Jennifer Kyrnin

Sams Teach Yourself

Responsive Web Design

in 24 Hours
Contents

Introduction
Making Your Site “Every Device” Friendly .................................................. 1
How to Use This Book .................................................................................. 1

PART I INTRODUCTION TO RESPONSIVE DESIGN

History of Responsive Web Design ................................................................. 5
Why We Need Responsive Web Design ....................................................... 11
Summary ......................................................................................................... 12
Workshop ....................................................................................................... 13

HOUR 2: Alternatives to Responsive Web Design ........................................ 15
Table-Based Layouts ...................................................................................... 15
CSS Layouts .................................................................................................. 21
Detection Scripts ............................................................................................ 30
Summary ......................................................................................................... 32
Workshop ....................................................................................................... 35

HOUR 3: The Growth of Mobile ................................................................... 37
Basic Cell Phones ........................................................................................... 37
Smartphones ..................................................................................................... 39
Tablets ............................................................................................................. 41
Retina Devices ................................................................................................. 42
Why Responsive Design Is Important .......................................................... 44
Summary ......................................................................................................... 44
Workshop ....................................................................................................... 45

HOUR 4: Progressive Enhancement ............................................................... 47
How to Use Progressive Enhancement on a Website .................................... 48
Benefits of Progressive Enhancement .......................................................... 50
Summary ......................................................................................................... 51
Workshop ....................................................................................................... 52

HOUR 5: HTML for Responsive Web Design ............................................... 55
Using HTML5 .................................................................................................. 55
Clean Code ...................................................................................................... 62
Don’t Forget Semantic Elements .................................................................. 63
Validating Your HTML ................................................................................... 67
Summary ......................................................................................................... 68
Workshop ....................................................................................................... 70

HOUR 6: Basic CSS ....................................................................................... 73
How to Write CSS Rules .................................................................................. 73
Embedded and External Style Sheets ........................................................... 76
Styling Fonts and Colors .............................................................................. 78
Creating a Layout with CSS .......................................................................... 83
Understanding Cascading and Specificity .................................................... 88
Summary ......................................................................................................... 90
Workshop ....................................................................................................... 92

HOUR 7: Unobtrusive JavaScript .................................................................. 95
What Is Unobtrusive JavaScript? ................................................................. 95
How to Implement Unobtrusive JavaScript .................................................... 98
Summary ......................................................................................................... 101
Workshop ...................................................................................................... 102

PART II BUILDING A RESPONSIVE WEBSITE

HOUR 8: Planning a Responsive Website ...................................................... 105
Should You Make Your Website Responsive? ............................................. 106
How to Plan for a Responsive Website ....................................................... 107
Summary ......................................................................................................... 112
Workshop ....................................................................................................... 113

HOUR 9: Mobile First .................................................................................... 115
Why Design for Mobile First? .................................................................... 115
What Makes a Site Mobile Friendly? ............................................................. 118
What About Mobile Only? ........................................................................... 121
Summary ......................................................................................................... 121
Workshop ....................................................................................................... 123

HOUR 10: CSS Media Queries .................................................................... 125
What Is a Media Query? ............................................................................... 125
Media Query Expressions ............................................................................ 129
Summary ......................................................................................................... 130
Workshop ....................................................................................................... 132

HOUR 11: Breakpoints ................................................................................. 135
What Is a Breakpoint? ................................................................................ 135
How to Define Breakpoints in CSS .............................................................. 136
Optimal Breakpoints .................................................................................... 141
Summary ......................................................................................................... 144
Workshop ....................................................................................................... 145

HOUR 12: Layout .......................................................................................... 147
What Is Web Layout? ................................................................................... 147
Types of Layouts ............................................................................................ 149
Columns in Layout ........................................................................................ 158
Summary ......................................................................................................... 162
Workshop ....................................................................................................... 164

