W3.CSS

SUCCINCTLY

BY JOSEPH D. BOOTH

SUCCINCTLY E-BOOK SERIES

Syncfusion

W3.CSS Succinctly

By
Joseph D. Booth

Foreword by Daniel Jebaraj

Technical Reviewer: James McCaffrey
Copy Editor: Courtney Wright
Acquisitions Coordinator: Tres Watkins, content development manager, Syncfusion, Inc.
Proofreader: Jacqueline Bieringer, content producer, Syncfusion, Inc.

Table of Contents

The Story Behind the *Succinctly* Series of Books

Daniel Jebaraj, Vice President
Syncfusion, Inc.

Staying on the cutting edge

As many of you may know, Syncfusion is a provider of software components for the Microsoft platform. This puts us in the exciting but challenging position of always being on the cutting edge.

Whenever platforms or tools are shipping out of Microsoft, which seems to be about every other week these days, we have to educate ourselves, quickly.

Information is plentiful but harder to digest

In reality, this translates into a lot of book orders, blog searches, and Twitter scans.

While more information is becoming available on the Internet and more and more books are being published, even on topics that are relatively new, one aspect that continues to inhibit us is the inability to find concise technology overview books.

We are usually faced with two options: read several 500+ page books or scour the web for relevant blog posts and other articles. Just as everyone else who has a job to do and customers to serve, we find this quite frustrating.

The *Succinctly* series

This frustration translated into a deep desire to produce a series of concise technical books that would be targeted at developers working on the Microsoft platform.

We firmly believe, given the background knowledge such developers have, that most topics can be translated into books that are between 50 and 100 pages.

This is exactly what we resolved to accomplish with the *Succinctly* series. Isn't everything wonderful born out of a deep desire to change things for the better?

The best authors, the best content

Each author was carefully chosen from a pool of talented experts who shared our vision. The book you now hold in your hands, and the others available in this series, are a result of the authors' tireless work. You will find original content that is guaranteed to get you up and running in about the time it takes to drink a few cups of coffee.

Free forever

Syncfusion will be working to produce books on several topics. The books will always be free. Any updates we publish will also be free.

Free? What is the catch?

There is no catch here. Syncfusion has a vested interest in this effort.

As a component vendor, our unique claim has always been that we offer deeper and broader frameworks than anyone else on the market. Developer education greatly helps us market and sell against competing vendors who promise to “enable AJAX support with one click,” or “turn the moon to cheese!”

Let us know what you think

If you have any topics of interest, thoughts, or feedback, please feel free to send them to us at succinctly-series@syncfusion.com.

We sincerely hope you enjoy reading this book and that it helps you better understand the topic of study. Thank you for reading.

Please follow us on Twitter and “Like” us on Facebook to help us spread the word about the *Succinctly* series!

About the Author

Joseph D. Booth has been programming since 1981 in a variety of languages, including BASIC, Clipper, FoxPro, Delphi, Classic ASP, Visual Basic, Visual C#, and the .NET Framework. He has also worked in various database platforms, including DBASE, Paradox, Oracle, and SQL Server.

He is the author of *GitHub Succinctly*, *Accounting Succinctly*, *Regular Expressions Succinctly*, and *Visual Studio Add-Ins Succinctly* from Syncfusion, as well as six books on Clipper and FoxPro programming, network programming, and client/server development with Delphi. He has also written several third-party developer tools, including CLIPWKS, which allows developers to programmatically create and read native Lotus and Excel spreadsheet files from Clipper applications.

Joe has worked for a number of companies including Sperry Univac, MCI-WorldCom, Ronin, Harris Interactive, Thomas Jefferson University, People Metrics, and Investor Force. He is one of the primary authors of Results for Research (market research software), PEPSys (industrial distribution software), and a key contributor to AccuBuild (accounting software for the construction industry).

He has a background in accounting, having worked as a controller for several years in the industrial distribution field, but his real passion is computer programming.

In his spare time, Joe is an avid tennis player, practices yoga and martial arts, and plays with his first granddaughter, Blaire.

Chapter 1 Introduction

W3.CSS is a free, no-license CSS framework from w3schools.com that you can use to produce responsive websites that work across all common browsers (Chrome, Edge, Firefox, Internet Explorer) and devices (desktops, tablets, mobile). It is much smaller than other frameworks and relies only on standard CSS. For example, many common frameworks (such as Bootstrap and Foundation) offer both a CSS and JavaScript component. For a comparison of uncompressed memory sizes, see Table 1.

Table 1: Approximate framework sizes

	JavaScript (min)	CSS(min)
Bootstrap	37 kb	119 kb
Foundation	86 kb	60 kb
W3.CSS	N/A	21 kb

While each framework has its own set of features and components, the W3.CSS is small and simple to learn, and is a worthwhile contender to consider when deciding on a CSS framework.

CSS only

Since W3.CSS is a CSS-only framework, there are no components, which some of the other frameworks include. This means that while you can use the framework to handle the appearance and responsive design elements of your site, there is no JavaScript. Bootstrap, Foundation, and other such frameworks are built with both CSS styling and JavaScript components. If you need scripting functionality in your site, you should either build it yourself using libraries like jQuery, or consider a larger framework.

Layers

A website consists of three layers, but only the data (or HTML) layer is required. A user could possibly disable all scripting and CSS styling, and the data should still be readable. CSS styling improves the look of the site by providing the browser with a set of rules indicating how to display HTML elements. Scripting (mostly JavaScript) adds interactive features, such as table searching and sorting, and client size form validation.

Table 2: Website layers

Layer	Content	Format
Data Layer	Text/images to display	HTML

Layer	Content	Format
Presentation layer	Style rules to overwrite browser defaults	CSS
Activity layer	Scripting to provide interactivity to the site	JavaScript, Typescript, etc.

Browsers have default rules indicating how to display HTML tags; your custom CSS or CSS frameworks (such as W3.CSS) provide overrides for these defaults. Everything in W3.CSS uses standard CSS styles, just organized in a way to provide consistent layout and responsive design (by wrapping some styles with a media/device query filter).

W3.CSS classes

All the classes in the W3.CSS framework begin with **`w3-`**, which reduces the likelihood of conflicting style names. You can combine multiple class names in a single **`class`** statement, allowing you to provide the basic CSS class and add additional features to it. For example, the following code will create a **`w3`** container, and set its background color to blue-gray.

Code Listing 1

```
<div class="w3-container w3-blue-gray">
</div>
```

You could add the class **`w3-round-large`** to add rounded corners to the **`<div>`** element.

Simple example

The following example shows a simple webpage displaying an image and some text in a container. You can resize your browser and the entire page will adjust to the new size.

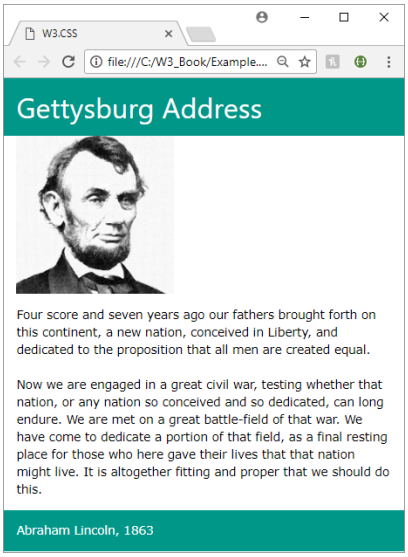

Figure 1: Sample webpage

Source page

Code Listing 1 is the source page for the above webpage. Note the link-reference to local file **w3.css** in the **<head>** section of the page. An alternative is to link to an external copy of the framework, as I'll explain shortly.

Code Listing 2: Sample W3.CSS page

```
<DOCTYPE html>
<html>
<head>
 <title>W3.CSS</title>
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <link rel="stylesheet" href="w3.css">
</head>
<body>
 <div class="w3-container w3-teal">
  <h1>Gettysburg Address</h1>
  </div>
 <div class="w3-container">
  <p><img src="abraham-lincoln.jpg" style="width:50%"></p>
  </div>
 <div class="w3-container w3-round-large">
  <p>Four score and seven years ago our fathers brought forth on this
continent, a new nation, conceived in liberty, and dedicated to the
proposition that all men are created equal.
 <br/><br/>
Now we are engaged in a great civil war, testing whether that nation, or
any nation so conceived and so dedicated, can long endure. We are met on
a great battle-field of that war. We have come to dedicate a portion of
```

```
that field as a final resting place for those who here gave their lives
so that that nation might live. It is altogether fitting and proper that
we should do this. </p>
  </div>
 <div class="w3-container w3-teal">
  <p>Abraham Lincoln 1863</p>
  </div>

</body>
</html>
```

Obtaining W3.CSS

You can download W3.CSS by following this link. It is totally free and does not require any license to use.

You can also link to the W3.CSS library via an external link, using the following snippet in your **<head>** section.

```
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
```

The code in this book is based on W3.CSS version 4. Version 4 was released in March of 2017, and is significantly faster than version 3.

Chapter 2 Layout

HTML elements on the screen all rely on the “box” model to display how the element appears. Each element is considered a box, with several styling options. Figure 2 shows the box model.

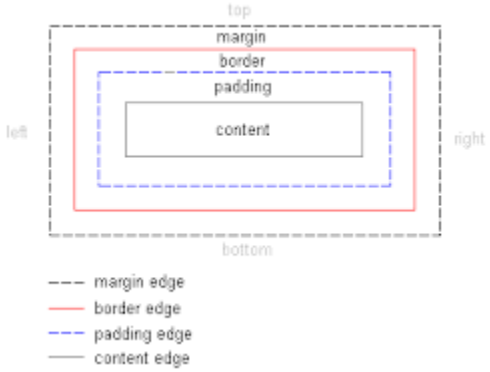

Figure 2: Box model

There are four parts to the box. The inner content is the default content shown in the element. The height and width refer to that content. The next layer is the padding, which is some number of pixels wrapped around the content. The border is the third layer, wrapped around the padding, and the final layer is the margin.

> ***Note: There is some confusion, particularly with older browsers, as to what the width and height actually represent. According to the W3C standards, the height and width should represent the content only. However, Internet Explorer calculated the width and height to include padding and border. The default behavior is the content, so a 50 x 20 pixel content with 5 pixels padding and border takes up 70 x 30 pixels on the screen.***

You can use the **`box-sizing`** CSS style rule to set the width and height to either **`content-box`** (default) or **`border-box`**. If you are going to use **`border-box`**, I would recommend setting the rule on the body tag, rather than individual elements. However, sticking with the W3C standard is your best bet. If you need to support older versions of Internet Explorer, or aren’t sure which model is used by the users’ browsers, I would add the **`box-sizing`** rule and set it to **`content-box`**.

CSS gives you total control over the margin, border, and padding; you can set them all to the same value or set different values for the individual sides (left and right), top, and bottom. One of the benefits of the W3 CSS framework is that the class definition sets them consistently for you.

CSS layout

W3 uses two basic classes to display cell content, the **w3-cell-row** class and the **w3-cell** class. The **row** element is a wrapper around a collection of cells. The cells take up 100 percent of the row width and are automatically adjusted based on their content. For example, take a look at the following code snippet.

Code Listing 3

```
  <div class="w3-cell-row">
   <div class="w3-container w3-red w3-cell">
    <p>John Smith</p>
   </div>
   <div class="w3-container w3-green w3-cell">
    <p>Lead developer and system architect on pipeline database
project</p>
   </div>
  </div>
```

This would produce the following appearance in the browser window. The two cells take up 100 percent of the width, and the larger content would take up more space.

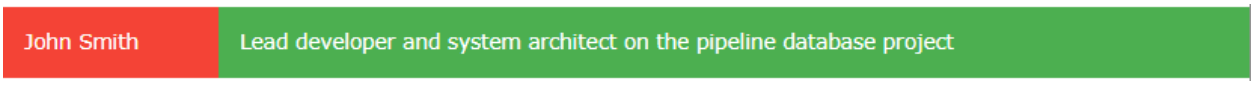

Figure 3: Row/cell appearance

One feature of the layout class is that the cell heights will match the height of the largest content, so if the cell's description had much more detail, it might appear as shown in Figure 4.

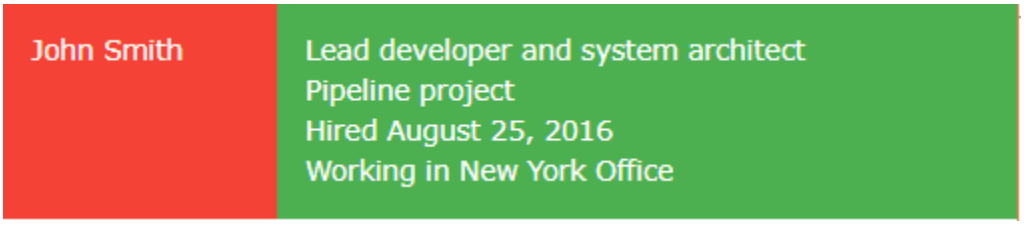

Figure 4: Same-height cells

You can adjust the alignment of the text within the cell using the **w3-cell-top**, **middle**, or **bottom** classes. For example, if we added the **w3-cell-bottom** class, the name **John Smith** would be aligned to the bottom of the red cell in the previous example.

