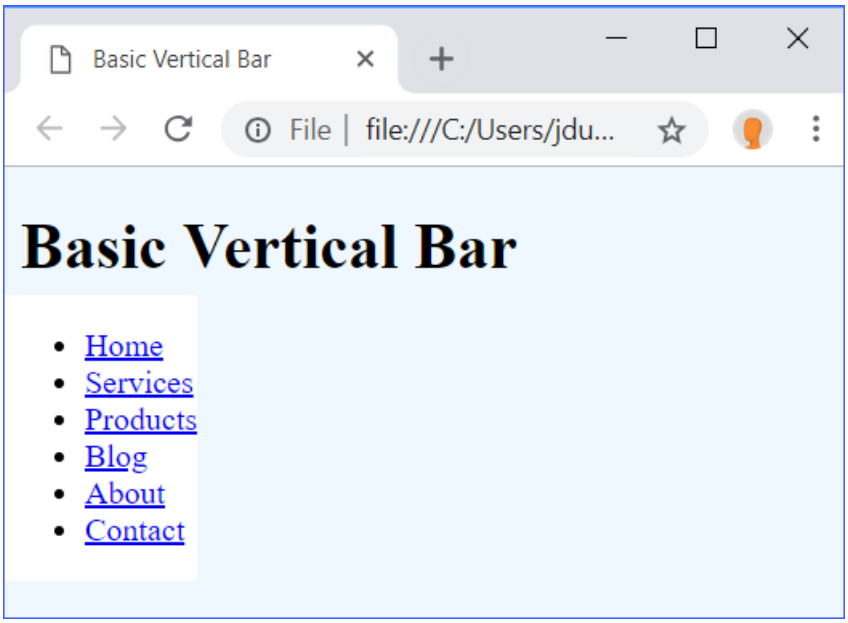
Let's look at how we could turn the basic list menu from the introduction into a stylish vertical navigation bar, like this:



We will follow these steps:

1. Open `CssListMenus/Exercises/basic-vertical-bar-styles.css` and `CssListMenus/Exercises/basic-vertical-bar.html` in your editor.
2. Style the `body` and `nav`:
    A. Add an `aliceblue` background to the `body`.
    B. Add a `white` background to the `nav` and position it absolutely `4rem` from the top and give it a left `margin` of `-0.5rem`.

EVALUATION COPY: Not to be used in class.

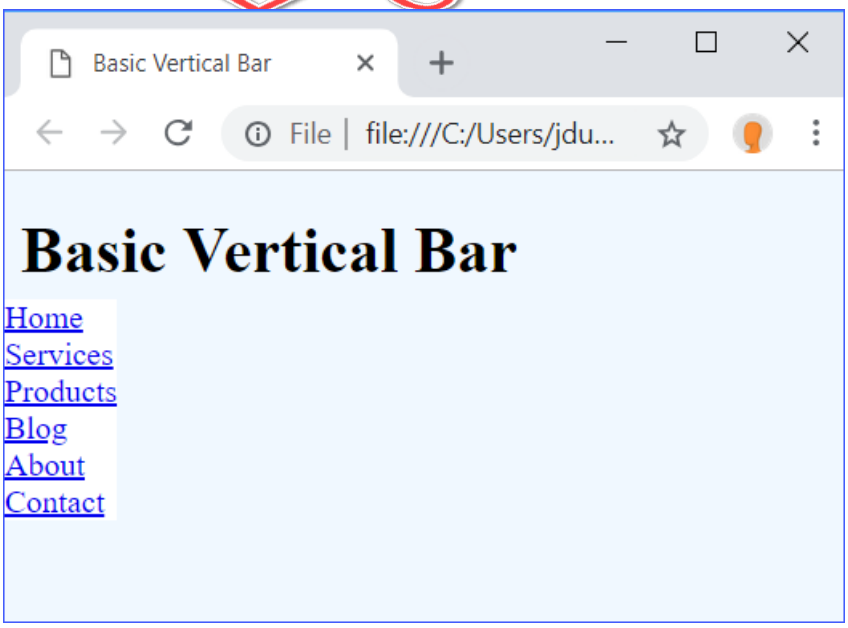


3. Get rid of default `ul` styles:

    A. Remove bullets by setting `list-style`[50] to `none`.

    B. Set `margin` and `padding` to `0`.

Evaluation Copy



4. Style the links:

50. https://developer.mozilla.org/en-US/docs/Web/CSS/list-style

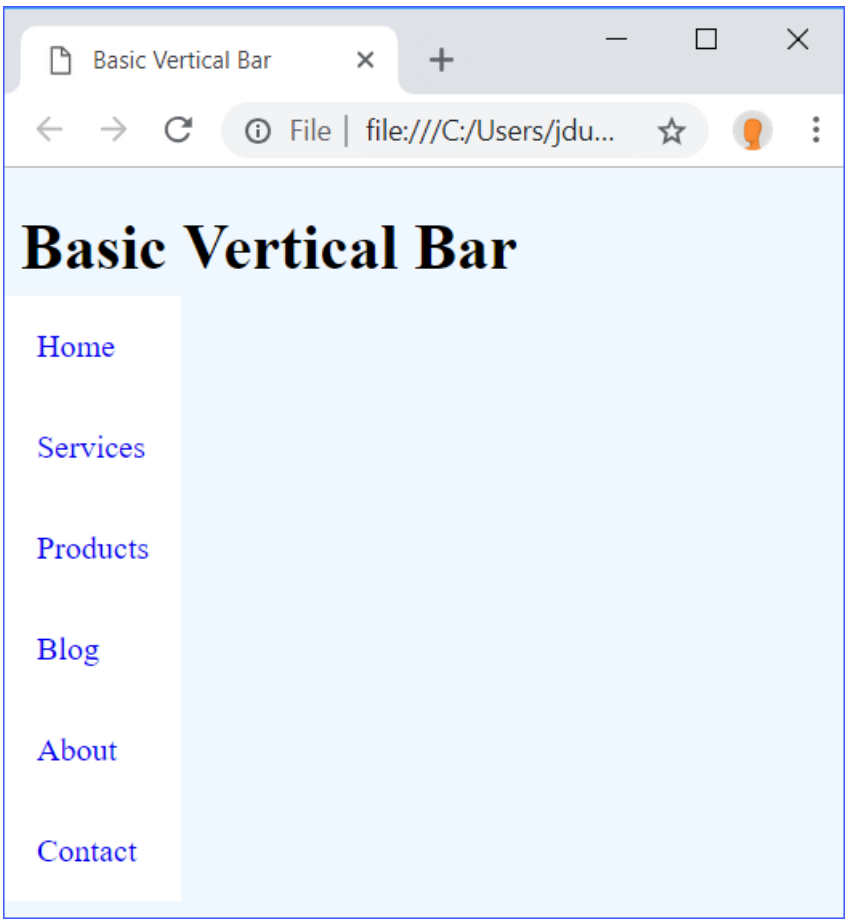
A. Set `display` to `block`.

B. Set `padding` to `1rem`.

C. Set `text-decoration` to `none`.

D. Change the `background` when the links are hovered over.

E. Change the `color` when the links are clicked.



You can see the final code below:

EVALUATION COPY: Not to be used in class.

Solution: CssListMenus/Solutions/basic-vertical-bar-styles.css

```
1.   body {
2.     background: aliceblue;
3.   }
4.
5.   nav {
6.     background: white;
7.     position: absolute;
8.     top: 4rem;
9.     margin: 0 -0.5rem;
10.  }
11.
12.  nav ul {
13.    list-style: none;
14.    margin: 0;
15.    padding: 0;
16.  }
17.
18.  nav a {
19.    display: block;
20.    padding: 1rem;
21.    text-decoration: none;
22.  }
23.
24.  nav a:hover {
25.    background: rgb(240, 240, 240);
26.  }
27.
28.  nav a:active {
29.    color: #cce;
30.  }
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

Solution: CssListMenus/Solutions/basic-vertical-bar.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.  <h1>Basic Vertical Bar</h1>
12.  <nav>
13.    <ul>
14.      <li><a href="index.html">Home</a></li>
15.      <li><a href="services.html">Services</a></li>
16.      <li><a href="products.html">Products</a></li>
17.      <li><a href="blog.html">Blog</a></li>
18.      <li><a href="about.html">About</a></li>
19.      <li><a href="contact.html">Contact</a></li>
20.    </ul>
21.  </nav>
22.  </body>
23.  </html>
```

13.1. Dynamic Drop-down and Fly-out Navigation Bars

Evaluation Copy

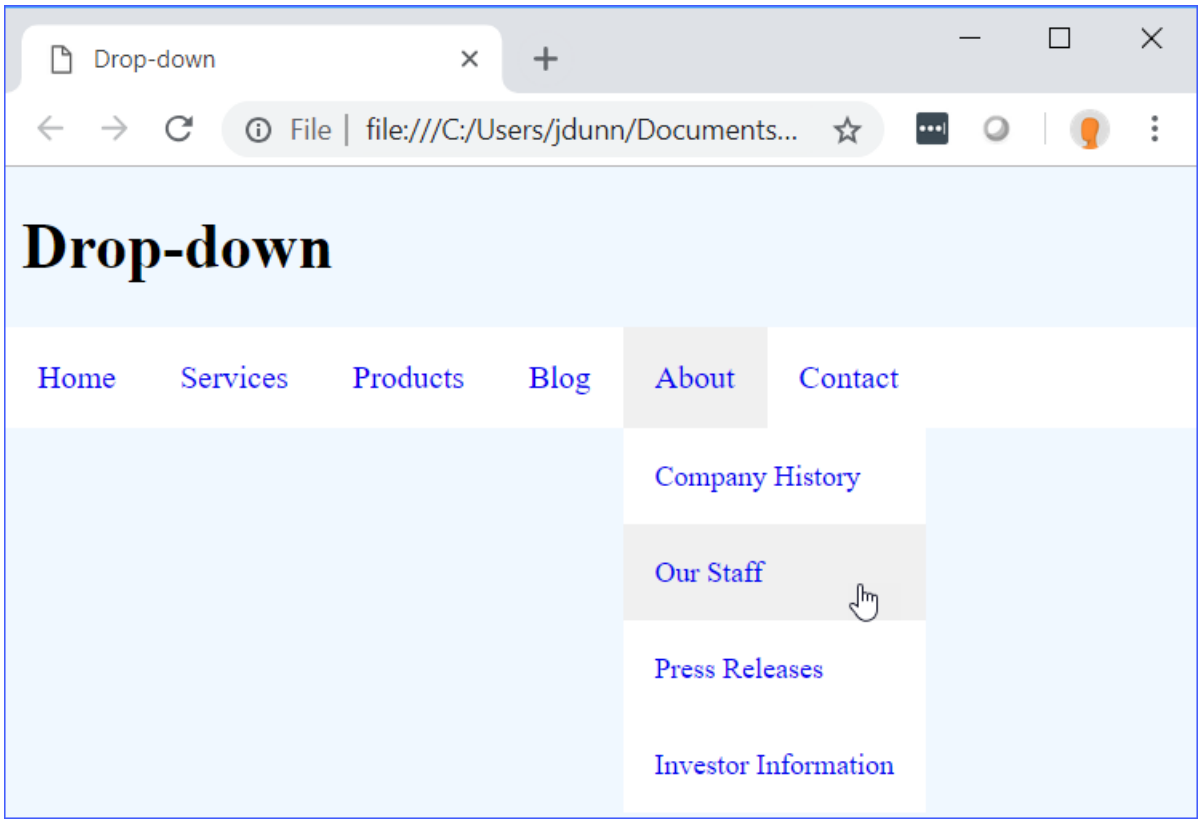
It is often the case that a navigation menu will have nested lists. For example, our basic list menu might have additional options underneath the "About" link:

We probably don't want the nested list to be shown all the time, so we can hide it and make it show up when the user hovers over the "About" link. For a horizontal bar, we will create a drop-down menu and for a vertical bar, we will create a fly-out menu.