HOUR 13: Navigation .................................................................................... 165
Why Responsive Navigation Is Important .................................................. 165
What Makes Navigation Mobile Friendly? 166
Basic RWD Navigation Patterns 168
Summary ........................................... 179
Workshop .......................................... 180

HOUR 14: Responsive Fonts and Typography 183
Using Web Fonts ................................. 183
Sizing Typography ................................. 187
Relative Versus Absolute Font Sizes 193
New CSS3 Measurement Units .......... 195
Summary ........................................... 196
Workshop .......................................... 197

HOUR 15: Creating and Using Images in RWD 199
Making Images Responsive .................. 199
Improving Download Speeds ............... 207
Building and Using Retina-Ready Images 207
Summary ........................................... 211
Workshop .......................................... 212

HOUR 16: Videos and Other Media in RWD 215
How to Make Videos Responsive .......... 215
Making YouTube Videos Responsive ..... 219
Summary ........................................... 221
Workshop .......................................... 222

HOUR 17: Tables in Responsive Web Design 225
Tables on Small Devices ....................... 225
Can Tables Be Responsive? ................. 226
Where Do Layout Tables Fit in RWD? ... 236
Summary ........................................... 240
Workshop .......................................... 241

HOUR 18: Responsive Web Forms 245
HTML5 Forms .................................... 245
Making Web Forms Usable ................. 251
Creating Responsive Forms ............... 254
Summary ........................................... 258
Workshop .......................................... 259

HOUR 19: Testing Responsive Websites 263
Testing in Your Browser ....................... 264
Testing in a Device for All Your Breakpoints 266
How to Test When You Don’t Have the Devices ........................................... 271
Summary ........................................... 273
Workshop .......................................... 274

HOUR 20: Problems with Responsive Web Design 277
Responsive Designs Can Be Slow ......... 277
RWD Can Make More Work for Designers 280
Not All Customers Like Responsive Sites 282
RWD May Break Advertising .............. 283
Summary ........................................... 286
Workshop .......................................... 287

PART III IMPROVING ON RESPONSIVE WEB DESIGN

Planning and Designing Your RWD Site 291
HTML Element and CSS Tools .......... 295
Web Editors for Building Responsive Web Pages 302
Summary ........................................... 303
Workshop .......................................... 304

HOUR 22: Device and Feature Detection 307
Why Use Detection Scripts ................. 307
Modernizr .......................................... 311
WURFL ............................................. 313
Summary ........................................... 315
Workshop .......................................... 316

HOUR 23: Using RESS with RWD 319
What Is RESS? .................................... 319
Benefits of Using RESS ...................... 320
Getting Started with RESS ............... 320
When to Use RESS .............................. 328
Summary ........................................... 330
Workshop .......................................... 331

HOUR 24: RWD Best Practices 333
Give Everyone the Best Experience ...... 334
Use the Best Breakpoints You Can ...... 336
Be Flexible and Think Small ............... 338
Don’t Forget the Content ..................... 340
Manage Costs .................................... 342
Summary ........................................... 344
Workshop .......................................... 345

Index .................................................. 347
About the Author

Jennifer Kyrnin has been teaching HTML, XML, and web design online since 1997. She has built and maintained websites of all sizes, from small, single-page brochure sites to large, million-page database-driven sites for international audiences. She lives with her husband and son and numerous animals on a small farm in Washington state.

Dedication

To Mark and Jaryth. This was much easier because you were around. I love you.

Acknowledgments

I would like to thank all the people at Sams Publishing for the opportunity to write this book and work with you. I would particularly like to thank Mark Taber and Seth Kerney for keeping me moving and the book on track, as well as my fabulous technical editor, Jon Morin, and my amazing and tolerant copy editor Kitty Wilson for all the great suggestions and corrections. Any technical errors you find in the book are mine alone; the editors probably tried to stop me.

I would also like to thank my family for putting up with me while I wrote the book, including my brother Brendan Kirby, who was instrumental in helping me get my thoughts together on one tough section, and the members of the Woodinville Writers Group, who were very kind in letting me interrupt their fiction stories with some cold hard HTML.
We Want to Hear from You!