Code Listing 4

```
<div class="w3-container w3-red w3-cell w3-cell-bottom">
 <p>John Smith</p>
</div>
```

The **w3-mobile** class can be added to a **w3-cell** as well. This will cause the cells to be laid out horizontally on a larger screen, but be stacked vertically on smaller screens, such as mobile devices (phones and some tablets).

Classes summary

The following is a list of the various **w3** classes for controlling your screen layout:

- **w3-cell-row**: Starts a container row for a collection of cells.
- **w3-cell**: Individual cell content within the row.
- **w3-cell-top**: Aligns content to top of the cell.
- **w3-cell-middle**: Centers content vertically within the cell.
- **w3-cell-bottom**: Aligns content to bottom of the cell.
- **w3-mobile**: Helper class added to cell; will display cell horizontally if there's room, but stack vertically on smaller devices.

CSS responsive

The W3.CSS framework includes a grid system designed to be responsive to the screen size, rearranging columns as needed to render on different devices. For example, let's look at a website that displays information about U.S. coins. A sample is shown in Figure 5.

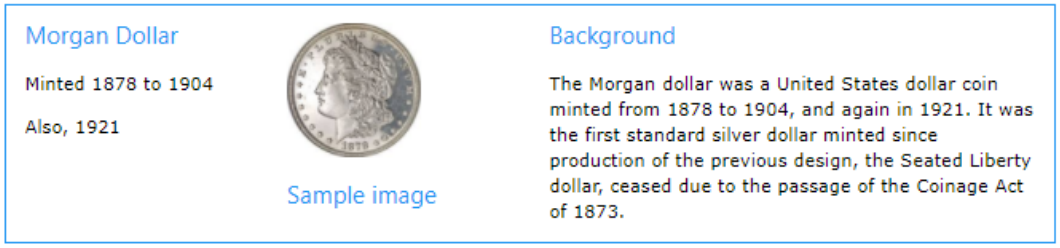

Figure 5: Responsive website example

When this screen is displayed on a mobile device, the layout will change to stack the information vertically, as shown in Figure 6.

Figure 6: Mobile device display

Responsive layout

To create the responsive layout, you must start with a **w3-row** or **w3-row-padding** container.

Code Listing 5

```
<div class="w3-row w3-border w3-border-blue w3-margin">
</div>
```

To be responsive, the elements must be nested with the **w3-row** class. Within the **w3-row div**, you can add additional **div** containers to create columns.

General-width columns

Here are the general column classes:

- **w3-half**: Takes up half of the row width.
- **w3-third**: Takes up one-third of the row width.
- **w3-twothird**: Takes up two-thirds of the row width.
- **w3-quarter**: Takes up one-fourth of the row width.
- **w3-threequarter**: Takes up three-fourths of the row width.

The coin website code snippet is shown in Code Listing 6. The first two columns (coin name and image) each take up one-fourth of the screen, while the background column takes up half of the row width.

Code Listing 6: Coin website

```
<div class="w3-row w3-border w3-border-blue w3-margin">
  <div class="w3-container w3-quarter">
  <h4 class="w3-text-blue">Morgan Dollar</h4>
```

```
    <p>Minted 1878 to 1904</p>
    <p>Also 1921</p>
  </div>
  <div class="w3-container w3-quarter w3-center">
  <p><img src="morgan.jpg" style="width:60%"></p>
  <h4 class="w3-text-blue">Sample image</h4>
  </div>
  <div class="w3-container w3-half">
  <h4 class="w3-text-blue">Background</h4>
    <p>The Morgan dollar was a United States dollar coin minted from 1878
to 1904, and again in 1921. It was the first standard silver dollar
minted since production of the previous design, the Seated Liberty
dollar, ceased due to the passage of the Coinage Act of 1873.</p>
  </div>
</div>
```

12-column responsive

In addition to the general column widths, you can use the **w3-col** class to create columns based on a 12-column grid. To do so, you need to add the **w3-col** class, followed by the number of columns (based on screen size). The size classes are:

- **l1** – **l12**: Number of columns on large screens (> 992 pixels).
- **m1** – **m12**: Number of columns on medium screens.
- **s1** – **s12**: Number of columns on small screens (< 601 pixels).

For example, the classes **w3-col**, **s6**, **m4**, and **l3** indicate to use half the screen width on a small device, one-third of the screen on a medium device, and one-fourth of the screen on a large device.

You can also set the width using the class **w3-col** and an inline style to set the **width** attribute. There is also a **w3-rest** class that indicates the column should take the remaining row width.

Show and hide columns

By default, all columns are visible on every device. However, that might not be the best approach for your website. For our example, we might want to skip the background column when viewing the site on a mobile device. For a restaurant site, you might consider the directions column critical when displaying the site on a mobile device, but hide it when viewing the site on a desktop device.

We can do this by adding the appropriate class (**w3-hide-small**) from the following list to the column we want to hide:

- **w3-hide-small**: Don't show column on a small device.
- **w3-hide-medium**: Don't show column on a medium-sized device.
- **w3-hide-large**: Don't show column on a large device.

When we add **w3-hide-small** to the background column, the screen will appear as shown in Figure 7 when viewed on a mobile device.

Code Listing 7

```
<div class="w3-container w3-half w3-hide-small">
```

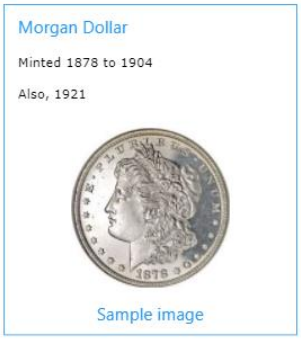

Figure 7: Mobile display without background column

CSS grid

The framework's grid system works just as the responsive system (meaning all columns must be contained with a **w3-row**), but relies on **w3-col** and **sizing** (based on 12 columns) to determine the column sizes.

Column widths

Each column is 8.33 percent (one-twelfth) of the container width, and you can specify the number of columns to display for device sizes. If nothing is specified for a particular device size, it is assumed to be 12 columns:

- **s1** – **s12**: Small (< 601 pixels)
- **m1** – **m12**: Medium
- **l1** – **l12**: Large (> 992 pixels)

Suppose we change our coin example to the following code snippet.

Code Listing 8

```
<div class="w3-row w3-border w3-border-blue w3-margin">
  <div class="w3-container w3-col s3 m2 l1">
  <h4 class="w3-text-blue">Morgan Dollar</h4>
    <p>Minted 1878 to 1904</p>
    <p>Also 1921</p>
  </div>
  <div class="w3-container w3-col s3 m2 l1 w3-center">
```

```
  <p><img src="morgan.jpg" style="width:60%"></p>
  <h4 class="w3-text-blue">Sample image</h4>
  </div>
  <div class="w3-container w3-col s6 m8 l10">
  <h4 class="w3-text-blue">Background</h4>
   <p>The Morgan dollar ... </p>
  </div>
</div>
```

You can see the column widths changing with the device size.

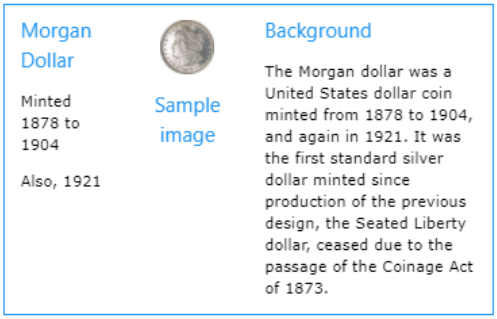

Figure 8: Small size (< 601 pixels)

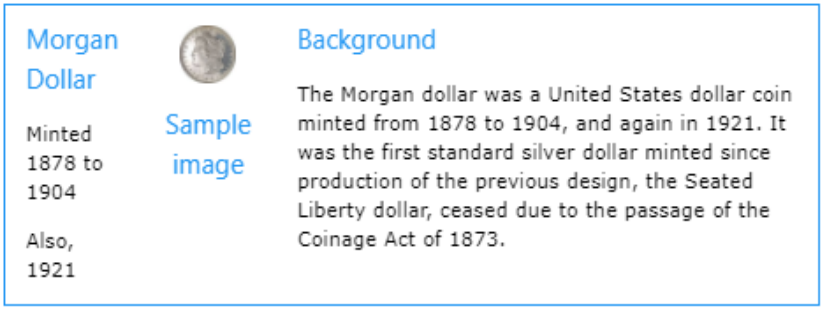

Figure 9: Medium size

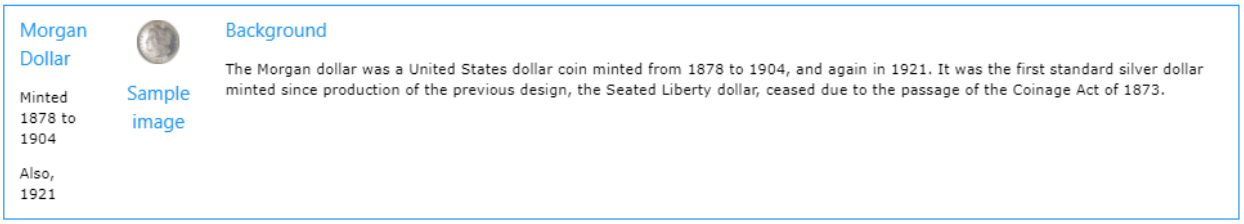

Figure 10: Large size (> 992 pixels)

By setting the **`w3-col`** and appropriate sizing in the grid, you can easily have your website adapt its appearance to the device size.

Note that you can also use the **w3-hide-size** classes to hide columns on different devices; however, be sure to adjust your size classes to avoid creating empty gaps. In our example code, if we wanted to hide the description column on a small device, the previous two columns should use **s6** sizing, rather than **s3.**

CSS display

The **w3-display-container** class allows you to display HTML elements at specific locations within other elements. For example, you might want to design a box where the close icon is always in the upper-right corner, while the Save and Cancel options always appear in the lower right. By using the **w3-display-container** class, you can position elements using class names that represent relative positions with the container.

Figure 11 shows where the class names place the elements within the container.

Figure 11: CSS display

For example, if we want to create a Save dialog box, with the top-right corner containing an X symbol to close the box, and the bottom middle the Save and Cancel buttons, we could use the following code snippet.

Code Listing 9

```
<div class="w3-display-container w3-margin w3-blue" style="height:250px">
 <div class="w3-display-topright w3-xxlarge">&times;</div>
  <div class="w3-display-bottommiddle">
   <button class="w3-btn w3-green w3-margin w3-border">Save</button>
   <button class="w3-btn w3-red w3-margin w3-border">Cancel</button>
  </div>
 </div>
</div>
```

You can still use other classes, such as **w3-hide-size**, to customize the appearance of a display container to different device sizes.

Summary

The W3.CSS framework provides a number of different layout options, allowing a lot of flexibility in designing your web application.

Chapter 3 Colors

The W3.CSS framework simplifies color management by providing classes to set the foreground and background colors of elements. The default color scheme is based on the Material Design colors. The color classes **`w3-color`** and **`w3-text-color`** can be applied to any element. There are hover classes, **`w3-hover-color`** and **`w3-hover-text-color`**, which can be applied when a user moves the mouse over an element.

Basic CSS colors

The basic color set available in the W3 framework is shown in the following figure.

Figure 12: W3.CSS colors

You can apply colors simply by adding the class name to the list of classes on the element. For example, the following container will have white lettering on a blue background.

```
<div class="w3-container w3-blue w3-text-white">
</div>
```

You can apply colors to any HTML element, such as ** ** or **<table>**.

Note: Colors should be used carefully when designing a site—in particular, check for 508 compliance. The Web Content Accessibility Guidelines require a 4.5-to-1 contrast ratio between colors, to increase readability. For larger text sizes, 14 pixels and bold, or 18 pixels and larger, a 3.0-to-1 ratio is required. You should use tools such as this website to check that color combinations meet these guidelines. People with low vision or color-blindness will appreciate the effort.

Color libraries

You can also download separate color libraries from the W3.CSS site. Table 3 lists the libraries that are available.

Table 3: Color libraries

Library Name	Description	Download name
Color Flat UI	Bold, bright colors for simple interfaces.	w3-colors-flat.css
Window Metro UI	Metro modern colors.	w3-colors-metro.css
Windows 8	Flat design and modern colors.	w3-colors-win8.css
Highway Colors	Highway colors based on federal standards.	w3-colors-highway.css
Safety Colors	Safety colors based on federal standards.	w3-colors-safety.css
European colors	European traffic colors, RA Color Standard.	w3-colors-signal.css
Fashion colors	Fashion colors from Color Trends.	w3-colors-2017.css
Vivid colors	Vibrant colors from ISCC-NBS standard.	w3-colors-vivid.css
Food colors	Colors match foods, such as wine, tomatoes.	w3-colors-food.css
Camouflage colors	Colors with earth tones, from federal standards.	w3-colors-camo.css

Note that the colors in the libraries all have unique names, so you can use multiple color libraries, in addition to the standard material design colors in the framework. Each color will begin with the final portion of the name, such as **_flat** or **_signal**.