Exercise 17: Basic Drop-down Menu

10 to 20 minutes

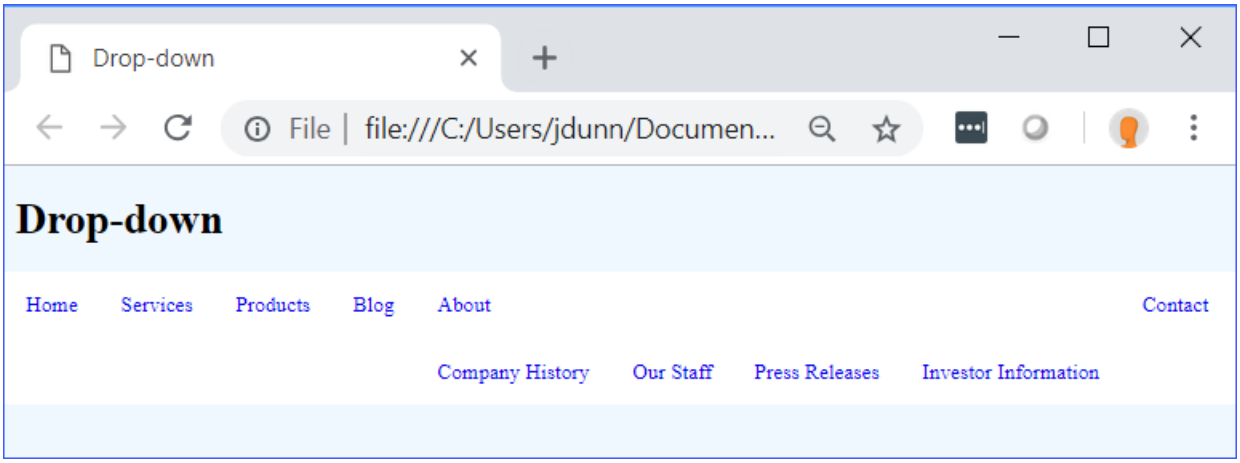
Start with the horizontal navigation bar that we already created, and add a drop-down menu, so that it looks like this:



Follow these steps:

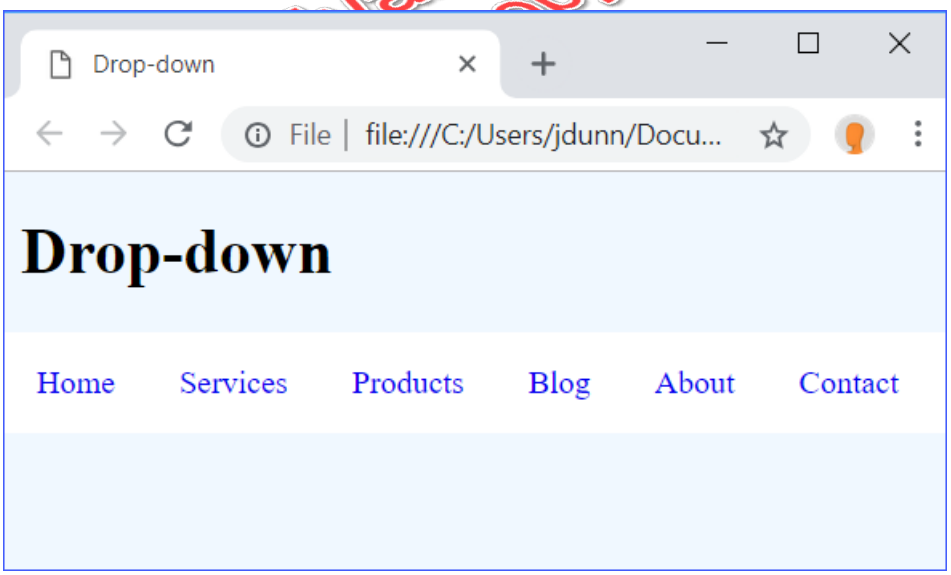
1. Open `CssListMenus/Exercises/drop-down-styles.css` and `CssListMenus/Exercises/drop-down.html` in your editor.
2. First, change the HTML to add the nested list to the "About" `li` and add a `contains` class to the "About" `li`.

EVALUATION COPY: Not to be used in class.

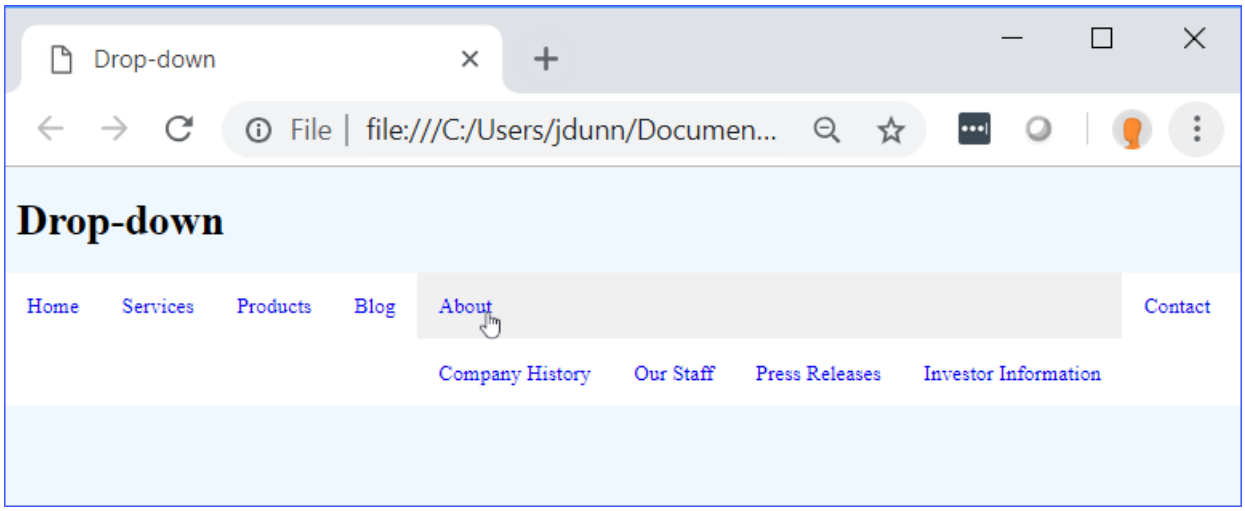


3. Copy and paste either your code from `CssListMenus/Exercises/basic-horizontal-bar-styles` or the solution code from `CssListMenus/Solutions/basic-horizontal-bar-styles` into `CssListMenus/Exercises/drop-down-styles.css`.

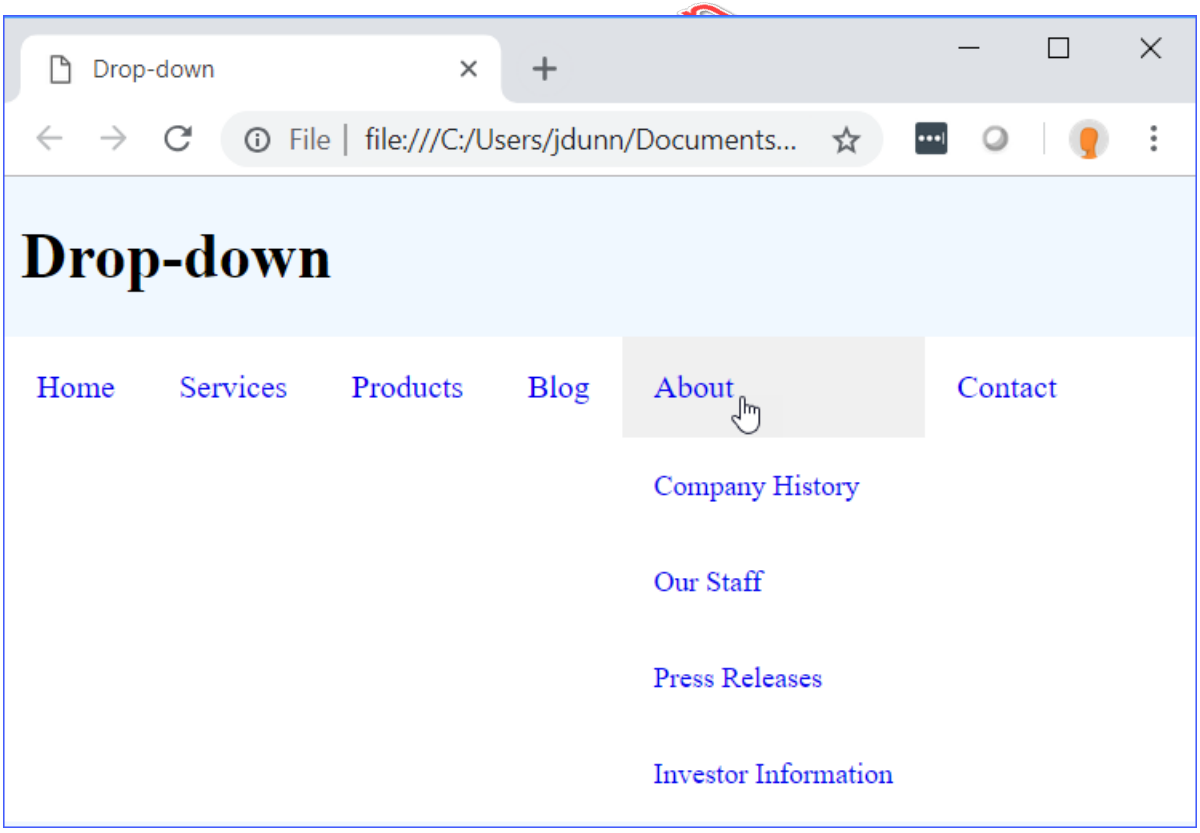
4. Hide the nested `ul` by setting its `display` to `none`.



5. Display the `ul` when the "About" `li` is hovered over.

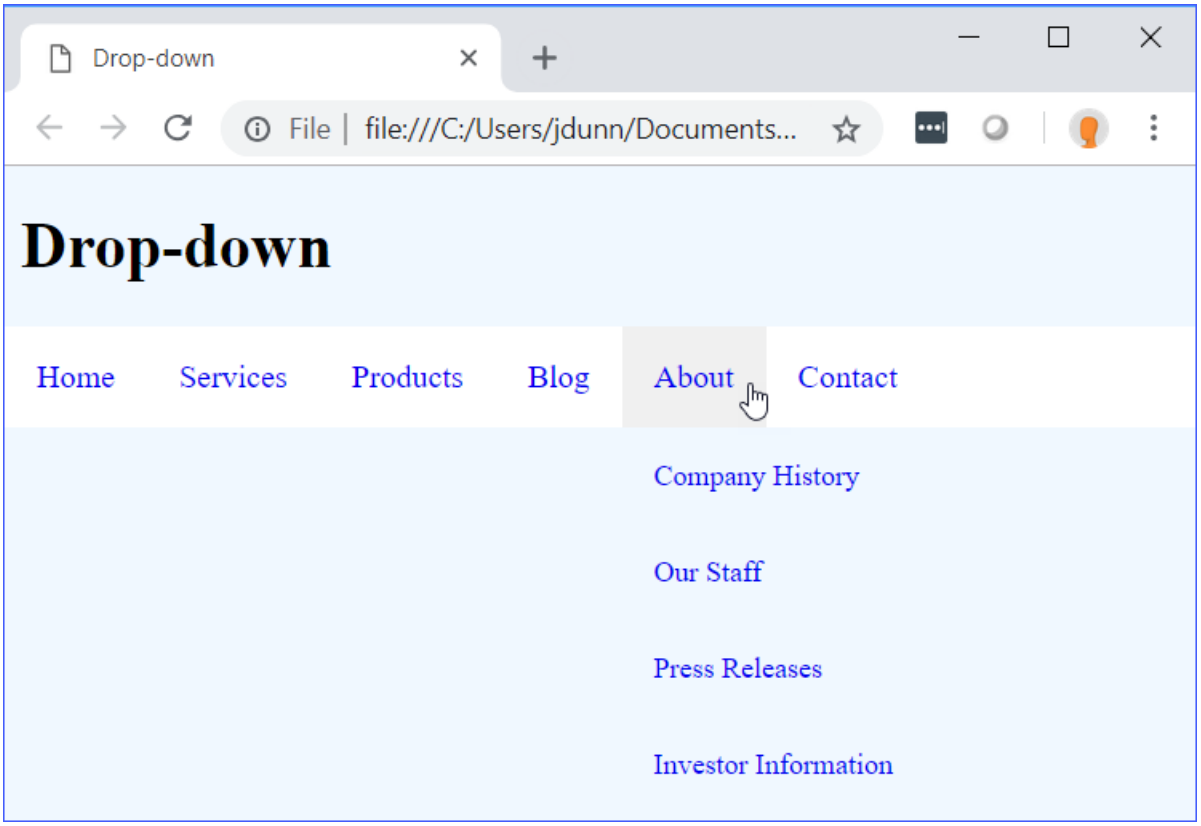


6. To keep the "About" `li` from growing and taking up so much space when its nested list shows up, make the nested list vertical by removing the `float`. Also make the font size slightly smaller for the nested `li` elements.

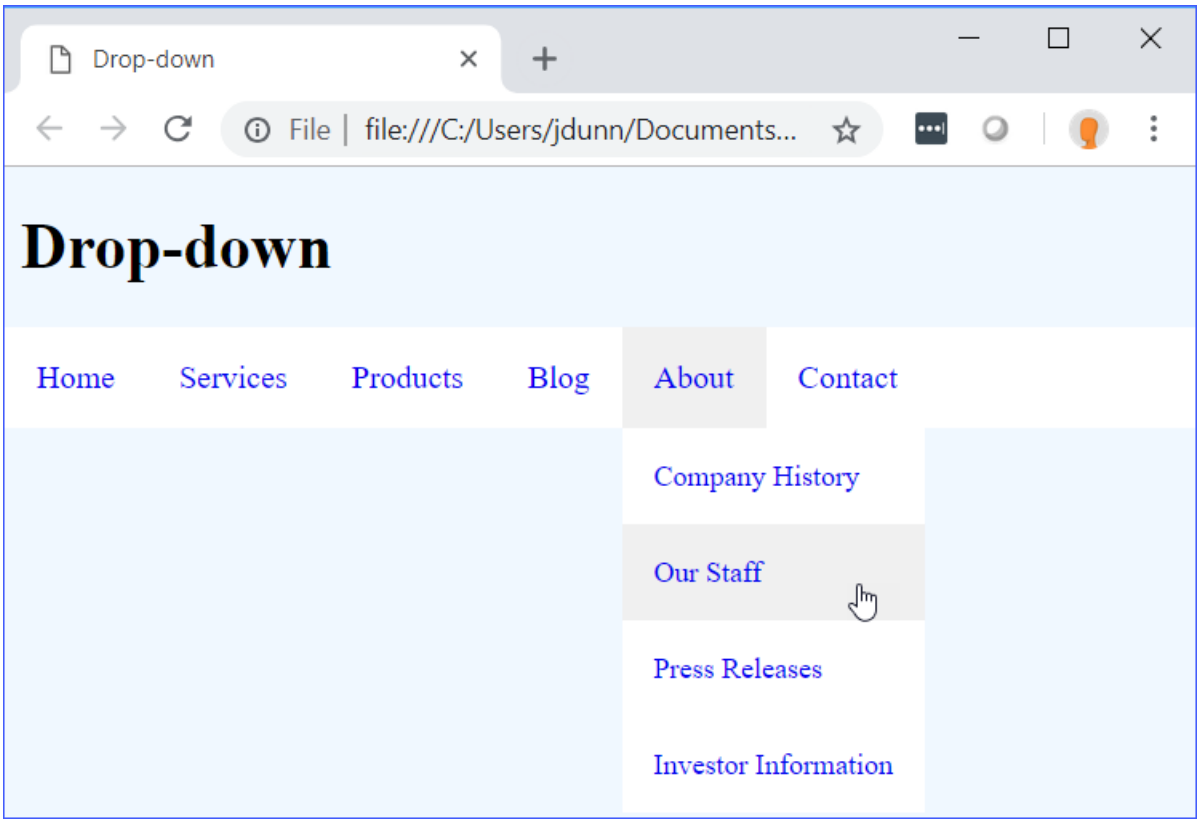


EVALUATION COPY: Not to be used in class.

7. The "About" `li` is still taking up too much space, so position the nested list absolutely to make it not affect other elements.



8. This looks okay, but make it match better with the rest of the navigation bar:
    A. Add a white background to the nested list.
    B. Change the link hover rule to a `li` hover rule, so that when a nested menu item is hovered over, the "About" `li` is still gray.



You can see the final code below:

EVALUATION COPY: Not to be used in class.

Solution: CssListMenus/Solutions/drop-down-styles.css

```
1.  body {
2.    background: aliceblue;
3.  }
4.
5.  nav {
6.    background: white;
7.    margin: 0 -0.5rem;
8.  }
9.
10. nav::after {
11.   content: "";
12.   display: block;
13.   clear: both;
14. }
15.
16. nav ul {
17.   list-style: none;
18.   margin: 0;
19.   padding: 0;
20. }
21.
22. nav li {
23.   float: left;
24. }
25.
26. nav a {
27.   display: block;
28.   padding: 1rem;
29.   text-decoration: none;
30. }
31.
32. nav li:hover {
33.   background: rgb(240, 240, 240);
34. }
35.
36. nav a:active {
37.   color: #cce;
38. }
39.
40. li.contains ul {
41.   display: none;
42. }
43.
44. li.contains:hover ul {
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
45.     background: white;
46.     display: block;
47.     position: absolute;
48.   }
49.
50.   li.contains li {
51.     float: none;
52.     font-size: 0.9rem;
53.   }
```

Solution: CssListMenus/Solutions/drop-down.html

```
      -------Lines 1 through 9 Omitted-------
10.   <body class="webucator">
11.   <h1>Drop-down</h1>
12.   <nav>
13.     <ul>
14.       <li><a href="index.html">Home</a></li>
15.       <li><a href="services.html">Services</a></li>
16.       <li><a href="products.html">Products</a></li>
17.       <li><a href="blog.html">Blog</a></li>
18.       <li class="contains"><a href="about.html">About</a>
19.         <ul>
20.           <li><a href="about.html#history">Company History</a></li>
21.           <li><a href="about.html#staff">Our Staff</a></li>
22.           <li><a href="about.html#press-releases">Press Releases</a></li>
23.           <li><a href="about.html#investor-info">Investor Information</a></li>
24.         </ul></li>
25.       <li><a href="contact.html">Contact</a></li>
26.     </ul>
27.   </nav>
28.   </body>
29.   </html>
```

Evaluation Copy

Exercise 18: Basic Fly-out Menu

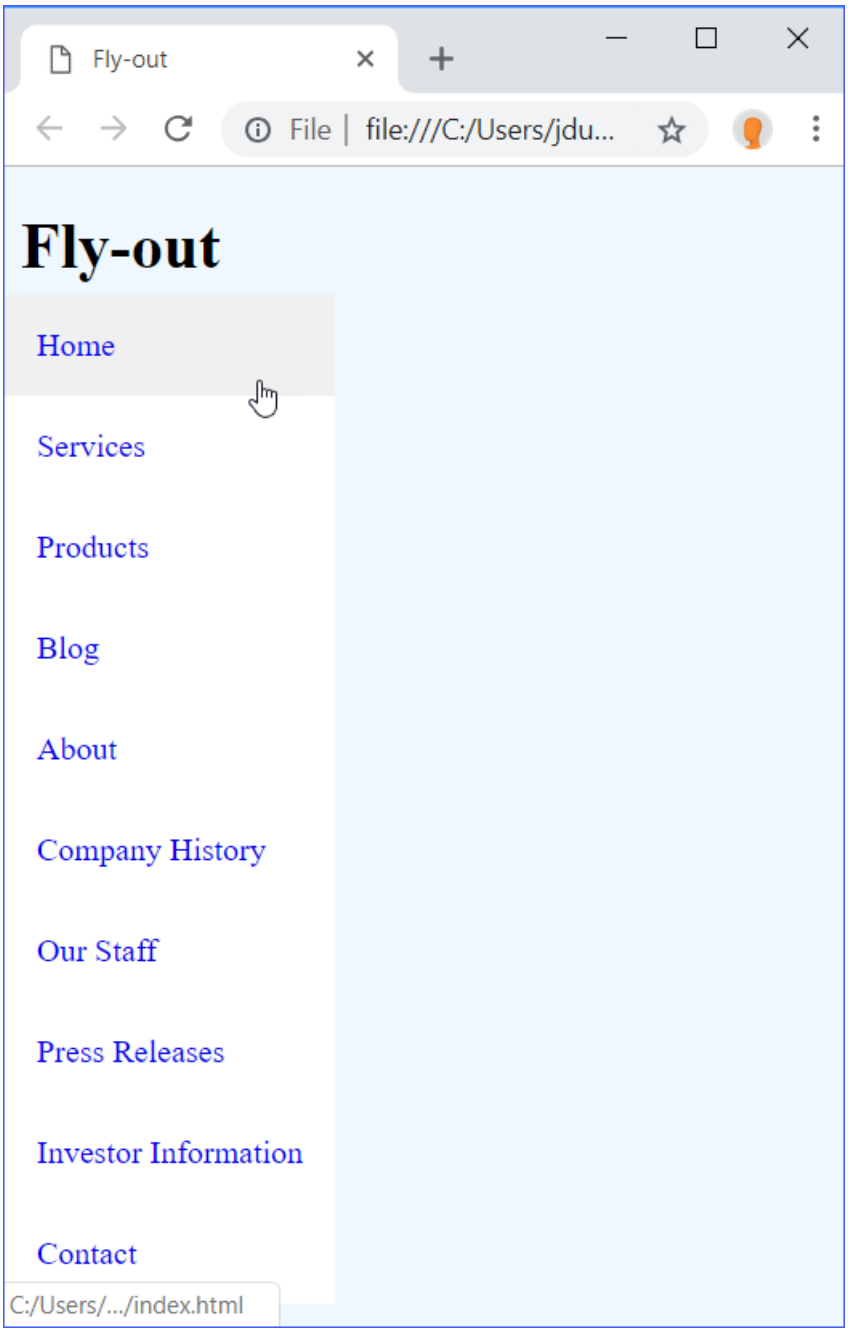
10 to 20 minutes

We will start with the vertical navigation bar that we already created, and add a fly-out menu, so that it looks like this:

Follow these steps:

EVALUATION COPY: Not to be used in class.