Color themes

Color themes are CSS classes that allow you to use variations of the same basic color. For example, if your site has a gray look about it, you can download the **w3-theme-grey.css** style sheet and include it in your style sheet references.

This will add 12 additional color classes for text, default theme color, and lighter/darker variations on theme.

Table 4: Color themes

Style name	Description
w3-text-theme	Text color, generally against white background.
w3-theme	Default theme color.
w3-theme-l5	Lightest variation of the theme background color (also w3-theme-light) .
w3-theme-lX	Where x is 4 down to 1, progressively darker backgrounds, but still lighter than the default theme color.
w3-theme-d5	Darkest variation of the theme background color (also w3-theme-dark).
w3-theme-dX	Where x is 4 down to 1, progressively lighter theme backgrounds, but still darker than the default theme color.

The following figure shows a sample theme (blue-gray) to illustrate the appearance of themes.

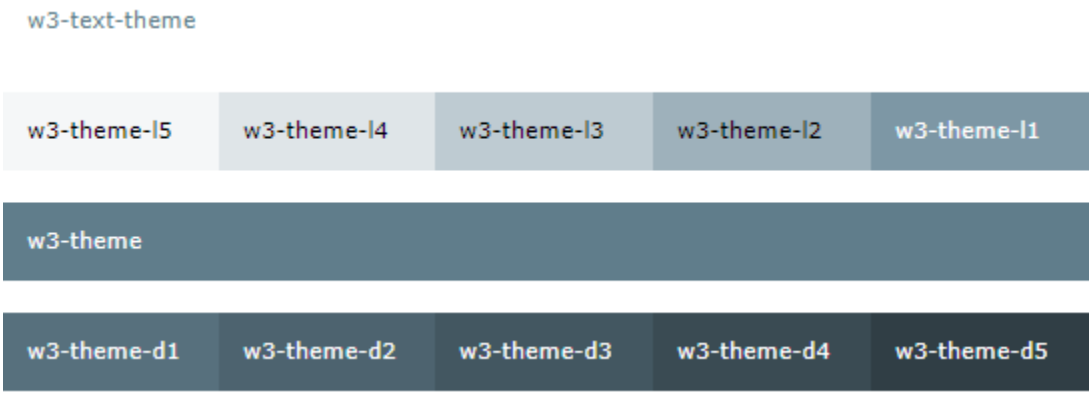

Figure 13: Sample blue-gray theme

You can use the theme color names and change the website color by simply specifying a different CSS theme style sheet. Note that many of the theme colors will not have sufficient contrast to meet accessibility guidelines. Be aware of the people using your site to ensure that a theme doesn't make the site more difficult for some users.

Available themes

Table 5 lists the predefined themes available from W3.CSS.

Table 5: W3.CSS themes

Theme name	Download name	Color
Amber	w3-theme-amber.css	**Text color**
Black	w3-theme-black.css	Text color
Blue	w3-theme-blue.css	Text color
Blue-grey	w3-theme-blue-grey.css	**Text color**
Brown	w3-theme-brown.css	**Text color**
Cyan	w3-theme-cyan.css	**Text color**
Dark Grey	w3-theme-dark-grey.css	**Text color**
Deep Orange	w3-theme-deep-orange.css	**Text color**
Deep Purple	w3-theme-deep-purple.css	**Text color**
Green	w3-theme-green.css	**Text color**
Grey	w3-theme-grey.css	**Text color**
Indigo	w3-theme-indigo.css	**Text color**
Khaki	w3-theme-khaki.css	**Text color**
Light Blue	w3-theme-light-blue.css	**Text color**
Light Green	w3-theme-light-green.css	**Text color**
Lime	w3-theme-lime.css	**Text color**
Orange	w3-theme-orange.css	**Text color**
Pink	w3-theme-pink.css	**Text color**
Purple	w3-theme-purple.css	**Text color**

Red	w3-theme-red.css	**Text color**
Teal	w3-theme-teal.css	**Text color**
Yellow	w3-theme-yellow.css	**Text color**

Chapter 4 Helper Classes

There are a few classes included that can be attached to various elements to perform some basic formatting of the element's appearance.

CSS round

By default, HTML elements appear as a square box. However, CSS allows you to set the **`border- radius`** property to create boxes with rounded corners. The W3.CSS framework provides several classes to easily create rounded corners.

Figure 14: Square box and rounded box

Table 6 lists the rounding classes. The larger the pixel size of the border radius, the more pronounced the rounded corners will be.

Table 6: Rounding classes

Class name	Border radius size
w3-round	4 pixels
w3-round-small	2 pixels
w3-round-medium	4 pixels (same as round)
w3-round-large	8 pixels
w3-round-xlarge	16 pixels
w3-round-xxlarge	32 pixels

Circle class

In addition to the rounded corners, the framework includes a **`w3-circle`** class to put the entire element in a circle (or oval), depending on the element size. You can nest circles within each other by adjusting the elements width using an inline style attribute.

CSS padding

The padding classes can be used to add a padding around the cell content. There are two types of padding classes. The **numeric** padding classes add top and bottom padding as specified by the class name. The **size** padding classes provide complete padding (all four sides) using size abbreviations.

Numeric padding

The numeric padding classes are:

- **`w3-padding-16`**: Adds 16 pixels top and bottom padding.
- **`w3-padding-24`**: Adds 24 pixels top and bottom padding.
- **`w3-padding-32`**: Adds 32 pixels top and bottom padding.
- **`w3-padding-48`**: Adds 48 pixels top and bottom padding.
- **`w3-padding-64`**: Adds 64 pixels top and bottom padding.

Size padding

The size padding classes add padding to all four sides: top, bottom, left, and right. The classes are:

- **`w3-padding`**: Adds 8 pixels top and bottom, and 16 pixels left and right (default).
- **`w3-padding-small`**: Adds 4 pixels top and bottom, and 8 pixels left and right.
- **`w3-padding-large`**: Adds 12 pixels top and bottom padding, 24 pixels left and right.

CSS margins

The margin classes allow you to add a 16-pixel margin to an element. The class allows margins to be added to all sides or to individual sides:

- **`w3-margin`**: Adds a 16-pixel margin to all sides.
- **`w3-margin-top`**: Adds a 16-pixel margin to top.
- **`w3-margin-right`**: Adds a 16-pixel margin to right side.
- **`w3-margin-bottom`**: Adds a 16-pixel margin to bottom.
- **`w3-margin-left`**: Adds a 16-pixel margin to left side.
- **`w3-section`**: Adds a 16-pixel margin to top and bottom.

You can combine classes if you only need margins on some edges. The **`w3-section`** class is a shorthand class, the same as adding classes **`w3-margin-top`** and **`w3-margin-bottom`** to the element.

CSS borders

The W3.CSS framework provides classes to place borders around elements. The are several border classes available.

Basic borders

The basic border classes allow you to add a border around the entire element or one or more sides. The classes are:

- **`w3-border`**: Adds borders to all sides (top, bottom, left, right) of the element.
- **`w3-border-top`**: Adds the border to the top of the element.
- **`w3-border-right`**: Adds the border to the right of the element.
- **`w3-border-bottom`**: Adds the border to the bottom of the element.
- **`w3-border-left`**: Adds the border to the left of the element.
- **`w3-border-0`**: Removes all borders from the element.

Border colors

You can add the **`w3-border-color`** classes to the border class (both classes are required) to change the border color. There are two class types for border colors:

- **`w3-border-color`**: Sets border to specified color (default is black).
- **`w3-hover-border-color`**: Sets border color to use when hovering over the element.

Thick borders

Thicker borders (6 pixels) can provide a nice visual touch to an element, as shown in Figure 15.

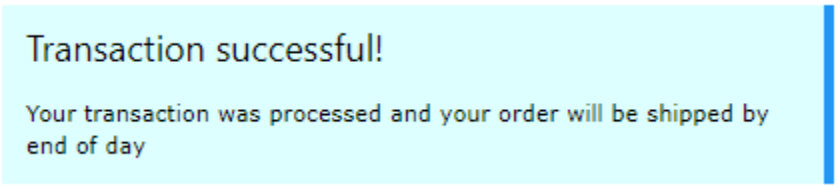

Figure 15: Thick border example

The following CSS classes (which can be combined with border colors) can be used to create thick borders around an element:

- **`w3-bottombar`**: Adds a thick border to the bottom of the element.
- **`w3-leftbar`**: Adds a thick border to the left side of the element.
- **`w3-rightbar`**: Adds a thick border to the right side of the element.
- **`w3-topbar`**: Adds a thick border to the top of the element.

CSS sizing

The W3.CSS framework provides a number of classes to change the size of elements:

- w3-tiny: Very small font, would make lawyers happy (10 pixels).
- w3-small: Smaller font (12 pixels).
- w3-medium: Default font of 15 pixels.
- w3-large: Font size of 18 pixels.
- w3-xlarge: Font size of 24 pixels.
- w3-xxlarge: Font size of 36 pixels.
- w3-xxxlarge: Font size of 48 pixels.
- w3-jumbo: Font size of 64 pixels.

These classes can be applied to any element, such as text, buttons, badges, and tables.

Summary

The helper classes and color classes, combined with the other base classes, provide a tremendous amount of control over your webpage elements.

Chapter 5 Containers

The W3.CSS framework provides a variety of container styles to use in your application. Working with these containers will require some JavaScript; however, it is minimal and simple scripting.

CSS accordions

An accordion is used when you have several text box elements, typically a header bar and some content, but don't want all the content exposed at once. The user will click on the header bar, and the text within that section will be shown (or hidden).

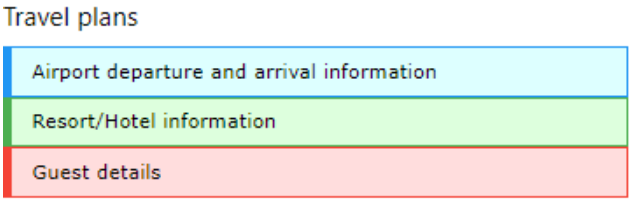

Figure 16: CSS accordion example

Accordion bars

The accordion bars are styled buttons, as shown in the following snippet.

Code Listing 10

```
<button onclick="openPanel('Airport');"
 class="w3-btn w3-block w3-pale-blue w3-border-blue
    w3-border w3-leftbar w3-left-align">
  Airport departure and arrival information
 </button>
```

The **`openPanel()`** JavaScript function will handle the opening and closing of the accordion content.

Building the sections

To create the section for the accordion, you will need to create a separate **`<div>`** for each section. The following code snippet shows a sample content **`<div>`**.

Code Listing 11

```
<div id="Airport" class="w3-container w3-hide">
 <p>Airports</p>
</div>
```

Each ** ** will have the **w3-hide** class applied so that the content is not visible. The JavaScript code will take care of displaying the content.

Navigation

To navigate between the accordions, you will need to employ some simple JavaScript. The code will have two tasks: one is to hide all of the inactive tabs (based on the class name), and the second is to make the selected tab visible.

Code Listing 12

```
function openPanel(id) {
 var x = document.getElementById(id);
  if (x.className.indexOf('w3-show') == -1) {
  x.className += ' w3-show'
 } else {
  x.className = x.className.replace(' w3-show','');
 }
}
```

When the user clicks on the accordion button, the **openPanel()** function checks to see if the **w3-show** class is on the current item. If it is not present, it is added and the content becomes visible. If the **w3-show** class is already there, it is removed (which causes the original **w3-hide** class to take effect).

CSS tabs

Tabs function in a similar manner to accordions, where only one section of content is visible at a time. However, the navigation among sections is done via a series of buttons (tabs), typically across the top of the page.

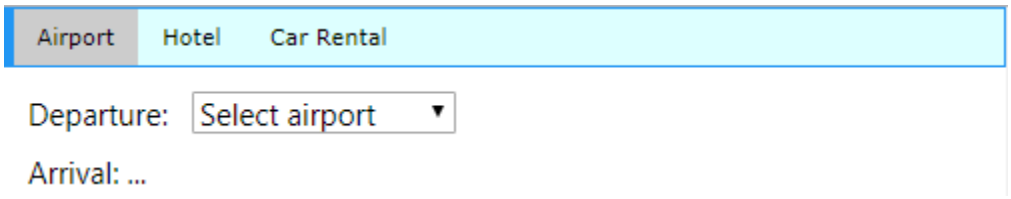

Figure 17: CSS tabs example

Tab classes

There are two classes you need to work with in tabs:

- **w3-bar**: The container to hold the tab bar across the top.
- **w3-bar-item**: The button representing the tab.