1. Open `CssListMenus/Exercises/fly-out-styles.css` and `CssListMenus/Exercises/fly-out.html` in your editor.
2. First, change the HTML to add the nested list to the "About" `li` and add a `contains` class to the "About" `li`.



3. Style the nested list for when it flies out:

A. Position it absolutely where you want the nested list to show up.

B. Give it a white background.

C. Make the font size smaller.

4. Hide the nested `ul` by setting `display` to `none` and make it reappear when "About" is hovered over.

EVALUATION COPY: Not to be used in class.

5. Change the link hover rule to a `li` hover rule, so that when a nested menu item is hovered over, the "About" `li` is still gray.

You can see the final code below:

EVALUATION COPY: Not to be used in class.

Solution: CssListMenus/Solutions/fly-out-styles.css

```
1.  body {
2.    background: aliceblue;
3.  }
4.
5.  nav {
6.    background: white;
7.    position: absolute;
8.    top: 4rem;
9.    margin: 0 -0.5rem;
10. }
11.
12. nav ul {
13.   list-style: none;
14.   margin: 0;
15.   padding: 0;
16.   position: relative;
17. }
18.
19. nav a {
20.   display: block;
21.   padding: 1rem;
22.   text-decoration: none;
23. }
24.
25. nav li:hover {
26.   background: rgb(240, 240, 240);
27. }
28.
29. nav a:active {
30.   color: #cce;
31. }
32.
33. .contains ul {
34.   background: white;
35.   display: none;
36.   font-size: 0.9rem;
37.   left: 88px;
38.   position: absolute;
39.   top: 201px;
40.   width: 150px;
41. }
42.
43. .contains:hover ul {
44.   display: block;
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
45.  }
```

Solution: CssListMenus/Solutions/fly-out.html

```
     -------Lines 1 through 9 Omitted-------
10.  <body class="webucator">
11.  <h1>Fly-out</h1>
12.  <nav>
13.    <ul>
14.      <li><a href="index.html">Home</a></li>
15.      <li><a href="services.html">Services</a></li>
16.      <li><a href="products.html">Products</a></li>
17.      <li><a href="blog.html">Blog</a></li>
18.      <li class="contains"><a href="about.html">About</a>
19.        <ul>
20.          <li><a href="about.html#history">Company History</a></li>
21.          <li><a href="about.html#staff">Our Staff</a></li>
22.          <li><a href="about.html#press-releases">Press Releases</a></li>
23.          <li><a href="about.html#investor-info">Investor Information</a></li>
24.        </ul>
25.      </li>
26.      <li><a href="contact.html">Contact</a></li>
27.    </ul>
28.  </nav>
29.  </body>
30.  </html>
```

Evaluation Copy

Exercise 19: CSS List Menus

25 to 40 minutes

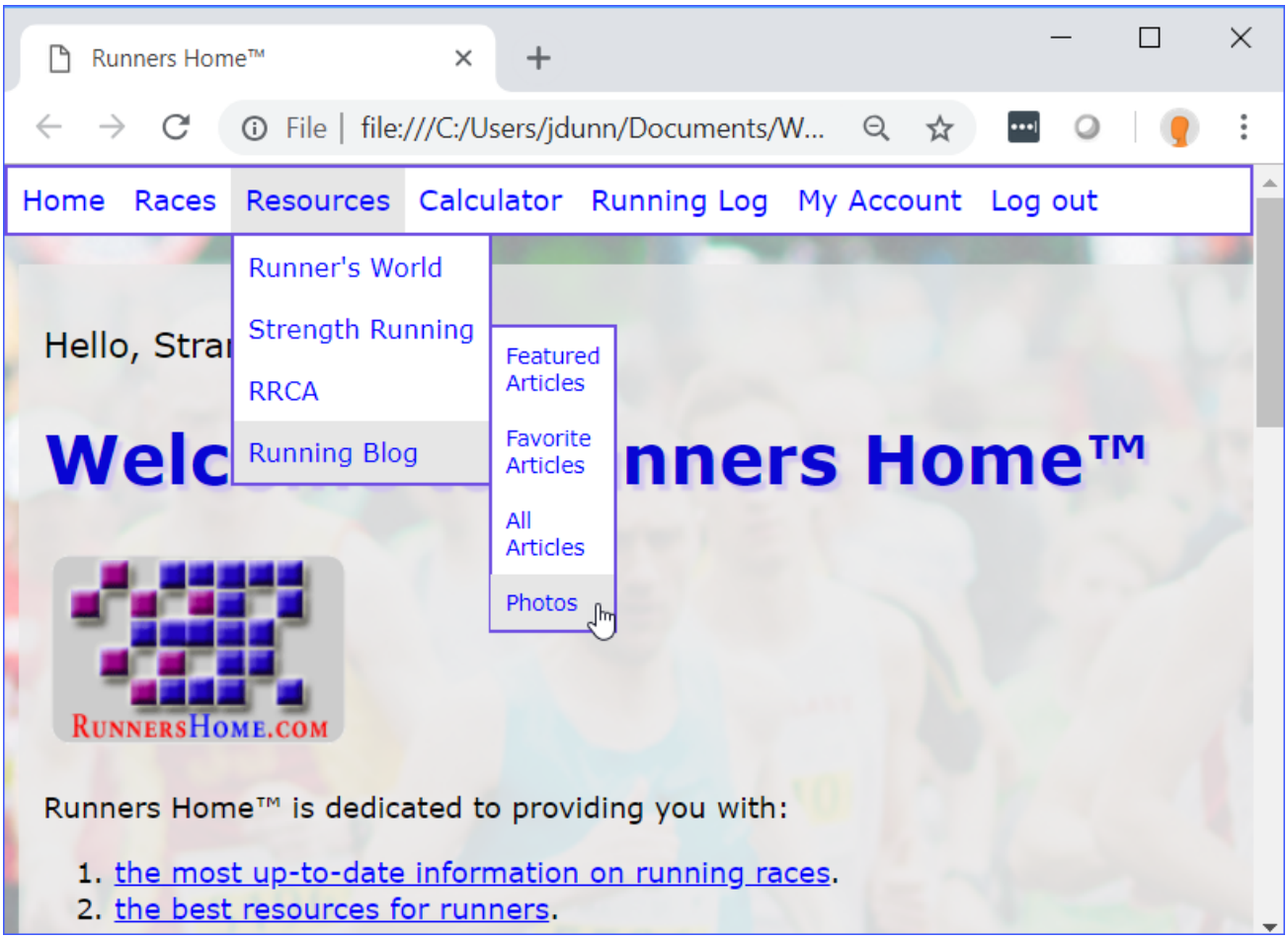
In this exercise, you will go back to working on the Runners Home home page.

1. Open `CssListMenus/Exercises/index.html` in your editor. Note that nested lists have been added to the top navigation.
2. Open `CssListMenus/Exercises/styles.css`. Feel free to continue working from your own stylesheet.
3. Using your new knowledge of list menus and navigation bars, restyle the Runners Home navigation. The object of this exercise is to practice creating dynamic navigation bars. See if you can find a way to combine drop-down and fly-out menus.
4. When you are done, open `index.html` in your browser to see the results. You are welcome to go back to the code and continue to work.

Evaluation Copy

You can design it however you like, or you can try to make it look something like:

EVALUATION COPY: Not to be used in class.


Runners Home™
file:///C:/Users/jdunn/Documents/W...
Home
Races
Resources
Calculator
Running Log
My Account
Log out
Runner's World
Strength Running
RRCA
Running Blog
Featured Articles
Favorite Articles
All Articles
Photos
Hello, Stra
Welc nners Home™
RUNNERSHOME.COM
Runners Home™ is dedicated to providing you with:
1. the most up-to-date information on running races.
2. the best resources for runners.

Solution: CssListMenus/Solutions/styles.css

```
     -------Lines 1 through 9 Omitted-------
10.
11.  nav {
12.    background-color: white;
13.    border: solid rgb(110, 78, 226) 0.1rem;
14.    margin: -1rem -0.5rem 1rem;
15.  }
16.
17.  nav ul {
18.    list-style: none;
19.    margin: 0;
20.    padding: 0;
21.  }
22.
23.  nav li {
24.    float: left;
25.  }
26.
27.  nav::after {
28.    content: "";
29.    display: block;
30.    clear: both;
31.  }
32.
33.  nav a {
34.    display: block;
35.    padding: 0.5rem;
36.    text-decoration: none;
37.  }
38.
39.  nav li:hover {
40.    background: rgb(230, 230, 230);
41.  }
42.
43.  nav a:active {
44.    color: red;
45.  }
46.
47.  #resources-li ul {
48.    background: white;
49.    border-color: rgb(110, 78, 226);
50.    border-style: none solid solid;
51.    border-width: 0.1rem;
52.    display: none;
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
53.     position: absolute;
54.     top: 40px;
55.   }
56.
57.   #resources-li li {
58.     float: none;
59.     font-size: 0.9rem;
60.   }
61.
62.   #blog-li ul {
63.     border-top: solid rgb(110, 78, 226) 0.1rem;
64.     position: absolute;
65.     top: 50px;
66.     left: 143px;
67.   }
68.
69.   #blog-li ul li {
70.     font-size: smaller;
71.   }
72.
73.   #resources-li:hover>ul {
74.     display: block;
75.   }
76.
77.   #blog-li:hover>ul {
78.     display: block;
79.   }
80.
      -------Lines 81 through 230 Omitted-------
```

Evaluation Copy

Code Explanation

We omitted all lines in the solution, except those related to the navigation bar.

Conclusion

In this lesson, you have learned to use lists to create dynamic horizontal and vertical navigation bars.

EVALUATION COPY: Not to be used in class.

LESSON 14
Media Queries

Topics Covered

☑ Media queries.

☑ Font Awesome.

☑ Breakpoints.

☑ Media types.

Introduction

In this lesson, you will learn how to use media queries to make your HTML pages look good on different devices at different dimensions. You will also learn about Font Awesome, a great resource for glyphs.

Evaluation Copy

14.1. What Are Media Queries?

Media queries are essentially conditionals that check for media features and allow us to style our pages based on those features. With the advent of mobile devices like smartphones and tablets, it became important to target these different devices to create websites that worked for them. Media queries were essential to this process as they can be used to specify styles based on more than a dozen different media features[51] (e.g., `aspect-ratio`, `resolution`, `orientation`, `resolution`, etc.). However, with the development of more flexible layouts like Flexible Box Layout Module[52] and Grid Layout[53], it has become less important to use all of these features to target specific devices. Instead, it is recommended to use media queries to add breakpoints where your design breaks.

❖ 14.1.1. Breakpoints

Breakpoints are specified `width` lengths where your layout changes to accommodate the larger or smaller viewport. Breakpoints should be based on your design and content and not on specific devices.

51. https://developer.mozilla.org/en-US/docs/Web/CSS/@media#Media_features
52. https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox
53. https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Grids

You want to target the widths where things start to go wrong in your layout (e.g., elements overlap or become too small, lines become too long to read comfortably, a lot of extra space is available, etc.).

❖ 14.1.2. Media Types

Media queries can also be used to style pages for different media types[54]. A common example of this is creating a different design for how a website looks when it is printed. This may include removing darker background colors, removing unnecessary images and advertisements, adjusting the font to be better suited for printing, etc.

❖ 14.1.3. Syntax

Media queries are performed using the `@media` at-rule. The syntax is below:

```
@media media-type {
  /* rules for specified media type
    (e.g., print) */
}
/* OR */
@media (media feature condition) {
  /* rules for specified media feature condition
    (e.g., max-width: 500px ) */
}
/* OR */
@media media-type and (media feature condition) {
  /* rules for specified media-type
    and media feature condition */
}
```

Evaluation Copy

14.2. Font Awesome

`Font Awesome (https://fontawesome.com)` provides a collection of free vector icons that you can use on your websites. You can get access to these icons through a free content delivery network (CDN). To do so, you need to go to their website to get a `<script>` tag with a kit code, which will looks something like this:

54. https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries#Media_types

EVALUATION COPY: Not to be used in class.

```
<script src="https://kit.fontawesome.com/kit-code.js"></script>
```

To get the `<link>` tag with the latest version of Font Awesome:

1. Go to `https://fontawesome.com`.
2. Click the **Start** menu item at the top of the page.
3. Enter your email address and follow the directions they give you. You will need to create an account with a password, but you should not need to provide further personal details.
4. Paste the `script` element into the `head` of your HTML:

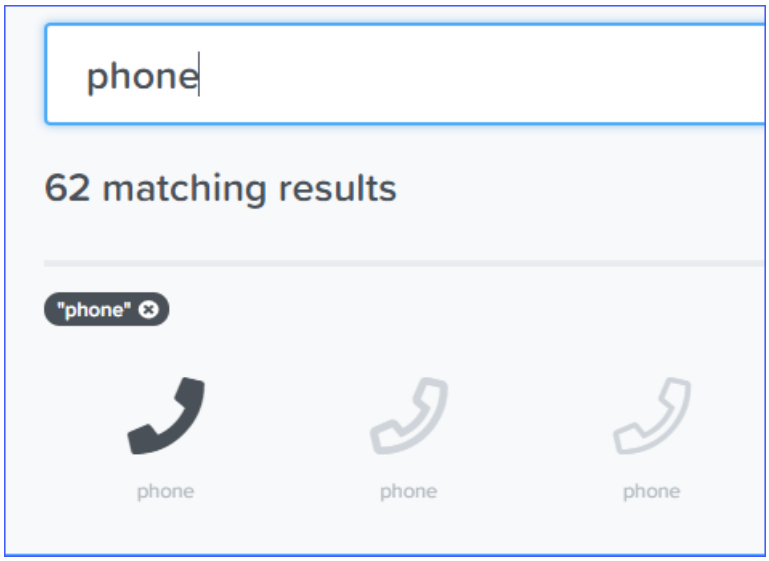
```
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,initial-scale=1">
<link rel="stylesheet" href="../normalize.css">
<link rel="stylesheet" href="../styles.css">
<script src="https://kit.fontawesome.com/kit-code.js"></script>
<title>Font Awesome</title>
</head>
```

Evaluation Copy

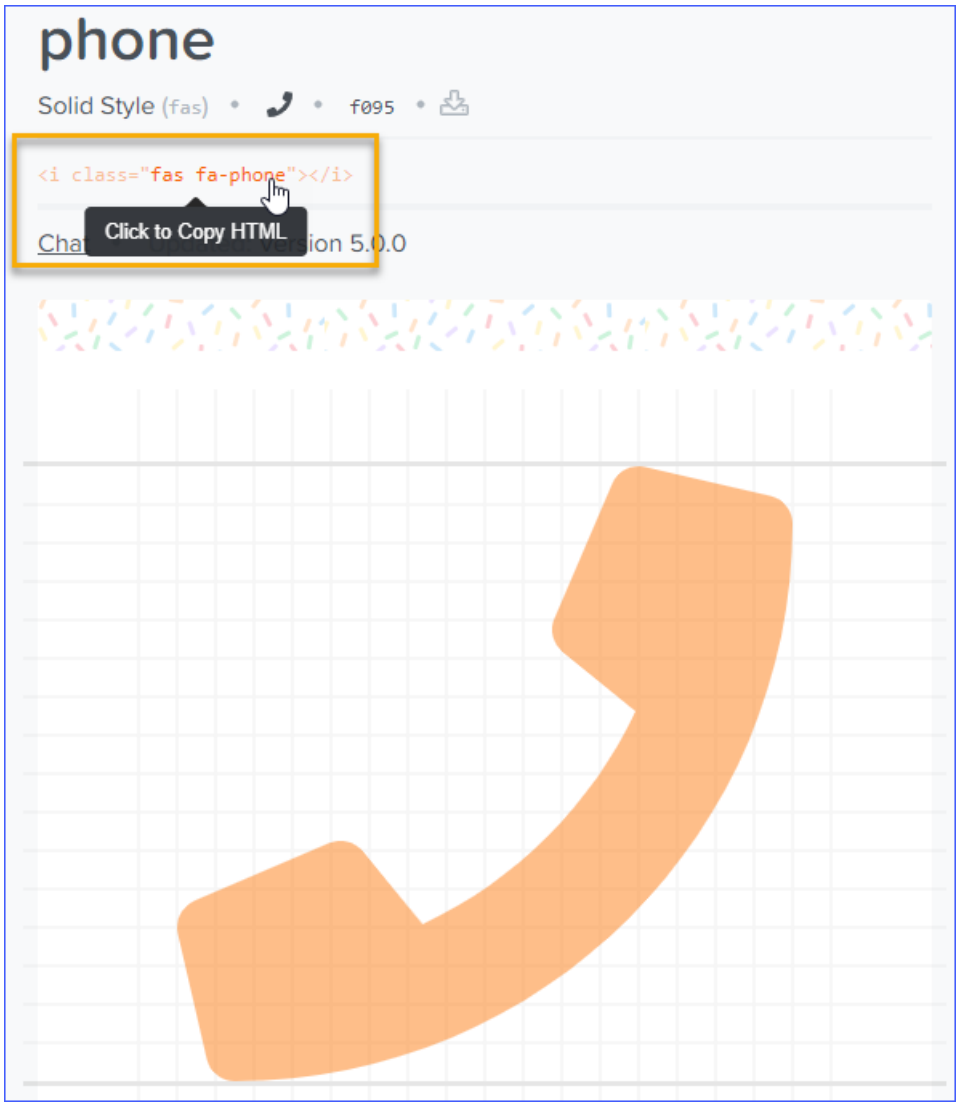
❖ 14.2.1. Finding and Using Icons

To find Font Awesome icons:

1. Use the search feature at `https://fontawesome.com/icons`:



2. Click the icon you want to use.
3. Click the HTML code snippet on the icon page to copy it:



4. Then paste that code snippet wherever you want the icon to show up.

Below is a page containing several Font Awesome icons:

EVALUATION COPY: Not to be used in class.

Demo 14.1: MediaQueries/Demos/font-awesome.html

```
1.   <!DOCTYPE html>
2.   <html lang="en">
3.   <head>
4.   <meta charset="UTF-8">
5.   <meta name="viewport" content="width=device-width,initial-scale=1">
6.   <link rel="stylesheet" href="../normalize.css">
7.   <script src="https://kit.fontawesome.com/f804978a2a.js"></script>
8.   <title>Font Awesome</title>
9.   </head>
10.  <body class="webucator">
11.  <main>
12.    <i class="fas fa-phone"></i>
13.    <i class="fas fa-envelope-square"></i>
14.    <i class="fas fa-search"></i>
15.    <i class="fab fa-grunt"></i>
16.    <i class="fab fa-facebook"></i>
17.    <i class="fas fa-chevron-up"></i>
18.    <i class="fas fa-chevron-down"></i>
19.  </main>
20.  </body>
21.  </html>
```

Evaluation Copy

Code Explanation

The icons on this page are shown below:

Open `MediaQueries/Demos/font-awesome.html` in your browser to see the web page.

Exercise 20: Breakpoints

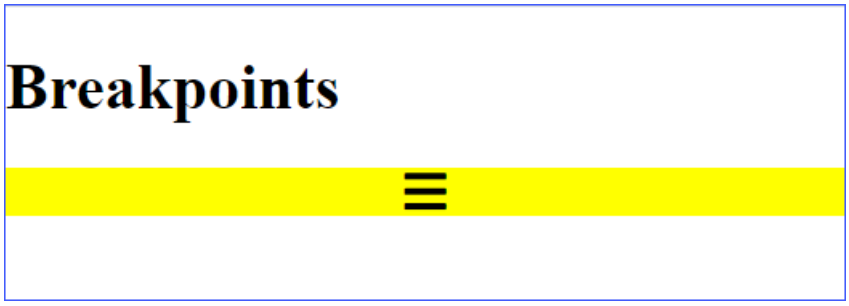
10 to 20 minutes

Let's look at how we could use a media query to create a breakpoint between a laptop menu layout and a mobile layout.

We are going to replace a standard horizontal menu with a vertical mobile menu that shows up when the user hovers over the Font Awesome bars icon[55].

Here is the standard horizontal menu we are starting with:

Here is the vertical mobile menu that is hidden, so only the bars icon is showing:



Here is the vertical mobile menu being shown:

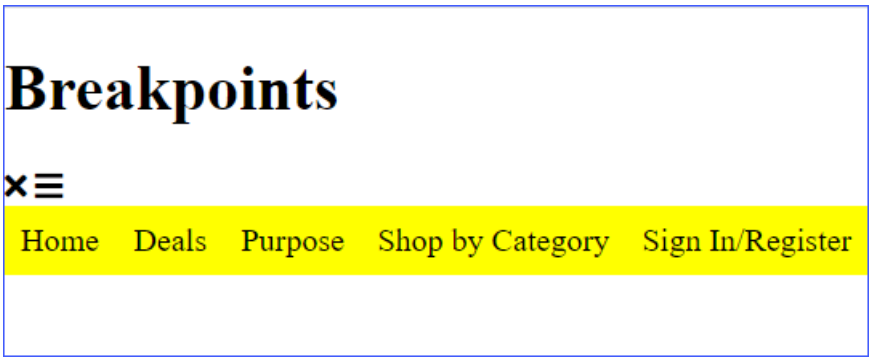
55. https://fontawesome.com/icons/bars?style=solid

EVALUATION COPY: Not to be used in class.

Follow these steps:

Evaluation Copy

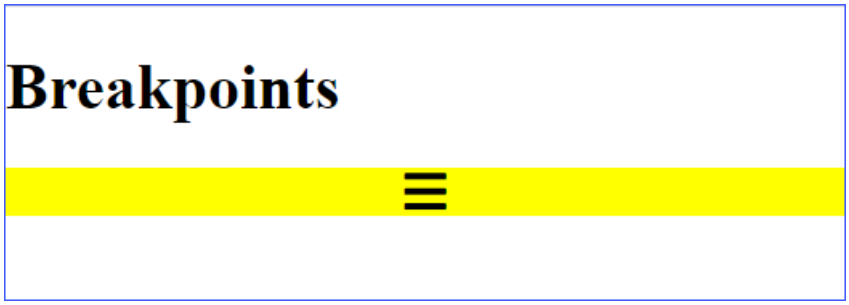
1. Open `MediaQueries/Exercises/breakpoints-styles.css` and `MediaQueries/Exercises/breakpoints.html` in your editor.
2. In `MediaQueries/Exercises/breakpoints.html`, uncomment the "menu-icon" `div`. This reveals two `Font Awesome (https://fontawesome.com)` icons that will be used to toggle the navigation menu in mobile view.