Building the tabs

Code Listing 13

```
<div class="w3-bar w3-pale-blue w3-leftbar w3-border-blue w3-border">
 <button onclick="openDiv('Airport');"
  class="w3-bar-item w3-button">Airport
 </button>
 <button onclick="openDiv('Hotel');"
  class="w3-bar-item w3-button">Hotel
 </button>
 <button onclick="openDiv('Rental');"
  class="w3-bar-item w3-button">Rental
 </button>
</div>
```

Building the sections

To create sections for the tabs, you will need to create a separate ** ** for each section within the tabs. The following code snippet shows a sample group of ** ** elements. You will need to define a class name for each of the tab sections—we used **travel** in our example code. The first ** ** should be displayed, while the remaining ** ** elements are hidden.

Code Listing 14

```
<div id="Airport class="travel">
 <p>Airport details</p>
</div>
<div id="Hotel class="travel" style="display:none;">
 <p>Hotel details</p>
</div>
<div id="Rental class="travel" style="display:none;">
 <p>Rental details</p>
</div>
```

Navigation

To navigate between the tabs, you will need to employ some simple JavaScript. The code will have two purposes: the first is to hide all of the inactive tabs (based on the class name), and the second is to make the selected tab visible.

Code Listing 15

```
function openDiv(divName) {
 var x = document.getElementsByClassName('travel');
  for (var i =0; i < x.length; i++) {
  x[i].style.display = 'none';
 }
 document.getElementById(divName).style.display = 'block';
}
```

When the user clicks on the tab, the JavaScript function is called and passed the **`divName`** to open.

Vertical tabs

Although tabs are typically horizontal, you can easily create vertical tabs with a few small changes to the layout.

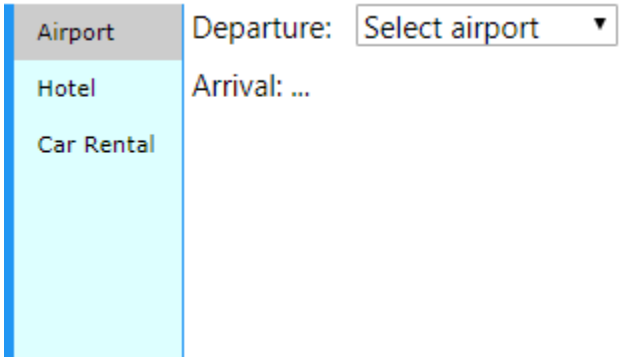

Figure 18: Vertical tabs example

You need to change the **`w3-bar`** class to **`w3-bar-block`** and add the **`w3-sidebar`** class. In addition, you'll need to set the width of the tab menu.

Code Listing 16

```
<div class="w3-sidebar w3-bar-block w3-pale-blue
    w3-border-blue w3-border" style="width:120px;">
```

You will also need to add **`margin-left`** style to the sections, as follows.

Code Listing 17

```
<div id="Airport" class="travel" style="margin-left:125px;">
```

Be sure to set the margin large enough to start past the width of the menu. You should add the **`margin-left`** style in the travel class, rather than inline.

CSS cards

A card is a container with a shadow around it to provide a paper-like appearance to an element. The framework provides a simple **`w3-card`** class to create a card element. The **`w3-card-4`** is similar, except that the shadow around the card is larger (4 pixels rather than 2 pixels). By default, the card will be white, although you can add any **`w3-color`** class to create a colored card.

For example, the following code snippet would display the card shown in Figure 19 on a webpage.

Code Listing 18

```
<div class="w3-card-4 w3-cyan w3-padding w3-margin">
 <p class="w3-text-white w3-large">Danielle Salsbury</p>
 <p>Tennis player extraordinaire</p>
</div>
```

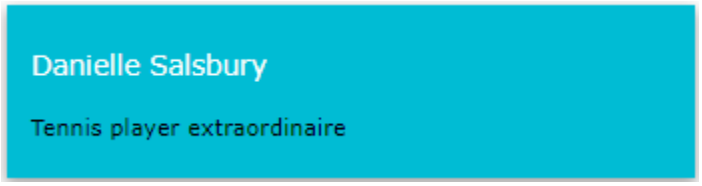

Figure 19 : CSS card sample

CSS sidebar

A common website container is the sidebar menu, where the navigation options appear on the side of the screen, and the content appears in a larger area off to the other side. It is constructed very similarly to vertical tabs, but has additional options available to increase its flexibility.

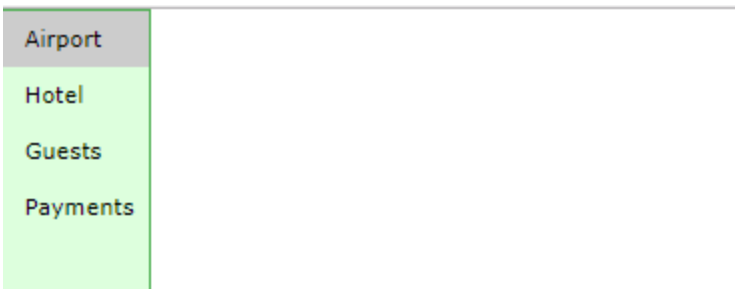

Figure 20: Basic sidebar

Basic sidebar

To create the sidebar, you need to use the **`w3-sidebar`** and **`w3-bar-block`** classes, as well as set the width of the sidebar. Background colors and borders are optional, but generally the sidebar should be distinguished from the rest of the screen. The following code snippet shows the basic sidebar setup.

Code Listing 19

```
<div class="w3-sidebar w3-bar-block w3-border">
 <a href="#" class="w3-bar-item w3-button">Airport</a>
 <a href="#" class="w3-bar-item w3-button">Hotel</a>
 ...
</div>
```

You will also need to set the left margin of the content elements large enough to be positioned past the right border of the sidebar.

Collapsible sidebar

A fairly common technique on websites, particularly mobile sites, is hiding the sidebar menu (providing more space for the main content), and only opening the menu when the user clicks an icon (typically the “hamburger” icon with three horizontal bars). With a little JavaScript, you can have the W3.CSS sidebar provide this behavior.

Creating the icon

You will need to add a header (or some container to show the menu icon). Be sure to provide an **`id`** for the menu icon (**`openNav`** in our example). The **`#9776`** is the HTML entity code for the hamburger menu icon. The following code snippet shows the header and icon example code.

Code Listing 20

```
<div class="w3-container w3-pale-green">
 <h4>
 <span id="openNav" class="w3-small w3-button"
```

```
   onclick="openMenu();">&#9776;
  </span>
  My travel plans</a>
 </h4>
</div>
```

When the user clicks on the icon, the **openMenu()** JavaScript function is called.

Creating the sidebar

The sidebar itself is created using the same code as the basic sidebar code, with a couple changes. First, the sidebar menu will need an **id**, since the JavaScript code will need to reference it. In addition, it initially will not be visible on the screen.

The other change is a new menu item, to call the JavaScript **closeMenu()** function to hide the sidebar when clicked.

Code Listing 21

```
<div class="w3-sidebar w3-bar-block w3-border" id="sideBAR"
   style="width:20%;display:none;">
   <button class="w3-bar-item w3-button"
     onclick="closeMenu();">&times;
  </button>
  <a href="#" class="w3-bar-item w3-button">Airport</a>
  <a href="#" class="w3-bar-item w3-button">Hotel</a>
  ...
</div>
```

Opening and closing

The JavaScript functions to open and close the menu simply need to toggle the menu's display style, to show or hide the menu. The following code snippet shows the necessary JavaScript.

Code Listing 22

```
function openMenu() {
 document.getElementsById('sideBAR').style.display = 'block';
}
function closeMenu() {
 document.getElementsById('sideBAR').style.display = 'none';
}
```

Keep in mind that the element name for the sidebar is case sensitive, so be sure the **id** name matches the **id** you've assigned to your sidebar menu.

Slide content to right

While the collapsible sidebar appears properly, there might be times when you want the content to shift to the right, rather than be overwritten by the sidebar menu. Fortunately, this is an easy change to the JavaScript function needed for the collapsible sidebar menu. The following JavaScript code will open the menu and adjust the screen content.

Code Listing 23

```
function openandMoveMenu() {
 document.getElementsById('main').style.marginLeft = '21%';
 document.getElementsById('sideBAR').style.width = '20%';
 document.getElementsById('sideBAR').style.display = 'block';
 document.getElementsById('openNav').style.display = 'none';
}
function closeMenu() {
 document.getElementsById('main').style.marginLeft = '1%';
 document.getElementsById('sideBAR').style.display = 'none';
 document.getElementsById('openNav').style.display = 'inline-block';
}
```

You will need to adjust the margin and width percentages to match your sidebar menu width, and adjust the left margin of the content of the main portion of your screen.

CSS panels

The **`w3-panel`** class is very similar to the **`w3-container`** class, but includes a 16-pixel top and bottom margin to have the panel stand out a bit from the background. This makes the panel a good choice for things like notes and alerts.

The following code snippet illustrates a panel to display a warning message.

Code Listing 24

```
<div class="w3-panel w3-red w3-round-xxlarge w3-margin">
 <h3 class="w3-text-yellow"><b>Warning</b></h3>
 <p>This action will delete hundreds of records and will be recorded in
the audit trail</p>
</div>
```

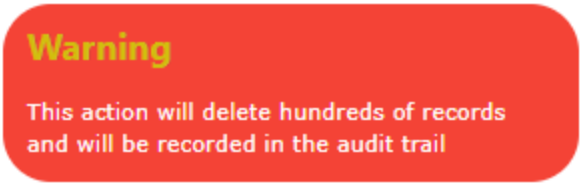

Figure 21: Warning panel

Summary

The W3.CSS framework provides a number of ways to present content and actions, and with minor JavaScript, you can have an easy-to-work system to provide the user with navigation between menus and content on your site.

Chapter 6 Visual Elements

W3.CSS includes a few classes for creating visual elements on the page. All these elements use the **w3-panel** as their base class.

CSS notes

Notes are boxes of information displayed on the website, and can be easily constructed using the **w3-panel** base and some helper classes. For example, the following code snippet would produce a bordered note container.

Code Listing 25

```
<div class="w3-container">
 <div class="w3-panel w3-border w3-pale-red w3-round-large">
 <p>All credit card payments MUST be verified prior to shipping!</p>
 </div>
</div>
```

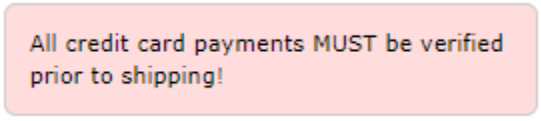

Figure 22 : CSS note example

Customizing the notes

In addition to colors and rounded corners, you can use the following classes to improve the appearance of notes:

- **w3-border-color**: Defines the color for border elements.
- **w3-leftbar**: Adds a vertical bar on left side of the container.
- **w3-rightbar**: Adds a vertical bar on right side of the container.
- **w3-topbar**: Adds a horizontal bar on the top of the container.
- **w3-bottombar**: Adds a horizontal bar on the container bottom.

Figure 23 shows the previous note with a **w3-border-red** and **w3-leftbar** class added.

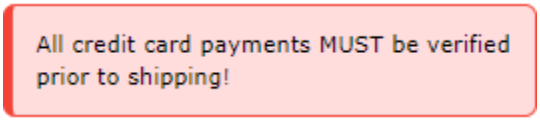

Figure 23: CSS custom note appearance

CSS alerts

Alerts are generally messages on the website that need attention from the user. Typically, the color of the alert suggests the severity of the issue. A red alert indicates a risky or negative situation, while green is generally positive, a confirmation that something worked.

The general structure of an alert is as follows.

Code Listing 26

```
<div class="w3-display-container w3-panel w3-pale-yellow w3-margin">
 <span onclick="this.parentElement.style.display='none'"
   class="w3-button w3-pale-yellow w3-large
      w3-display-topright">&times;</span>
 <h3>Confirm</h3>
 <p>Please confirm address prior to shipping!</p>
</div>
```

In this example, we are using the **`w3-display-container`** class to place an X in the upper corner. Clicking the X will dismiss the alert. We are using the **`&times`** HTML entity rather than the letter X for the closing button.

The **`onclick`** code simply changes the alert's display style property to **`none`**, causing the alert to disappear from the screen.

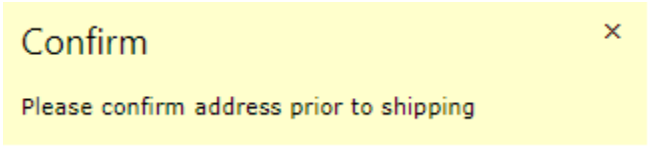

Figure 24: Sample alert panel

Generally, the following colors could be used for alert messages:

- Red or pale red: Error condition, needs attention.
- Yellow or pale yellow: Warning, might need attention.
- Green, pale green: Success!
- Blue, cyan: Information only.

Note: Colors are not universal, and have different meanings in different cultures. For example, the color red in western cultures generally means danger, while in China, the color red represents luck. Be sure to consider your audience when choosing colors for alerts and boxes.

Chapter 7 Text

The default font family in the W3.CSS framework is **Verdana, sans-serif**, with a font size of 15 pixels. It also has a default line spacing of 1.5, which produces webpages that are easy to read for most users. The font is set on the HTML and body tags, which means it will be used for all elements on the site, unless specifically changed.

The H1 through H6 header sizes use the font family of **Segoe UI**, **Arial**, **sans-serif**.

Note: A serif font (such as Times Roman or Georgia) has little decorative strokes on the end of the letters, while a sans serif font (such as Helvetica or Arial) does not have these strokes. The font family tells the browser to try each font in the list, and if not found, just choose any font that most closely matches the last item in the family. W3.CSS suggests Verdana, and if not found, asks the browser to use a sans-serif font.

CSS fonts

You could easily override the Verdana font by adding a font family to your HTML and body elements after the W3.CSS file is loaded, for example.

Code Listing 27

```
<link rel="stylesheet" href="css/w3.css">
<link rel="stylesheet" href="clientStyle.css">
```

Where the file **clientStyle.css** contains a line similar to the following.

Code Listing 28

```
html, body {
 font-family: "Georgia", sans-serif;
}
```

You could modify the W3.CSS file as well, but I would advise you to put your unique settings into a separate style sheet, rather than risk losing them if you download a W3.CSS update.

Print style

Verdana is a good, flexible font, but you might want to choose a different font for printing. The following code snippet shows a media query to change the printed font to an alternative font.

Code Listing 29

```
@media print {
 html, body {
  font-family: Helvetica, sans-serif;
 }
 h1, h2, h3, h4, h5, h6 {
  font-family: "Century Gothic", Helvetica, sans-serif;
 }
}
```

Web fonts

When a browser chooses a font, it relies on the fonts that are installed on the user's computer. However, CSS allows you to use fonts that are available at a web location, rather than the user's computer. These are referred to as *web fonts*.

A popular source of web fonts is the Google Fonts library. This site contains hundreds of web fonts you can use to improve your website's appearance.

Installing the font

To use a web font, you will need to install it into your webpage. For example, the Tangerine Google font could be added via the following code placed in the **<head>** section of the website.

Code Listing 30

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

This will add a new font called **Tangerine** to your site. Tangerine is a script font, so you could create a class to use when displaying signatures.

Code Listing 31

```
.SignatureLine {
 font-family: Tangerine, script, serif;
}
```

Be sure to include a default font in case the requested font is not loaded for any reason. Also, keep in mind that the font file needs to be loaded when the page starts, which can slow down the startup time.

Alignment

There are three classes you can use to align text (and other elements) horizontally within a container. These classes are:

- **`w3-left-align`**: Aligns text to the left.
- **`w3-right-align`**: Aligns text to the right.
- **`w3-center`**: Centers text within the container.

Text features

You can change the appearance of your text a bit by using a couple of additional classes with the framework. These classes allow you to widen the text or set the opacity. For example, the following code snippet adds 4 pixels of spacing between the characters in the text.

```
<p class="w3-wide">On sale today only!</p>
```

The **`w3-opacity`** class sets the opacity of the element to 60 percent. The opacity value determines how translucent the text is. The smaller the number, the more translucent the element appears. The following figure shows how you can use the opacity class to display an error message, but still let the user see the screen beneath the error message.

Figure 25: Opacity example

You can also use the **`w3-opacity-max`** class to set the opacity to 25 percent, and the **`w3-opacity-min`** class to set the level to 75 percent. The **`w3-opacity-off`** class will set the opacity to 100 percent (i.e. totally opaque).

Tip: Opacity can be used when you are loading content via Ajax. Set the class to `w3-opacity-max` during the `before Send` event, and set the class to `w3-opacity-off` during the `complete` event. This will provide a visual indication to the user that a portion of the screen has been updated.

CSS code

If you want to display some sort of code, such as programming code or computer instructions, you can use the **`w3-code`** class.

Code Listing 32

```
<div class="w3-code" translate="no">
 function DisplayError(msg)<br/>
 {<br/>
     alert(msg);<br/>
 }<br/>
</div>
```

Note:** `Translate="no"` **is an HTML5 attribute indicating the following content should not be translated to other languages.

The default font families for code are Consolas and Courier New.

Chapter 8 Menus

Navigation refers to the hyperlinks and references that allow a user to move among pages in a website. The W3.CSS framework provides classes to allow you to control the appearance of your menu links.

CSS navigation

The basic navigation element is an anchor tag **<a>**, with an **HREF** attribute indicating the content to load when the element is clicked. You can also use JavaScript and Ajax to load partial content, by using the **onclick** event. The syntax is as follows.

Code Listing 33

```
<a href="#"
 onclick="CallJavaScript();return false;">
 Menu text
</a>
```

The value of the **HREF** indicates the link to execute when clicked. If an **onclick** event is called, then the **HREF** will only be used if the **onclick** function returns **TRUE**. Single page applications keep the menu structure generally intact, and only update the content area by calling JavaScript to build the content window.

Basic structure

The following code snippet shows the basic menu structure, without any classes applied yet.

Code Listing 34

```
<div>
 <a href="#">Home</a>
 <a href="#">Customers</a>
 <a href="#">Orders</a>
 <a href="#">Shipping</a>
</div>
```

Running this code produces the screen shown in Figure 26, simply a row of hyperlinks.

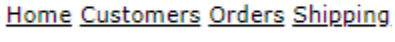

Figure 26: Menu items

Horizontal menu

Adding the **w3-bar** class, an optional color class to the ** ** tag, and the **w3-bar-item** and **w3-button** classes, produces a horizontal menu with a blue background.

Code Listing 35

```
<div class="w3-bar w3-blue">
 <a href="#" class="w3-bar-item w3-button">Home</a>
 <a href="#" class="w3-bar-item w3-button">Customers</a>
 <a href="#" class="w3-bar-item w3-button">Orders</a>
 <a href="#" class="w3-bar-item w3-button">Shipping</a>
</div>
```

The **w3-button** class displays the hyperlinks as buttons, so they are not underlined. The result is shown in Figure 27.

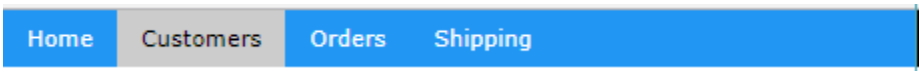

Figure 27: Horizontal menu

Other classes

You can add some helper classes to adjust the menu's appearance even more. These classes are:

- **w3-border**: Adds a border around the entire bar.
- **w3-card**: Displays the bar in a "card," with a shadow.
- **w3-round**: Creates a rounded menu effect.
- **w3-mobile**: Causes the items to stack vertically on smaller screens.
- **w3-size**: Increases the font size of the menu items.

For example, the following code snippet would produce the pale green, rounded menu bar shown in Figure 28.

Code Listing 36

```
<div class="w3-bar w3-pale-green w3-border w3-round-xxlarge w3-margin">
```

Figure 28: Rounded menu bar

Menu items

The **w3-bar-item** class is the basic class to create menu items within the bar. When you hover over a menu item, the background color will change to gray to indicate the menu item to be clicked.

Changing hover colors

Adding **w3-hover-color** class to an item allows you to overwrite the hover color from gray to another color. You can also add the **w3-hover-none** if you don't want the hover effect at all.

Code Listing 37

```
<!-- Change hover color to green. -->
<a href="#" class="w3-bar-item w3-button w3-hover-green">Customers</a>
<!-- Remove hover entirely for this item. -->
<a href="#" class="w3-bar-item w3-button w3-hover-none">Home</a>
```

Right-align items

The **w3-right** class, when added to the bar item, causes the item to be aligned to the right side of the bar. Often, the help link is moved to the right, to keep it distinct from the operational links.

Adding text and buttons to the menu bar

Since the bar is the primary navigation for the site, you might need to add text or input elements to the bar. The following screen illustrates a navigation bar showing the currently selected club and the ability to search for a team or player.

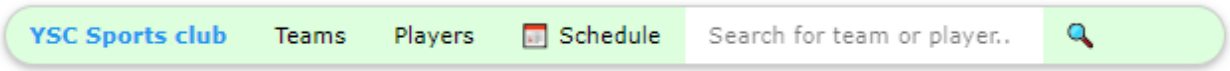

Figure 29: Soccer site navigation

The following code illustrates adding text (YSC Sports club) and a search box and button to the bar items. You do not need to limit bar items to just buttons.

Code Listing 38

```
<div class="w3-bar w3-pale-green w3-border w3-round-xxlarge w3-margin">

<span class="w3-bar-item w3-text-blue"><b>YSC Sports Club</b> </span>

  <a href="#" class="w3-bar-item w3-button">Teams</a>

  <a href="#" class="w3-bar-item w3-button">Players</a>
```

```
  <!-- HTML entity for a calendar icon -->
  <a href="#" class="w3-bar-item w3-button">&#128197;Schedule</a>
  <input type="text" class="w3-bar-item w3-input"
  placeholder="Search for team or player.." >
  <!-- HTML entity for a search icon -->
  <a href="#" class="w3-bar-item w3-button">&#128269;</a>
</div>
```

Drop-down menu items

You can also add a drop-down menu to your navigation using the **w3-dropdown-hover** and **w3-dropdown-content** classes. The following code snippet shows an example of added schedule menu options to our soccer navigation bar.

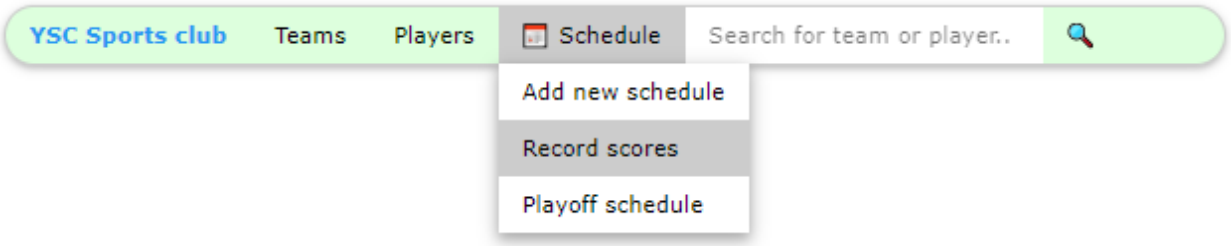

Figure 30: Drop-down menus

The following is the code for this example. The menu option is a wrapper in the **w3-dropdown-hover** class, and the actual drop-down menu is in the **w3-dropdown-content** class.

Code Listing 39

```
<div class="w3-dropdown-hover">
 <button class="w3-button">&#128197; Schedule</button>
  <div class="w3-dropdown-content w3-bar-block w3-card-4">
   <a href="#" class="w3-bar-item w3-button">Add new schedule</a>
   <a href="#" class="w3-bar-item w3-button">Record scores</a>
   <a href="#" class="w3-bar-item w3-button">Playoff schedule</a>
  </div>
</div>
```

The bar items within the drop-down menu can also have custom colors and hover colors. They are regular menu bar items, and can even be text and input text elements.

Navigation bar positions

Often, you want to keep the navigation bar positioned at the top or bottom of the page. This is accomplished simply by wrapping the entire menu bar and items with a ** ** tag using the **w3-top** or **w3-bottom** classes.

Code Listing 40

```
<div class="w3-top">
 Menu bar items
</div>
```

Vertical navigation bar

You can flip the navigation bar to a vertical orientation simply by replacing the **w3-bar** class with the **w3-bar-block** class, as follows.

Code Listing 41

```
<div class="w3-bar-block w3-pale-green w3-margin" style="width:27%;">
```

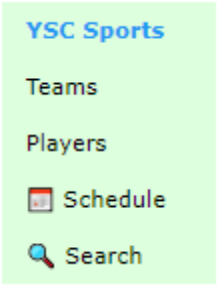

Figure 31: Vertical bar

Notice that you might need to set the width of the block, since the default is the entire width of the parent container.

Summary

Navigation among pages should be easy and intuitive, and the W3 classes make it simple to create visually appealing menu structures. Here is a summary of the classes:

- w3-bar: Creates the menu bar.
- w3-bar-block: Creates a vertical menu bar.
- w3-top: Keeps menu bar at top of page.
- w3-bottom: Keeps menu bar at bottom of page.
- w3-bar-item: Adds button, text, or input for item on the menu.
- w3-dropdown-hover: Creates a drop-down element on the menu.
- w3-dropdown-content: Menu bar items within the drop-down menu.

Chapter 9 Tables and Lists

The HTML table system is primarily designed for displaying tabular data in rows and columns. Your HTML code must create the basic table structure, but W3 provides classes to easily enhance the table's appearance.

CSS tables

A basic table consists of the **<table>** tag, followed by some number of rows of **<tr>** elements. Within the **<tr>**…**</tr>** row will be any number of **<td>**…**</td>** elements. Each row should have the same number of elements; if not, blank spaces will appear within the table structure.