3. In `MediaQueries/Exercises/breakpoints.html`, uncomment the `script` element in the `head`. This is JavaScript that makes toggling work. Watch how the navigation disapears when the screen becames smaller than `500px` and try click the bars and times icons to display and hide the navigation.

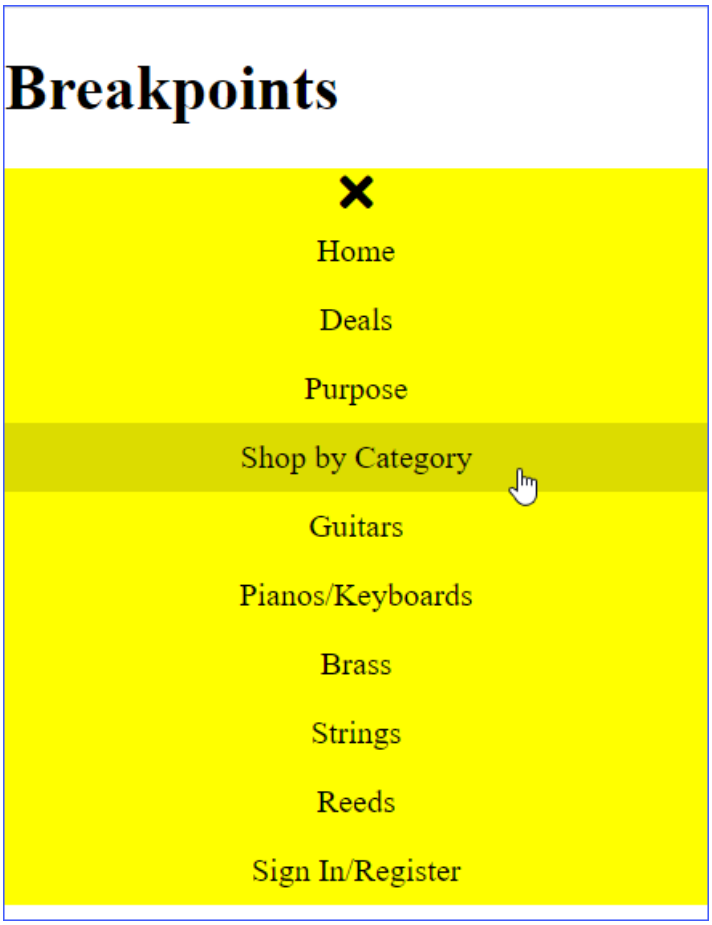
4. Now, let's use CSS to make it look good. At the bottom of `MediaQueries/Exercises/breakpoints-styles.css` create the `@media` rule targeting viewports no larger than `500px` (`@media (max-width: 500px)`).
5. In the new `@media` rule, style the menu icons by targeting the `nav #menu-icon`:
    A. Make the background yellow and the color black.
    B. Set `cursor` (`https://developer.mozilla.org/en-US/docs/Web/CSS/cursor`) to `pointer`.
    C. Make the font size bigger (`1.5rem`).
    D. Center the icons by setting `text-align` to `center`.

Evaluation Copy



6. In the media query, change the `flex-direction` to `column` of the `nav ul` to make it vertical.

EVALUATION COPY: Not to be used in class.



7. This doesn't look quite right. The nested `ul` is visible. Hide it by setting the `display` of `ul` elements within `li.contains` to `none`.

You can see the final code below:

EVALUATION COPY: Not to be used in class.

Solution: MediaQueries/Solutions/breakpoints-styles.css

```
      -------Lines 1 through 40 Omitted-------
41.
42.   @media (max-width: 500px) {
43.     nav #menu-icon {
44.       background: yellow;
45.       color: black;
46.       cursor: pointer;
47.       font-size: 1.5rem;
48.       text-align: center;
49.     }
50.
51.     nav ul {
52.       flex-direction: column;
53.     }
54.
55.     nav ul li.contains ul {
56.       display: none;
57.     }
58.   }
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

Solution: MediaQueries/Solutions/breakpoints.html