Code Listing 42

```
<table>
  <tr>
  <th>Player name</th>
  <th>Team</th>
  <th>Yards</th>
  <th>TD</th>
  </tr>
  <tr>
  <td>Nick Foles</td>
  <td>Philadelphia Eagles</td>
  <td>373</td>
  <td>3</td>
  </tr>
  <tr>
  <td>Tom Brady </td>
  <td>New England Patriots </td>
  <td>505</td>
  <td>3</td>
  </tr>
</table>
```

Each of the available W3 classes is applied to the **<table>** element.

Basic table

The **w3-table** class provides the basic table structure, with a width of 100 percent. You can add this class to any table within a site.

Code Listing 43

```
<table class="w3-table">
</table>
```

Figure 32 shows the sample table.

Player name	Team	Yards	TD
Nick Foles	Philadelphia Eagles	373	3
Tom Brady	New England Patriots	505	3
Trey Burton	Philadelphia Eagles	1	1

Figure 32: Sample table

Additional classes can be added to control the table's appearance.

Striped table

The **`w3-striped`** class adds striping, so alternating rows will have a shaded effect, as shown in Figure 33.

Player name	Team	Yards	TD
Nick Foles	Philadelphia Eagles	373	3
Tom Brady	New England Patriots	505	3
Trey Burton	Philadelphia Eagles	1	1

Figure 33: Striped table

The striped rows make the table easier to read.

Borders

There are two classes for adding borders to the table. The **`w3-border`** class adds a border around the entire table. The **`w3-bordered`** class adds a bottom border to each row in the table.

w3-table-all

The **`w3-table-all`** class adds all the features (borders and shading) to the table. Figure 34 shows the class applied to our sample table.

Player name	Team	Yards	TD
Nick Foles	Philadelphia Eagles	373	3
Tom Brady	New England Patriots	505	3
Trey Burton	Philadelphia Eagles	1	1

Figure 34: All table features

Hovering

There are times when a table can be used as an edit list, so it is important to let the user know which row is currently selected. W3.CSS makes this easy to do by adding the **`w3-hoverable`** class to the table. Figure 35 shows a table with the first row selected.

Player name	Team	Yards	TD		
Nick Foles	Philadelphia Eagles	373	3	✍	✕
Tom Brady	New England Patriots	505	3	✍	✕
Trey Burton	Philadelphia Eagles	1	1	✍	✕

Figure 35: Hoverable table

We've added a new column with the HTML entity **`&#270d;`** to produce the "writing hand" Edit icon and the entity **`&#2716;`** (with the class **`w3-text-red`**) for the red Delete icon.

You can use the **`w3-hover-color`** classes to change the hover color to something other than a gray background. The hover color is set on the row level, as shown in the following code snippet.

Code Listing 44

```
<tr>
<td class="w3-hover-green">Nick Foles</td> ...
</tr>
<tr>
<td class="w3-hover-blue">Tom Brady</td> ...
</tr>
```

Hovering is a simple visual indicator of which row you are going to edit or delete.

CSS lists

HTML allows you to create lists using the **<ul>** (unordered list) or **<ol>** (ordered list) tags with a collection of **<li>** tags contained within. The basic structure is shown in the following code snippet.

Code Listing 45

```
<ul>
 <li>Soccer</li>
 <li>Football</li>
</ul>
```

The framework provides classes to style the lists.

Basic list style

The **w3-ul** tag sets the basic list structure and handles the indentation of the list elements. The following code produces the list as shown in Figure 36.

Code Listing 46

```
<ul class="w3-ul">
  <li>Soccer</li>
  <li>Football</li>
  <li>Baseball</li>
 </ul>
```

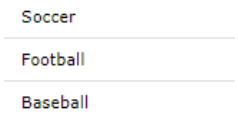

Figure 36: Basic list

Tweaking the list

You can use many of the classes to adjust the appearance of the basic list design. These include:

- **w3-border**: Places a border around the entire list box.
- **w3-card-4**: Wraps the list in a shadowed card.
- **w3-center**: Centers the elements within the list.
- **w3-color**: Provides a color to the list.

Keep in mind that the basic list class sets the list width to 100 percent of its container. If you need to make a smaller width list, you will need to use the CSS style to set the width.

Code Listing 47

```
<ul class="w3-ul w3-border w3-margin" style="width:20%;" >
```

Hoverable list

The **w3-hoverable** class can be added to the **<ul>** element to cause a gray background to be used when the mouse hovers over one of the list elements. You can change the background color by setting the **w3-hoverable-color** class on any of the list elements.

Closable list

You can add an X icon to the list items, and with a little JavaScript call, give the user the ability to remove an item from the list. For example, the following code snippet will use the **&times** HTML entity and, when clicked, will hide the list item.

Code Listing 48

```
<li class="w3-display-container">Baseball
  <span onclick="this.parentElement.style.display='none'"
  class="w3-button w3-display-right">&times;</span>
</li>
```

Adding icons to list items

List items are not limited to just text—you can add images using the **<img>** tag with the **w3-bar-item** class to show an image, rather than text. For the example, the following code snippet will add sports images next to the sport name.

Code Listing 49

```
<li><img src="soccer.jpg" class="w3-bar-item" width="32" height="32">
   Soccer</li>
<li><img src="football.jpg" class="w3-bar-item" width="32" height="32">
   Football</li>
<li><img src="baseball.jpg" class="w3-bar-item" width="32" height="32">
   Baseball</li>
```

The result will appear as shown in Figure 37.

Figure 37: List with images

Tip: When using images, you will get better performance (smaller file sizes) if you adjust the image file size, rather than using the HTML height and width attributes to resize the image.

Summary

The table and list classes make it easy to style your basic tables and lists, and are summarized as follows:

- **`w3-table`**: Basic table class.
- **`w3-striped`**: Adds stripes to alternate rows in a table.
- **`w3-border`**: Adds a border around the entire table.
- **`w3-bordered`**: Adds a border between table rows.
- **`w3-table-all`**: Adds borders and striping to a table.
- **`w3-ul`**: Basic list class.

Chapter 10 Buttons and Labels

A button element on a webpage is typically used to initiate an action or toggle a feature. The basic HTML syntax is as follows.

Code Listing 50

```
<input type="button"
  value="button text"
  href="#"
  onclick="javascript function">
```

If the **href** attribute is specified, it represents a link that the user will go to when they click the button. You can also have a JavaScript function called when the button is clicked. Note that if the function returns **FALSE**, the **href** link will not be activated.

An alternate approach for buttons is the **<button>** element or an **<a>** anchor element. You can use any approach to create a clickable area on the screen.

CSS buttons

There are two basic classes available to style the button's appearance in the framework. The **w3-button** class creates a button that will change color when you hover over it. The **w3-btn** class creates a button that will display a slight shadow effect when you hover over it. Both button classes default to a black button with gray hover effects. In addition, hovering over the button will change the cursor to a pointer, indicating that the user can click the button.

Button variations

You can use the **w3-color** attributes to change the background and hover colors of the buttons. For example, the following code snippet might be used to allow the user to save or discard updates.

Code Listing 51

```
<a href="save.aspx" class="w3-btn w3-green">Save changes</a>
<a href="discard.aspx" class="w3-btn w3-red">Discard changes</a>
```

Figure 38 shows sample colored buttons.

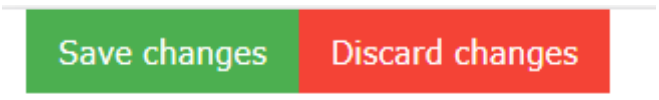

Figure 38: Sample buttons

Button colors

You can add the **w3-color** or **w3-hover-color** classes to set button colors or hover colors (the default hover color is gray).

Button shapes

The default shape of the buttons is a rectangle. You can use the **w3-round** classes to add rounded corners to the buttons. The following code snippet will produce the rounded buttons shown in Figure 39.

Code Listing 52

```
<div class="w3-container w3-margin">
 <a href="#" class="w3-btn w3-green w3-round-xxlarge">
  Save changes</a>
 <a href="#" class="w3-btn w3-red w3-round-xxlarge">
  Discard changes</a>
</div>
```

Figure 39: Rounded buttons

Disabled buttons

Buttons have a visual effect, either background color change or shadowing, as well as the cursor changing to indicate the button is clickable. If a button needs to be disabled, you can add the **w3-disabled** class to the button. This will cause the button to have a grayed-out appearance and disable the cursor when the user moves the mouse over the button.

Tip: You might want to dynamically add the `w3-disabled` class via JavaScript when the button is clicked. This could provide a visual indication that the button has been pressed and the action is being performed. Remove the class when the work is complete.

Ripple effect

A ripple effect is a visual effect that indicates the button has been clicked, providing some feedback to the user. It is considered good UX design to make sure the user knows that the button has been clicked. You can add the **w3-ripple** class to any button to provide this effect.

CSS badges

A badge is a circular label, typically containing a number, which indicates how many items are found in some collection. The W3.CSS framework has a single class, **w3-badge**, that will create a circular badge with a black background.

You can use the **w3-color** and **w3-size** classes to control the color and size of the badge element. The following code snippet creates the badges shown in Figure 40.

Code Listing 53

```
<div class="w3-container w3-margin" style="width:18%;">
 <p>w3-badge <span class="w3-badge w3-right">9</span></p>
 <p>w3-badge w3-red
   <span class="w3-badge w3-red w3-right">4</span></p>
 <p>w3-badge w3-red w3-large
   <span class="w3-badge w3-blue w3-right w3-large">2</span></p>
</div>
```

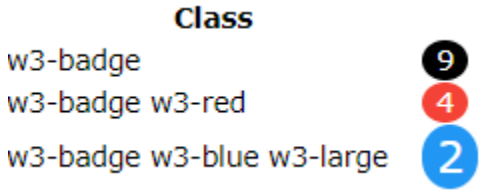

Figure 40: Badge examples

Badges within other elements

The **w3-badge** class can be used within other elements using a **** tag. For example, we might have a list of customers and use the badge to indicate number of orders.

Code Listing 54

```
<div class="w3-container w3-margin" style="width:18%;">
 <ul class="w3-ul w3-border">
  <li>Christy
   <span class="w3-badge w3-right w3-margin-right">4</span></li>
  <li>Rachel
   <span class="w3-badge w3-right w3-margin-right">8</span></li>
  <li>Alyssa
   <span class="w3-badge w3-right w3-margin-right">3</span></li>
 </ul>
</div>
```

This code snippet produces the list shown in Figure 41.

Figure 41: Badges with list elements

CSS tags

Tags are labels on a site, typically used to label some element on the screen. The W3.CSS framework provides the **w3-tag** class to format the label. The following code snippet shows a tag that displays the status of the current operation.

Code Listing 55

```
<div class="w3-container>
  <p>Status:  <span class="w3-tag w3-green">Processing</span></p>
</div>
```

Figure 42: Processing tag

Customizing the tags

You can use the various **w3** helper classes to customize the appearance of a tag. For example, the following code snippet adds several helper classes to produce a warning tag.

Code Listing 56

```
<div class="w3-container">
   <span class="w3-tag w3-red w3-center w3-round-large w3-padding">
    The thermal printer should only be<br/>
    used to print prescriptions sheets.
   </span>
</div>
```

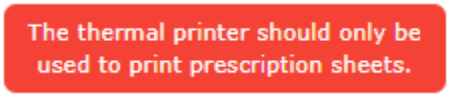

Figure 43: Warning tag

Summary

The button, badge, and tag classes make it easy to style your contents, and are summarized as follows:

- **`w3-button`**: Basic button with gray background on hover.
- **`w3-btn`**: Button with shadow effect on hover.
- **`w3-badge`**: Creates a circular “badge” effect.
- **`w3-tag`**: Creates a tag or label effect.

Chapter 11 Forms

Many web applications need to collect information from the user, and HTML provides a variety of elements to enter information into the site. The W3.CSS framework provides classes to create a consistent appearance for these elements.

Figure 44 shows a sample input form, formatted with W3 classes.

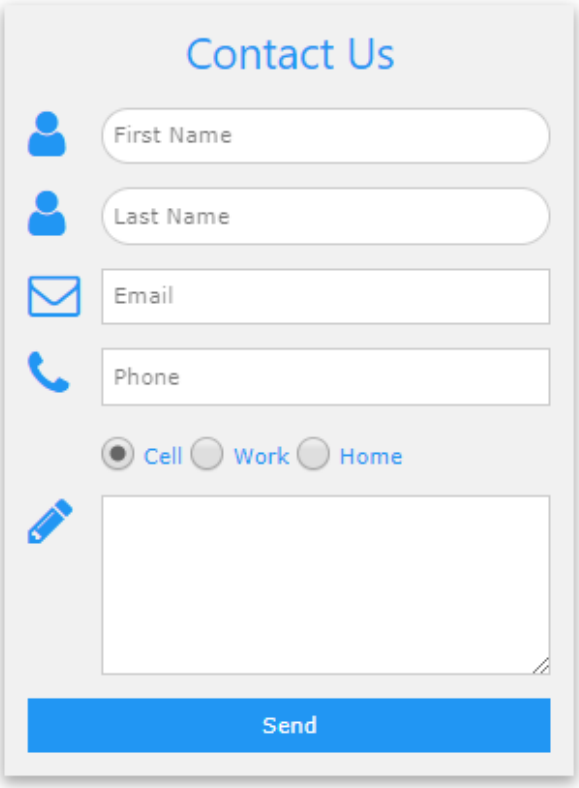

Figure 44: Sample input form

Text boxes

Text boxes are used to allow the user to enter freeform text into a form. The basic syntax is as follows.