```
1.  <!DOCTYPE HTML>
2.  <html>
3.  <head>
4.  <meta charset="UTF-8">
5.  <meta name="viewport" content="width=device-width,initial-scale=1">
6.  <link href="../normalize.css" rel="stylesheet">
7.  <link href="breakpoints-styles.css" rel="stylesheet">
8.  <script src="toggle.js"></script>
9.  <!-- You may need to replace the tag below with your code. -->
10. <script src="https://kit.fontawesome.com/f804978a2a.js"></script>
11. <title>Breakpoints</title>
12. </head>
13. <body>
14. <h1>Breakpoints</h1>
15. <nav>
16.   <div id="menu-icon">
17.     <i class="fas fa-times"></i>
18.     <i class="fas fa-bars"></i>
19.   </div>
20.   <ul>
21.     <li><a href="#">Home</a></li>
22.     <li><a href="#deals">Deals</a></li>
23.     <li><a href="#purpose">Purpose</a></li>
24.     <li class="contains"><a href="#">Shop by Category</a>
25.       <ul>
26.         <li><a href="#">Guitars</a></li>
27.         <li><a href="#">Pianos/Keyboards</a></li>
28.         <li><a href="#">Brass</a></li>
29.         <li><a href="#">Strings</a></li>
30.         <li><a href="#">Reeds</a></li>
31.       </ul>
32.     </li>
33.     <li><a href="#">Sign In/Register</a></li>
34.   </ul>
35. </nav>
36. </body>
37. </html>
```

Evaluation Copy

Exercise 21: Targeting Print

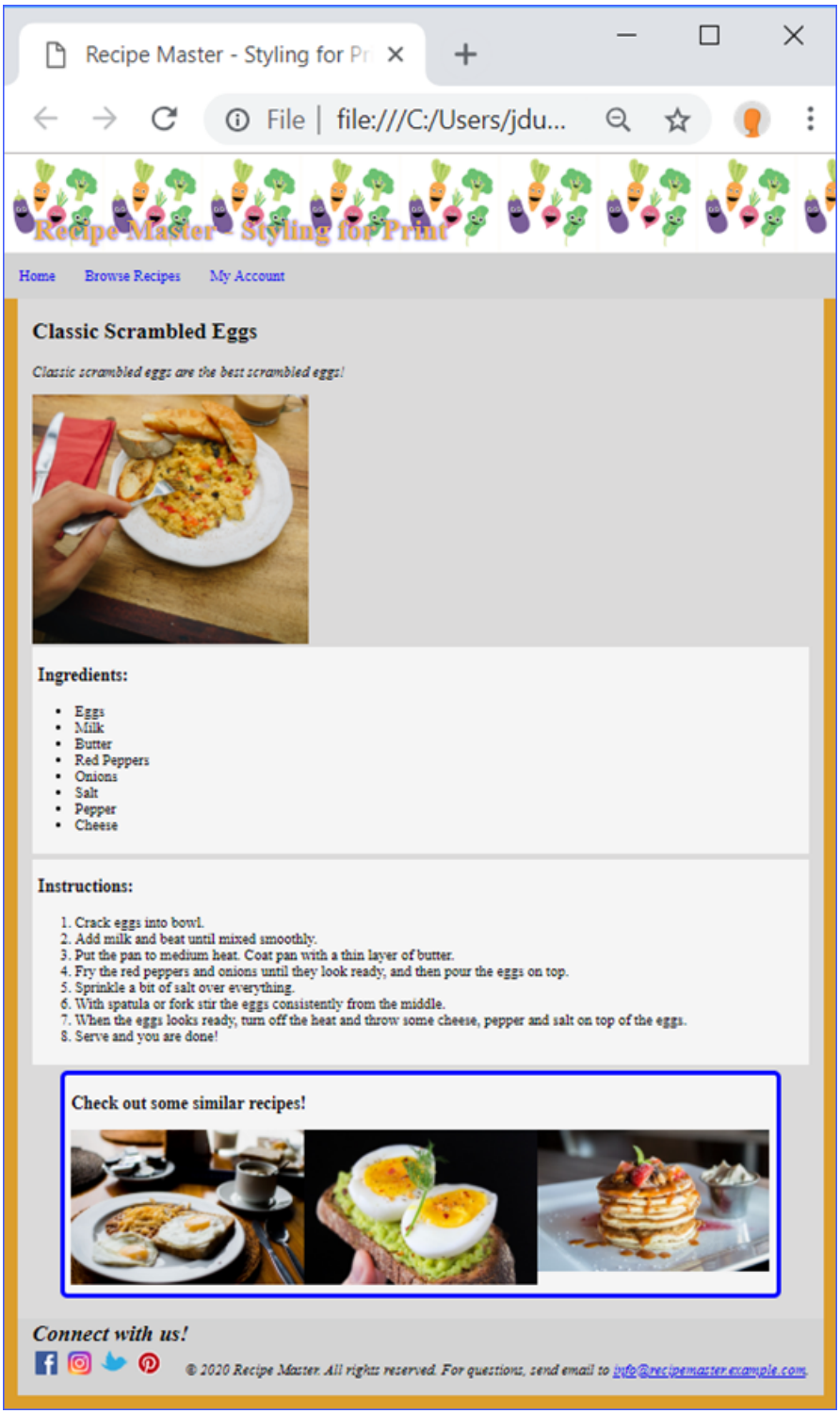
10 to 20 minutes

Evaluation Copy

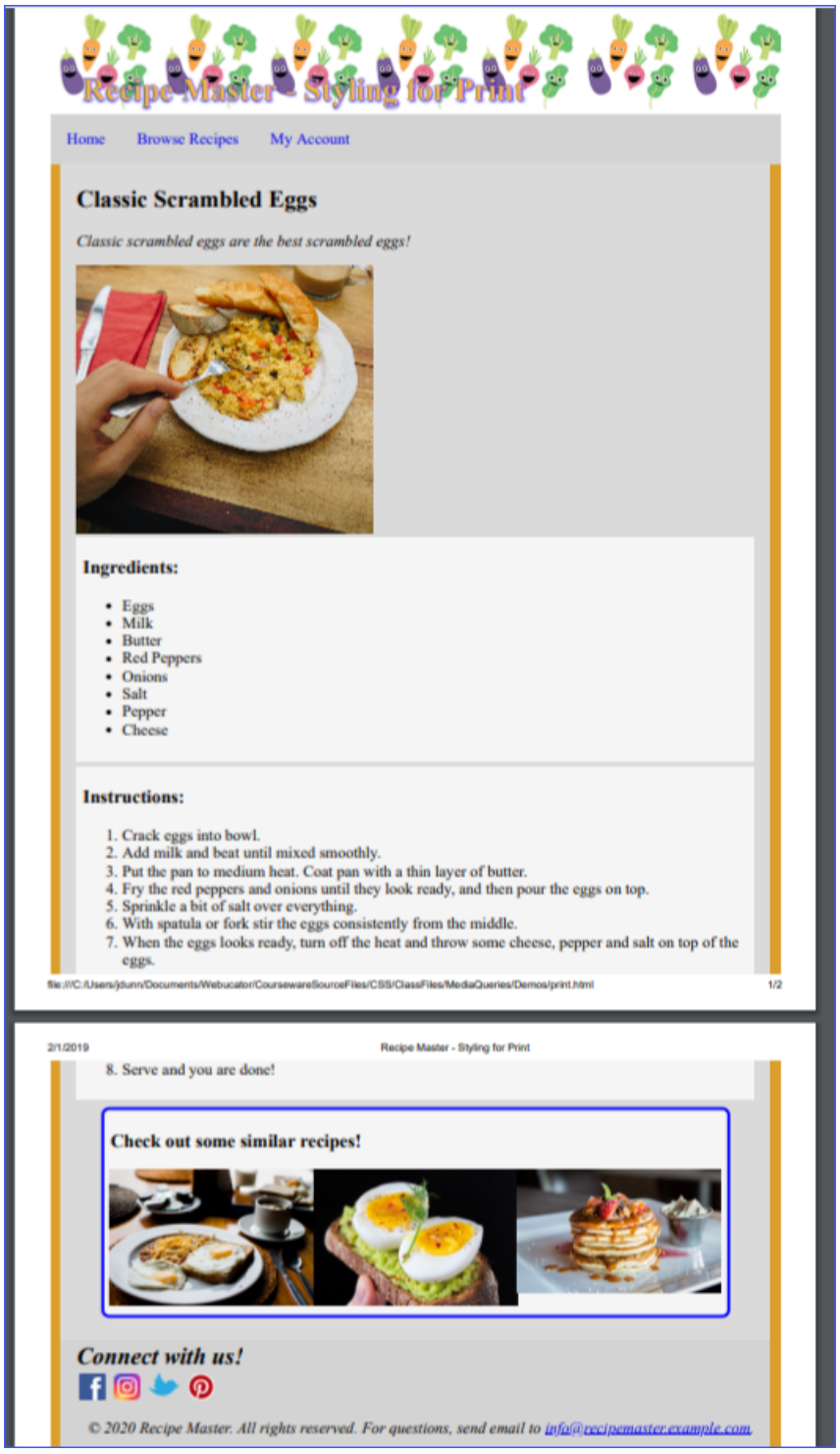
Let's look at how we might use a media query to change the print design of a web page.

Here we have a page called Recipe Master that is opened to a recipe for "Classic Scrambled Eggs":

EVALUATION COPY: Not to be used in class.

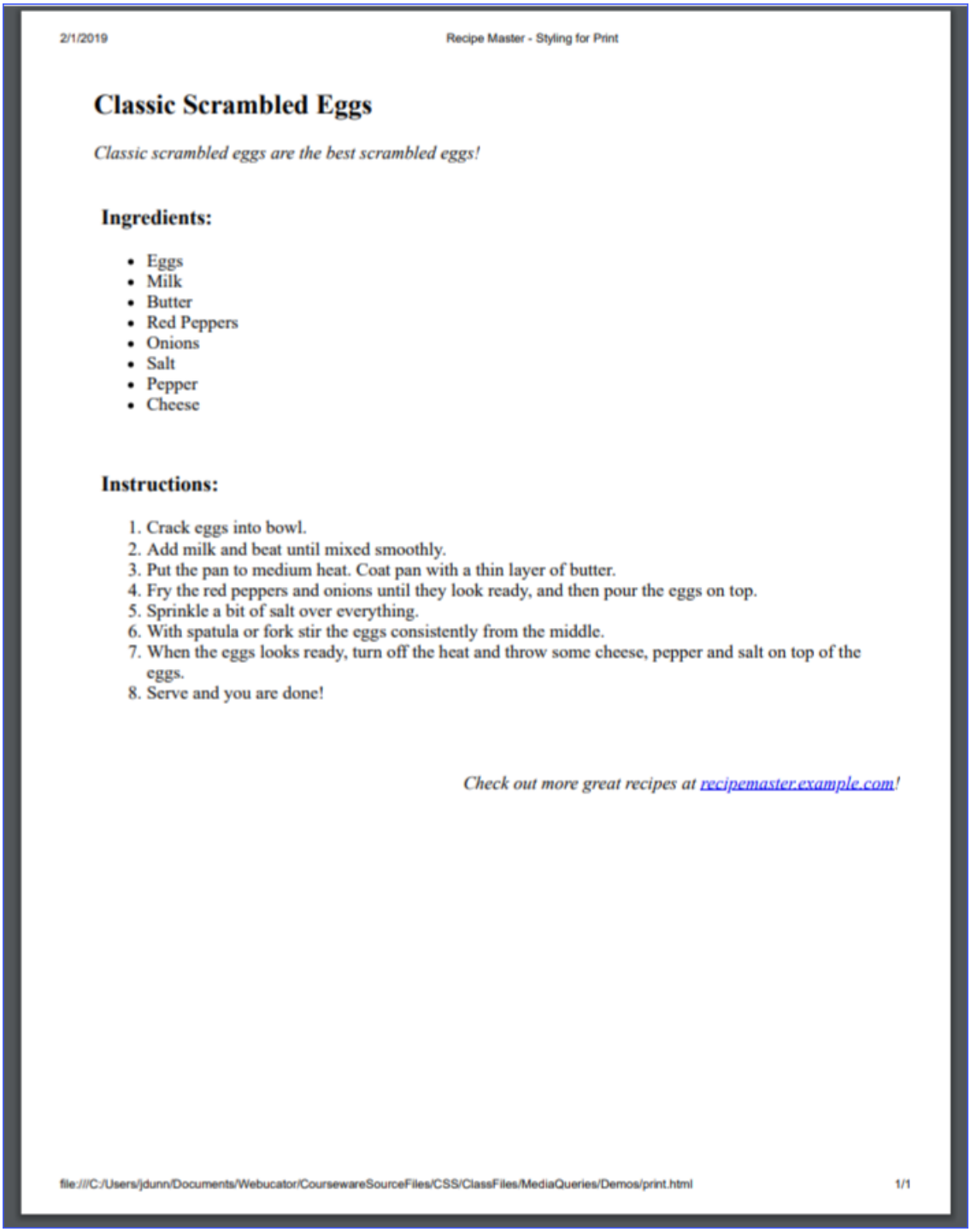


This is how the page currently looks in print preview mode:



Following these steps, make it look like the image below:

EVALUATION COPY: Not to be used in class.

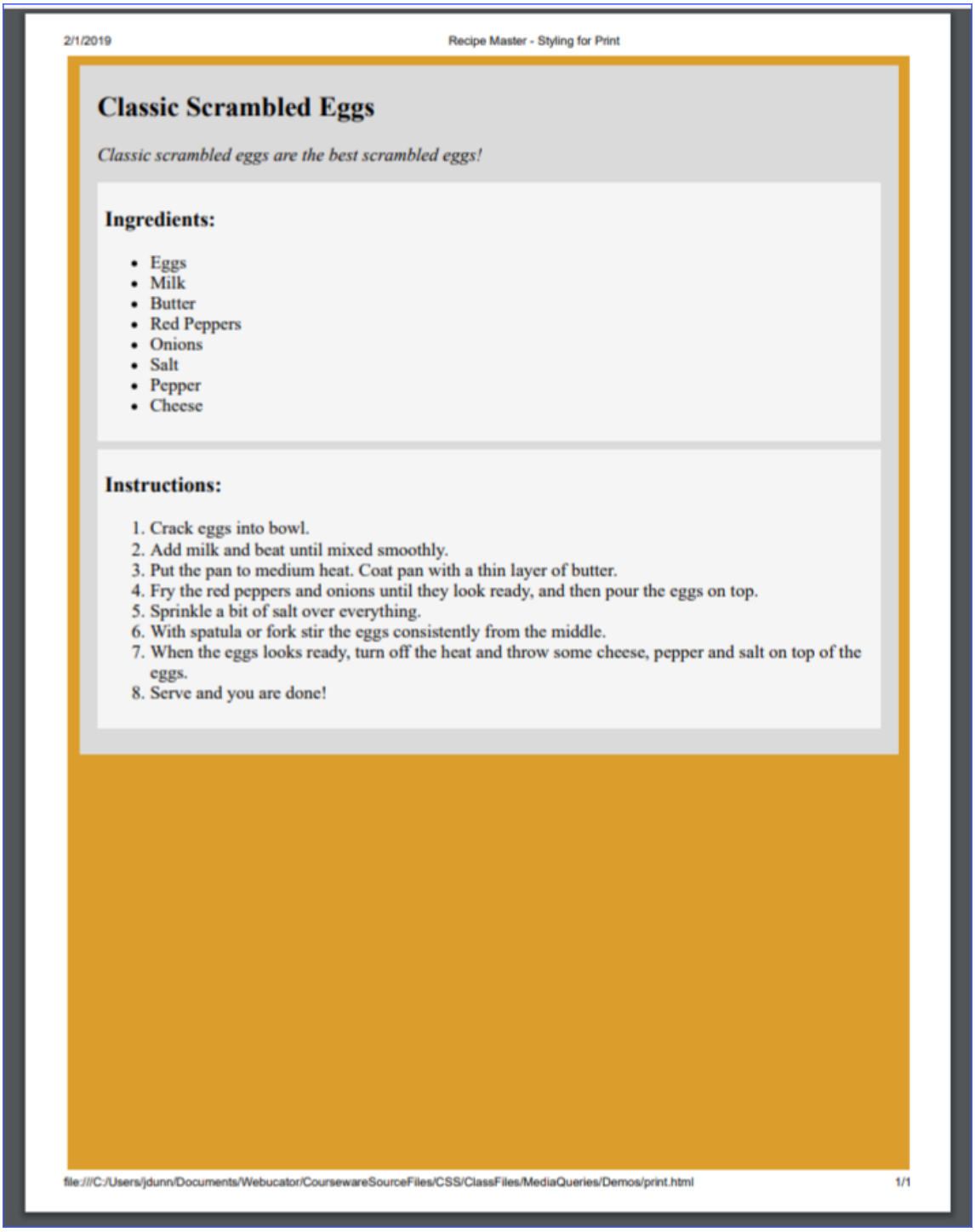


1. Open `MediaQueries/Exercises/print-styles.css` and `MediaQueries/Exercises/print.html` in your editor. Use the HTML file as reference. You will only need to make changes in the CSS file.

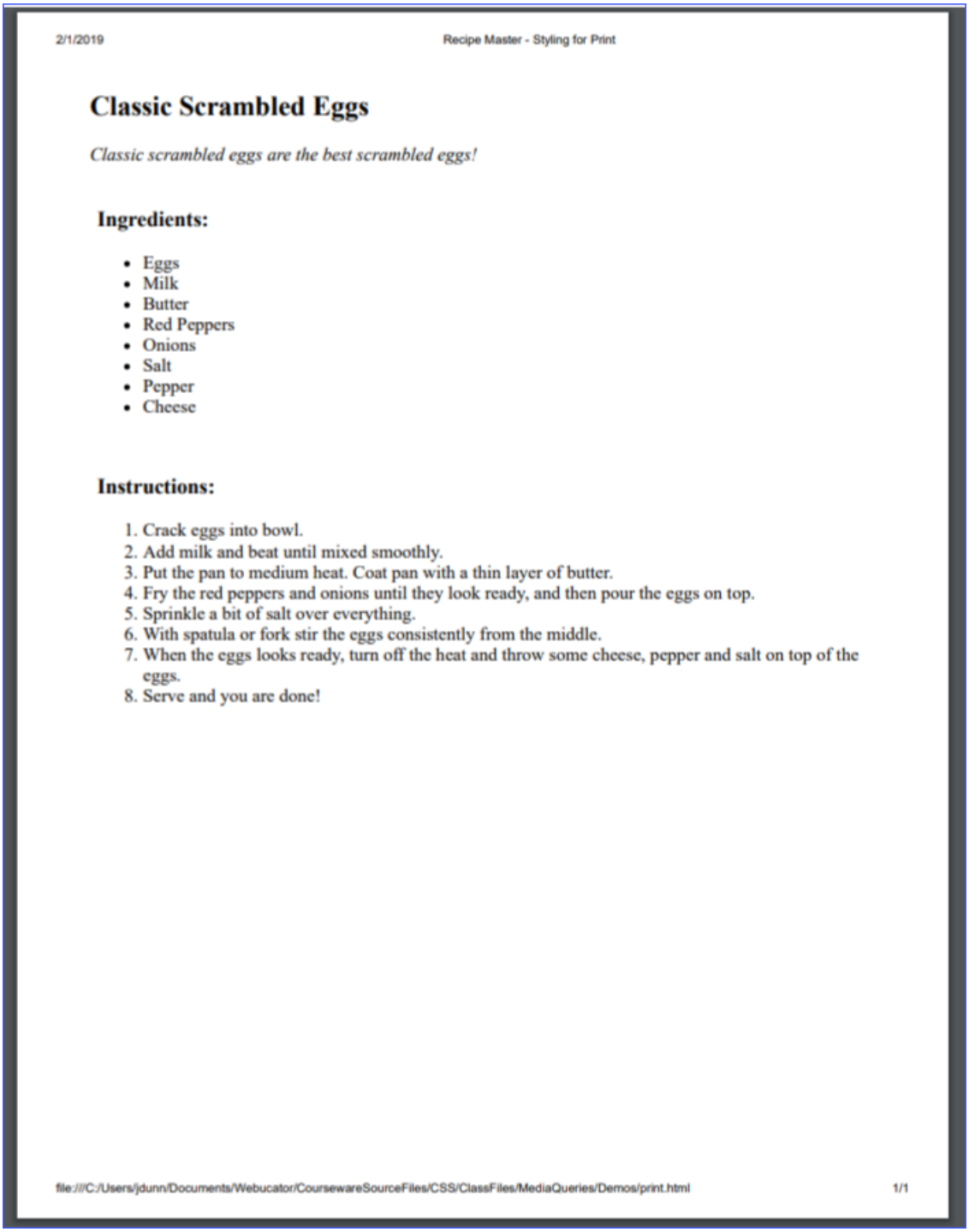
2. At the bottom of `MediaQueries/Exercises/print-styles.css` create the `@media` rule targeting `print`.
3. Add a rule that makes the `title`, `h1`, `header`, `footer`, `aside`, and `recipe-photo img` disappear (`display: none`).

Evaluation Copy

EVALUATION COPY: Not to be used in class.

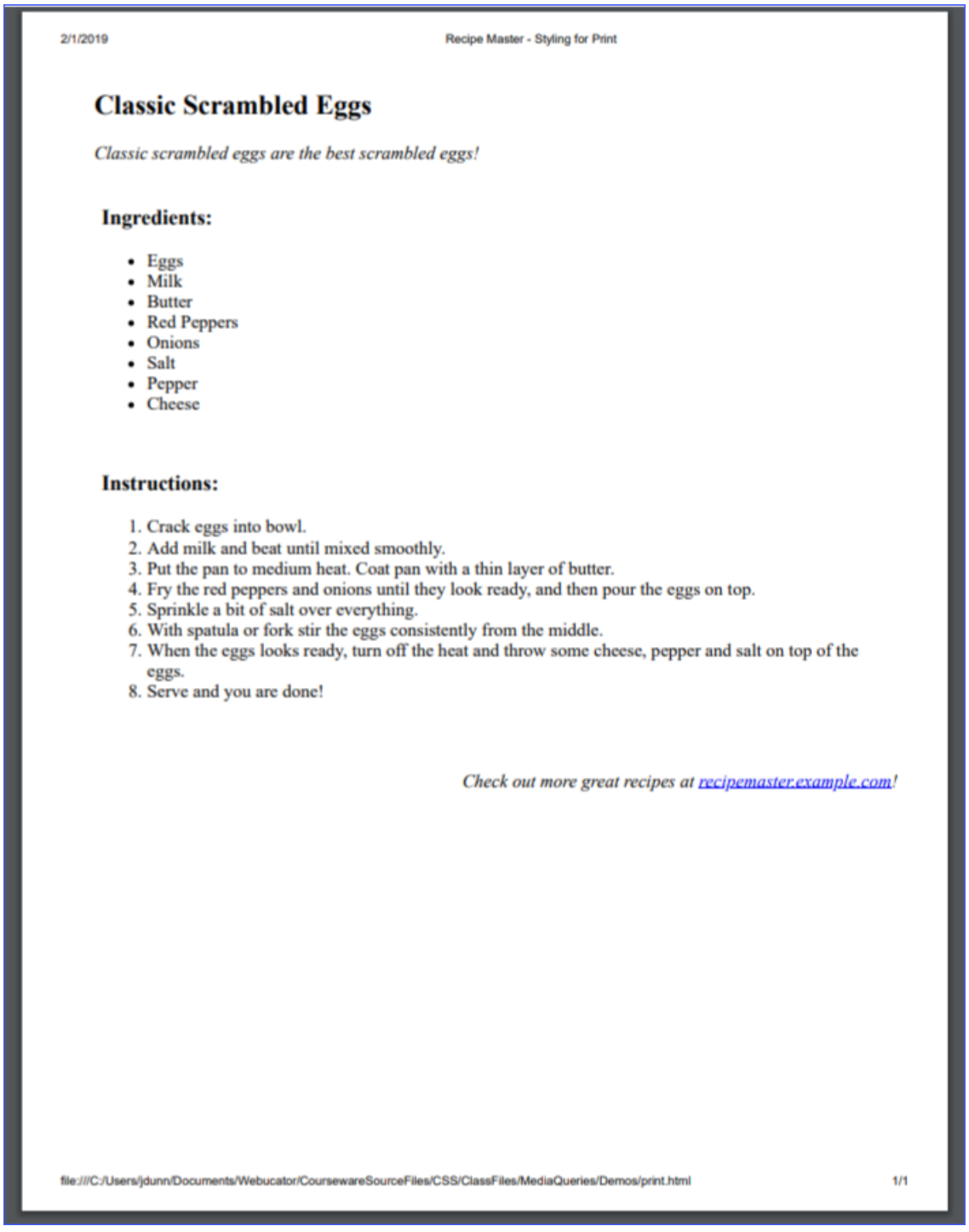


4. Add a rule that turns the orange and off-white colors to white, which is better for printing.



5. Add a rule that makes the `print-callout div` appear.

EVALUATION COPY: Not to be used in class.


2/1/2019
Recipe Master - Styling for Print
Classic Scrambled Eggs
Classic scrambled eggs are the best scrambled eggs!
Ingredients:
Eggs
Milk
Butter
Red Peppers
Onions
Salt
Pepper
Cheese
Instructions:
1. Crack eggs into bowl.
2. Add milk and beat until mixed smoothly.
3. Put the pan to medium heat. Coat pan with a thin layer of butter.
4. Fry the red peppers and onions until they look ready, and then pour the eggs on top.
5. Sprinkle a bit of salt over everything.
6. With spatula or fork stir the eggs consistently from the middle.
7. When the eggs looks ready, turn off the heat and throw some cheese, pepper and salt on top of the eggs.
8. Serve and you are done!
Check out more great recipes at recipemaster.example.com!
file:///C:/Users/jdunn/Documents/Webucator/CoursewareSourceFiles/CSS/ClassFiles/MediaQueries/Demos/print.html
1/1

Solution: MediaQueries/Solutions/print-styles.css

```
1.  body {
2.    background: rgb(219, 158, 44);
3.  }
4.
5.  h1.title {
6.    background: url(images/veggies.png) left/contain;
7.    color: rgb(240, 181, 72);
8.    padding: 2em 1em 0.2em;
9.    margin: -1rem -0.5rem 0;
10.   text-shadow: 0 0 0.2rem blue;
11. }
12.
13. header {
14.   background: lightgray;
15.   padding-right: 1rem;
16.   margin: 0 -0.5rem;
17.   width: 100%;
18. }
19.
20. nav ol {
21.   list-style: none;
22.   margin: 0;
23.   padding: 0;
24. }
25.
26. nav ol li {
27.   float: left;
28. }
29.
30. nav ol li.contains ol {
31.   background: lightgray;
32.   display: none;
33.   font-size: smaller;
34.   position: absolute;
35. }
36.
37. nav ol li.contains:hover ol {
38.   display: block;
39. }
40.
41. nav ol li ol li {
42.   float: none;
43. }
44.
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
45.  nav::after {
46.    content: "";
47.    display: block;
48.    clear: both;
49.  }
50.
51.  nav a {
52.    display: block;
53.    padding: 1rem;
54.    text-decoration: none;
55.  }
56.
57.  nav li:hover {
58.    background: rgb(209, 181, 148);
59.    transition: background 0.4s;
60.  }
61.
62.  nav a:active {
63.    color: red;
64.  }
65.
66.  main {
67.    background: rgb(219, 218, 218);
68.    margin: 0 auto 0;
69.    padding: 0.2rem 1rem 1rem;
70.    width: 95%;
71.  }
72.
73.  .subtitle {
74.    font-style: italic;
75.  }
76.
77.  .recipe-photo {
78.    width: 300px;
79.  }
80.
81.  section,
82.  aside {
83.    background: whitesmoke;
84.    padding: 0.1rem 0.4rem 0.4rem;
85.    margin: 0 auto 0.4rem;
86.  }
87.
88.  aside {
89.    border: solid blue;
```

Evaluation Copy

```
90.     border-radius: 0.5rem;
91.     width: 90%;
92.   }
93.
94.   aside::after {
95.     content: "";
96.     display: block;
97.     clear: both;
98.   }
99.
100.  aside div {
101.    float: left;
102.    width: 33.3%;
103.  }
104.
105.  aside div a {
106.    display: block;
107.    width: 100%;
108.  }
109.
110.  aside div img {
111.    width: 100%;
112.    transition: transform 0.5s;
113.  }
114.
115.  aside div img:hover {
116.    transform: scale(0.95);
117.    transition: transform 0.5s;
118.  }
119.
120.  footer {
121.    background: lightgray;
122.    font-size: medium;
123.    font-style: italic;
124.    min-width: 640px;
125.    margin: 0 auto;
126.    padding: 0.2rem 1rem;
127.    width: 95%;
128.  }
129.
130.  footer::after {
131.    content: "";
132.    display: block;
133.    clear: both;
134.  }
```

Evaluation Copy

EVALUATION COPY: Not to be used in class.

```
135.
136. footer h2 {
137.   margin: 0;
138. }
139.
140. footer a.image {
141.   margin: 0.2rem;
142. }
143.
144. footer a {
145.   text-decoration: underline;
146. }
147.
148. footer p {
149.   float: right;
150. }
151.
152. .print-callout {
153.   display: none;
154. }
155.
156. @media print {
157.   h1.title,
158.   header,
159.   footer,
160.   aside,
161.   img.recipe-photo {
162.     display: none;
163.   }
164.
165.   body,
166.   main,
167.   section {
168.     background: white;
169.   }
170.
171.   .print-callout {
172.     display: block;
173.     font-size: medium;
174.     font-style: italic;
175.     text-align: right;
176.   }
177. }
```

Evaluation Copy

Conclusion

In this lesson, you have learned:

1. What media queries are.
2. To use Font Awesome.
3. To create breakpoints.
4. To target specific media types.

Evaluation Copy

EVALUATION COPY: Not to be used in class.