Code Listing 57

```
<input type="text" class="w3-input">
```

The **w3-input** class provides the basic formatting for a text box, providing the necessary padding and borders.

Note: HTML5 introduces several new input types besides the basic text input. The following types are supported:

- ***date: Display a date editor.***
- ***email: Get an email address.***
- ***number: A numeric spinner.***
- ***range: A slider between two numeric values.***
- ***tel: Get a phone number.***
- ***color: A color picker.***
- ***datetime-local: Enter a date/time, and no time zone.***
- ***month: Select a month and year.***
- ***search: A text box for searching.***
- ***time: A box to select a time and a.m. or p.m.***
- ***url: A text box that expects input to look like a URL.***
- ***week: Pick a week and year.***

If a browser does not support one of the new types, it will fall back to text input. Also, many mobile devices will change the virtual keyboard to accommodate the input type.

Text box options

You can use the **w3** helper classes to change the appearance of the input text box. For example, the following code snippet produces our rounded input boxes in the sample.

Code Listing 58

```
<input type="text" class="w3-input w3-border w3-round-xxlarge"
   name="first" placeholder="First Name">
```

A recent trend in text input is not to use borders at all around text. You can use the **w3-border-0** class to achieve that look (shown in figure 45).

Figure 45: Borderless input

While text elements are very common and easy for the user to understand, the lack of constraints on what the user can enter makes it more difficult for the code to process. When possible, use the HTML5 types with constraints and provide placeholder examples of how the user should populate the text boxes.

Option buttons

Option buttons (also referred to as radio buttons) present a list of choices to the user and allow them to pick only one. They appear as circular buttons with labels next to each option. Figure 46 shows sample option buttons.

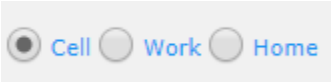

Figure 46: Option buttons

The **`w3-radio`** class formats the option buttons. The following code snippet shows how to create a group of buttons.

Code Listing 59

```
<input type="radio" class="w3-radio" name="phoneType" value="cell" checked>
<label>Cell</label>
<input type="radio" class="w3-radio" name="phoneType" value="work">
<label>Work</label>
<input type="radio" class="w3-radio" name="phoneType" value="home">
<label>Home</label>
```

All elements within the option group must have the same **`name`** attribute value. This allows the browser to enforce only one item within the group being selectable.

Note: Option buttons were originally called radio buttons because most older cars had buttons on the radio, designed so that when one button was pushed, the other buttons would automatically pop up (see Figure 47). Many younger developers were not familiar with the meaning, since very few cars still use radio buttons—hence the new name, option buttons.

Figure 47: Radio buttons

Check boxes

Check boxes are like option buttons, except that the user can check more than one. The element for check boxes is a square, to distinguish them from option buttons. Figure 48 shows how check boxes appear on the form.

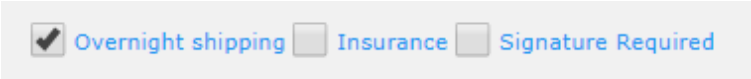

Figure 48: Check boxes

The following code creates the check boxes.

Code Listing 60

```
<input type="checkbox" class="w3-check" checked>
<label>Overnight shipping</label>
<input type="checkbox" class="w3-check" >
<label>Insurance</label>
<input type="checkbox" class="w3-check" >
<label>Signature required</label>
```

The **w3-check** class provides the formatting for check boxes.

Select elements

Check boxes and option buttons are good elements to use if your number of choices is relatively small. For a larger number of choices, you should consider a select box. To create the box, you declare the **<select>** element wrapper, and then include **<option>** elements. The following code snippet shows how to design a select list. The **w3-select** class provides the formatting. In the following example, we added a **w3-border** class to improve the appearance of the box.

Code Listing 61

```
<select class="w3-select w3-border" name="option">
  <option value= "" disabled selected>Choose your option</option>
  <option value= "1">Reduced interest rate</option>
  <option value= "2">Longer payment terms</option>
  <option value= "3">Cash rebate</option>
</select>
```

Figure 49 shows a select box produced from this code snippet.

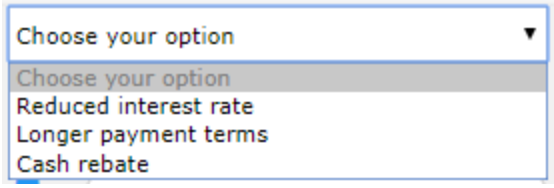

Figure 49: Select box

Adding labels

You can add labels around the input elements, using either text or icons and images. In the following code snippet, we are using the **w3-row** and **w3-col** classes to split the screen between the label and the input element.

Code Listing 62

```
<div class="w3-section w3-row">
  <div class="w3-col" style="width:50px;">
   <i class="w3-col fa fa-envelope-o"></i></div>
  <div class="w3-rest">
   <input class="w3-input w3-border" type="email"
      name="email" placeholder="Email address" >
  </div>
</div>
```

This example uses the Font Awesome library to provide an envelope icon. You could also use the HTML Entity code **✉** to produce an envelope icon.

Summary

The following classes are used to format input elements:

- **w3-input**: Text input class
- **w3-radio**: Radio (option) buttons
- **w3-check**: Check boxes
- **w3-select**: Select list

Chapter 12 Animations

Animation is a CSS technique that causes elements to gradually appear on the website. Judicious use of animation can make a website seem friendlier. For most people, things that appear suddenly in front of us are startling, and possibly dangerous. A dialog box that gradually fades in can provide a smooth, less jarring interface.

Animating elements

To animate any element's appearance on the page, you simply need to add the class indicating how you want the element to appear. There are several classes for animation.

Directions

You can animate the element from the top or bottom, or from either side. The following classes are used for directional animation:

- **`w3-animate-top`**: Slides element down from the top.
- **`w3-animate-bottom`**: Slides element up from the bottom.
- **`w3-animate-left`**: Slides element from the left side.
- **`w3-animate-right`**: Slides element from the right side.

Zooming

The zooming animation gradually increases the element's size from 0 to 100 percent. The animation occurs over six-tenths of a second. The **`w3-animate-zoom`** class attached to the element performs the zooming animation.

Opacity

Opacity is a setting that determines how transparent an element appears on the screen. A setting of **`1`** is totally opaque. Lower values create a translucent effect on the element. The W3.CSS framework provides two classes to animate the opacity of the element.

Animate opacity

Adding the **`w3-animate-opacity`** class causes a fade-in effect, where the element goes from totally translucent (invisible) to opaque in 1.5 seconds. You can use this effect with a modal dialog to gradually bring up the dialog box.

Fading

The **w3-animate-fading** class creates a fade-in and then fade-out effect on the element, where the opacity will go from 0 to 1 in 1.5 seconds, and then from 1 to 0 again.

Spinner

The **w3-spin** class causes the element to rotate 360 degrees and can be used to suggest to a user that a process is running. If you install Font Awesome, there are a few icons that make effective spinners to indicate a process is running. The code snippet below shows a spinning icon.

Code Listing 63

```
<div class="w3-section w3-row">
  <div class="w3-col" style="width:50px;">
   <i class="w3-xxlarge w3-spin fa fa-spinner"></i></div>
  <div class="w3-rest">Saving...</div>
</div>
```

Figure 50 shows the screen (but on a website, the spinner will keep rotating).

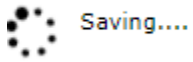

Figure 50: Sample spinner

Tip: Spinners are often used during Ajax calls, to show something is happening. However, be sure that if an error occurs, the spinner is removed. Otherwise, users see the spinner still moving, and complain about performance.

Summary

Animation can improve the general feel of a site, making your pop-up dialog boxes appear smoother, and allowing you to show processing activities. The W3.CSS animation classes make it easy to add animation.

If the animation speed is too fast or slow, you can easily clone the base CSS classes and create a **w3-animate-zoom-slowly** class, simply by adjusting the number of seconds that the animation takes. If you consider this approach, be sure to create a separate CSS so you won't lose your changes if you update the W3.CSS framework.

Chapter 13 Modals

Modal dialogs are pop-up windows that appear on a website, overwriting the content beneath the dialog. In general, they are forms for the user to complete for tasks such as editing a customer or selecting airlines. W3.CSS allows you to create a modal form and keep it hidden until some button or action triggers the modal to display.

Creating a modal dialog

To create a modal dialog, you need to use the **`w3-modal`** class for the container **`<div>`** that will hold the modal dialog's contents. You will also need to ensure that this container has a unique ID, since you'll need it to open the modal dialog. The following code snippet shows the basic setup for the modal container.

Code Listing 64

```
<div id="ContactUs" class="w3-modal w3-center">
</div>
```

The actual content of the modal dialog is wrapped within the **`w3-modal-content`** class. The wrapper structure looks as shown in the following.

Code Listing 65

```
<div id="ContactUs" class="w3-modal w3-center">
  <div class="w3-modal-content">
   <!-- Actual modal content -->
  </div>
</div>
```

Displaying the modal

To display the modal dialog, you will need some simple JavaScript and a button. For example, if we wanted to display the Contact Us modal dialog, we could use the following code.

Code Listing 66

```
<button onclick="document.getElementById('ContactUs').style.display='block'"
    class="w3-button">
</button>
```

When the user clicks the **Contact Us** button, the modal dialog's display style is set to **`block`**, causing the dialog to pop up on the user screen.

Animating the display

The modal dialog will suddenly pop up on the screen. While this is common in websites, it can sometimes have a jarring effect. You can take advantage of the animation classes to present a smoother appearance of the modal. Simply add the desired animation class to the container ** **, as follows.

Code Listing 67

```
<div id="ContactUs" class="w3-modal w3-center w3-animate-opacity">
</div>
```

This will cause the modal dialog to fade in, rather than just quickly appear.

Closing the modal dialog

Once the modal dialog is opened, it will stay on the screen (since the display style is now set to **block**). You will need to add some imbedded JavaScript code to allow the user to close the modal dialog. Typically, this is triggered by clicking the X icon in the upper-right corner, although if the modal dialog saves information, you would put the close code into the buttons that save and/or cancel the dialog.

Code Listing 68

```
<header class="w3-teal w3-display-container">
 <span onclick="document.getElementById('ContactUs').style.display='none'"
    class="w3-button w3-large w3-display-topright">&times;</span>
  <h2>Contact Us</h2>
</header>
```

Summary

Modal dialogs are typically used to overlay the website and provide a dialog for the user to focus on. The framework requires a little additional markup and some JavaScript to open and close the dialog.

Animating the dialog opening will add a smoother feel to the site, and you should consider adding one of the animation classes to the modal container.

Chapter 14 Images

You can use the **<img>** tag to display a picture on a website. The W3.CSS framework provides several classes that can be used to control the image's borders and appearance.

CSS Images

Here is the HTML to display an image.

Code Listing 69

```
<div class="w3-container w3-margin">
 <img src="BabyGirlCuddles.jpg" alt="Granddaughter">
 </div>
```

This will produce the image shown in Figure 51.

Figure 51: Grandpa cuddles

We can use the following classes to control how the image appears on the page:

- **w3-circle**: Places the image in a circle.
- **w3-rounded**: Adds rounded corners.
- **w3-bordered**: Adds a border around entire image.

Figure 52: Sample image classes

Responsive images

You can add the **`w3-image`** class to the **`<img>`** tag to create a responsive image. The image size will adjust to the screen, but will never be larger than the original image size.

Note: If you want a full responsive image (meaning it can be larger than original size), you need to set the width property to `100%` (via an inline style).

Image opacity

You can use the opacity classes to set the opacity of the image from 25 percent to 75 percent. The class names are:

- **`w3-opacity-min`**: 75 percent opacity
- **`w3-opacity`**: 50 percent opacity
- **`w3-opacity-max`**: 25 percent opacity

Figure 53 shows the various opacity effects.

Figure 53: Image opacity

Grayscale

You can use the grayscale classes to set the level of grayscale for an image. The class names are:

- w3-grayscale-min: Minimal grayscale effect.
- w3-grayscale: Mostly grayscale image.
- w3-grayscale: Full grayscale effect.

Figure 54 shows the grayscale effect on an image.

Figure 54: Grayscale

CSS slideshow

You can use the W3.CSS framework and a bit of JavaScript to create a slideshow of images. Figure 55 shows the basic appearance of the slideshow. The user can click the buttons to move among the images in the slideshow.

Figure 55: Slideshow

Setting the images

The first step is to add the images you want to view and assign them the same class name. The following snippet shows the image setup and the two buttons for navigation.

Code Listing 70

```
<div class="w3-display-container w3-content" style="width:8%;">
  <img class="w3-image mySlides" src="BabyGirlCuddles.jpg">
  <img class="w3-image mySlides" src="FingersSmall.jpg">
  <p>
   <button class="w3-button w3-black"
    onclick="plusDivs(-1)">&#10094;</button>
   <button class="w3-button w3-black"
    onclick="plusDivs(1)">&#10095;</button>
  </p>
</div>
```

The HTML entities are used for the buttons to provide the navigation arrows. We will need a JavaScript function called **`plusDivs()`** to move among the images.

JavaScript code

The following is the code that needs to be added within a script tag in the **`<head>`** section of the website to allow the navigation between images.

Code Listing 71

```
<script>
 var slideIndex =1;
 showDivs(1);

 function plusDivs(n) {
  showDivs(slideIndex += n);
  }

  function showDivs(n) {
  var x = document.getElementsByClaseName('mySlides');
 if (n > x.length) { xslideIndex =1;}
 if (n < 1) { slideIndex =x.length;}
 for (var i=0; i < x.length; i++) {
   x[i].style.display  ="none";
  }
  x[i].style.display  ="block";
 }
</script>
```

The script declares a variable (**slideIndex**) to keep track of the currently selected image and initializes it to the first image. The buttons will call the **plusDivs()** function with a parameter indicating the next image to display.

ShowDivs() will simply set the display style of all the images to **none**, and then set the selected image's style to **block**, causing it to be shown.

Summary

W3.CSS provides classes to control the appearance and shape of images, and makes it very easy to create a slideshow of your image collection with minor JavaScript.

Chapter 15 W3.CSS Example Code

We've covered a lot of classes and examples in this book. In this chapter, we will explore in detail one of the examples from the W3.CSS site. The email template is a good, simple demo of a responsive website for a mail client. Figure 56 shows the sample template.

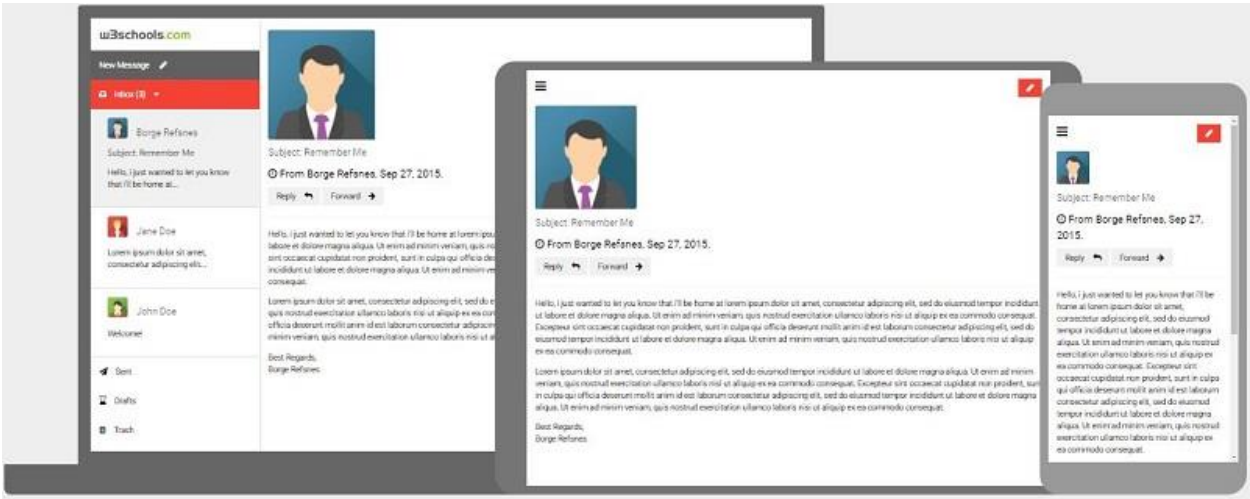

Figure 56: Email template

Head section

The **<head>** section of most templates will look very much the same, including the W3.CSS style sheet, possibly using an external web font.

Code Listing 72

```
<!DOCTYPE html>
<html>
<head>
 <title>W3.CSS </title>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <link rel="stylesheet" type="text/css"
   href="https://fonts.googleapis.com/css?family=RobotoDraft">
```

In this example, we are using a Google font called **RobotoDraft**, and linking it into our website. While the use of external web fonts is optional, you can improve the overall look of the site easily with minimal code changes.

Font Awesome

Font Awesome is a very popular library of icons, defined as CSS classes. You can explore the library here.

To add Font Awesome to the site, the following line is added to the **<head>** section.

Code Listing 73

```
<link rel="stylesheet" type="text/css"
  href="https://cdnjs.cloudfare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
```

To use the fonts, you simply add the class name to any element, and the appropriate icon will appear. The following example shows some Save and Cancel buttons taking advantage of the Font Awesome library.

Code Listing 74

```
<button class="fa fa-save w3-text-green w3-xlarge w3-padding">
   Save changes</button>
<button class="fa fa-times w3-text-red w3-xlarge w3-padding">
   Cancel</button>
```

This produces the buttons shown in Figure 57.

Figure 57: Font Awesome buttons

HTML entities

Throughout the course of this book, we also took advantage of some HTML entities to add small icons to a few of our examples. Some useful HTML entities are:

- **☰** Hamburger menu
- **📅** Calendar
- **🔍** Search icon
- **✎** Edit pencil
- **જ** Cancel icon
- **઎** Scissors
- **☎** Telephone
- **⦻** Circle with X

While HTML entities are easy to use, not all entities will work with all fonts. Be sure to test your entities against your website font or use the Font Awesome library, which imbeds its own font.

Setting the font family

The final step when using a web font is to adjust the styles on various elements to use the new font family. The final piece of code in the **<head>** section does that.

Code Listing 75

```
<style>
 html, body, h1, h2, h3, h4, h5, h6 {
  font-family: "RobotoDraft","Roboto", sans-serif;
 }
 .w3-bar-block, .w3-bar-item { padding:16px; }
</style>
```

Body section

The body section consists of a sidebar menu for navigation among the email folders (the inbox, sent items, drafts, and deleted items). Note that there is no implementation code for these folders; it would typically be an Ajax call to gather all the items within the folder and display them.

Sidebar

The following code creates the sidebar navigation menu (some classes are removed for readability).

Code Listing 76

```
<nav class="w3-sidebar w3-bar-block w3-collapse w3-card" id="mySideBar"
   style="width:200px;">
   <a href="#" onclick="w3_close();" title="Close side menu"
   class="w3-bar-item w3-button w3-hide-large">Close
   <i class="fa fa-remove"></i></a>
   <a href="#" class="w3-bar-item w3-button w3-left-align"
   onclick="document.getElementById('id01').style.display='block'">
   New Message <i class="w3-padding fa fa-pencil"></i></a>
   <a id="myBtn" href="#" onclick="myFunc('Demo1');"
   class="w3-bar-item w3-button">
   <i class="fa fa-inbox w3-margin-right"></i>Inbox  
   <span class="w3-tag w3-circle">3</span>
   <i class="fa fa-caret-down w3-margin-left"></i></a>
   <a href="#" class="w3-bar-item w3-button">
   <i class="fa fa-paper-plane w3-margin-right"></i>Sent</a>
   <a href="#" class="w3-bar-item w3-button">
   <i class="fa fa-houeglass-end w3-margin-right"></i>Drafts</a>
   <a href="#" class="w3-bar-item w3-button">
```

```
   <i class="fa fa-trash w3-margin-right"></i>Trash</a>
</nav>
```

Although the code uses several classes and structures, there are a few items worth noting. First is the **w3-hide-large** class on the Close sidebar menu item. This indicates that the Close option will not appear on larger devices, where it's assumed there is enough screen room for both the sidebar menu and the email contents.

Compose new mail

When the user clicks on a new email, a modal dialog will be displayed to get the information for the email.

Code Listing 77

```
<!-- Modal that pops up for new message -->
<div id="id01" class="w3-modal">
  <div class="w3-modal-content w3-animate-zoom">
    <div class="w3-container w3-padding w3-red ">
    <span
     onclick="document.getElementById('id01').style.display='block'"
     class="w3-button w3-right w3-xxlarge">
     <i class="fa fa-remove"></i>
   </span>
   <h2>Send Mail</h2>
    <!-- Form elements... -->
    </div>
  </div>
</div>
```

Notice that the pop-up dialog uses the zoom animation to appear on the screen, and the header section provides the ability to close the dialog.

The button code shown in the following code snippet will close the dialog box once the user either sends or cancels the email.

Code Listing 78

```
<a class="w3-button w3-red"
  onclick="document.getElementById('id01').style.display='none'">
  Cancel <i class="fa fa-remove"></i></a>
<a class="w3-button w3-right"
  onclick="document.getElementById('id01').style.display='none'">
  Send <i class="fa fa-paper-plane"></i></a>
```

Summary

The templates and example code on the site are a great way to see the framework with some working code examples.

Chapter 16 Versions

There are two additional versions of the framework, depending on your needs. Both versions are also free, and are smaller than the base W3.CSS file. However, they do not include any colors; it is up to you to define your own colors.

CSS Pro

The CSS Pro version, available here, is 16 KB. You can add your own color support with your own style sheet, in-line styles, or a W3 theme. For example, if you wanted to use the flat color scheme, rather than the default material design colors, you could import the following style sheets.

Code Listing 79

```
<html>
<head>
 <title>W3 mobile with metro colors</title>
 <meta name="viewport" content="width=device-width,initial-scale=1">
 <link rel="stylesheet" href="css/w3pro.css">
 <link rel="stylesheet" href="css/w3-colors-flat.css">
```

This will result in a smaller style sheet, and only make the flat CSS colors available to your site.

CSS Mobile

The CSS Mobile version, available here, is 14KB. You can add your own color support with your own style sheet, in-line styles, or a W3 theme. In addition to no colors, this sheet also has no mobile-first media queries—it is strictly intended for mobile devices.

To use this style sheet with the metro colors, you could use the following code snippet.

Code Listing 80

```
<html>
<head>
 <title>W3 mobile with metro colors</title>
 <meta name="viewport" content="width=device-width,initial-scale=1">
 <link rel="stylesheet" href="css/w3mobile.css">
 <link rel="stylesheet" href="css/w3-colors-metro.css">
```

Chapter 17 Summary

W3.CSS is a simple and fast framework, making it easier to apply CSS styles to create responsive websites. Its color support and responsive design, as well as consistent and simple class names, make it an easy-to-use framework that can be used comfortably with other frameworks and libraries.

Visit the W3.CSS website for updates to the framework and color libraries and themes, as well as online documentation and interactive examples to let you try out the framework.

Site

The W3.CSS site provides examples of how to create various elements with the classes. In addition, there are several demo sites available, such as a photo album, newspaper layout, and blog.

There are several templates available, which are complete sites for various types of businesses. These can save you some startup time if you are creating a website for a business. These templates are responsive, as you can see from the snapshot of the social media site template shown in Figure 58.

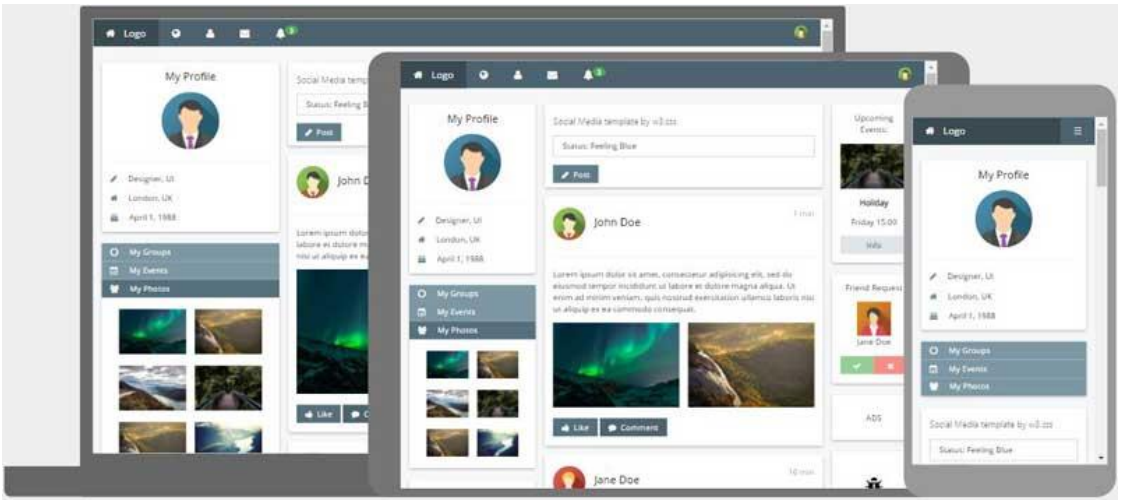

Figure 58: W3.CSS Social media template

If you are looking for an easy-to-use, free CSS framework with lots of examples, templates, and online documentation, W3.CSS is certainly a worthwhile contender.