As the reader of this book, you are our most important critic and commentator. We value your opinion and want to know what we’re doing right, what we could do better, what areas you’d like to see us publish in, and any other words of wisdom you’re willing to pass our way.

You can email or write to let us know what you did or didn’t like about this book—as well as what we can do to make our books stronger.

Please note that we cannot help you with technical problems related to the topic of this book, and that due to the high volume of mail we receive, we might not be able to reply to every message.

When you write, please be sure to include this book’s title, edition number, and author, as well as your name and contact information.

Email: feedback@samspublishing.com
Mail: Sams Publishing
     800 East 96th Street
     Indianapolis, IN 46240 USA

Reader Services

Visit our website and register this book at informit.com/register for convenient access to any updates, downloads, or errata that might be available for this book.
Introduction

Responsive web design (RWD) is a way of thinking about web pages that allows designers to work on one website for all visitors but tailor the site to the device each visitor is viewing it on. More and more companies are requiring web designers to build their sites responsively, and knowing how to do it well will help web designers differentiate themselves in the industry.

Making Your Site “Every Device” Friendly

This book covers more than just how and when to build CSS breakpoints to build a responsive site. Once you’ve finished this book, you will understand all these ideas:

► How the use of RWD got started and why web designers began using it instead of other techniques
► What progressive enhancement is and how important it is to RWD
► Basic HTML5, CSS3, and JavaScript to build a website framework that is easy to make responsive
► How mobile devices affect responsive design and how they aren’t the whole story when it comes to creating responsive websites
► The basics of building RWD, including media queries and breakpoints
► How to make your layout, navigation, images, tables, fonts, videos, and forms responsive
► How to test your designs even if you don’t have a lot of mobile devices
► The common problems in RWD and how to alleviate them
► How to use RESS and other tools and technologies to improve your designs
► Best practices for using RWD and building a responsive site

How to Use This Book

This book is divided into 24 lessons, called “hours.” Each lesson covers a specific topic related to building responsive web pages using responsive web design. Each lesson takes about an hour to complete.
Organization of This Book

The book is divided into three sections:

- Part I, “Introduction to Responsive Design,” introduces you to RWD and explains the basic HTML, CSS, and JavaScript you need to know.

- Part II, “Building a Responsive Website,” addresses specific aspects of RWD, such as navigation, images, and tables, and shows you how to make them responsive.

- Part III, “Improving on Responsive Web Design,” introduces you to some tools and techniques you can use to improve your RWD and describes problems you may have as well as best practices in the field.

Conventions Used in This Book

Code samples are written in monospaced font within the text of the book, while blocks of code appear separately, like this:

This is a block
Of code

Some code samples that are too long to display as one line in the book use the ➥ symbol to indicate that these lines should be all on one line, like this:

```html
<link rel="stylesheet" href="styles-320.css" media="only screen and (max-width:320px)">
```

This book has three types of sidebars:

<table>
<thead>
<tr>
<th>NOTE</th>
<th>TIP</th>
<th>CAUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notes provide additional information about the topics that are discussed in the hour.</td>
<td>Tips share interesting facts or tidbits about the related content.</td>
<td>Cautions alert you to things that can cause problems for your web designs.</td>
</tr>
</tbody>
</table>

You can also use the Try It Yourself sections to practice what you’ve learned in the hour.

TRY IT YOURSELF

Try It Yourself

Nearly every hour has at least one step-by-step tutorial called “Try It Yourself” to help you use what you’ve learned.
Q&A, Quiz, and Exercises

Every hour ends with a short question-and-answer section that anticipates follow-up questions you may have after reading the hour. You can also take a short quiz on what you’ve learned as well as do some suggested exercises to help you get more out of what you learned and apply your new knowledge to your own web designs.

Where to Go to Learn More

This book has a companion website at http://html5in24hours.com, where you can go to see the examples, view and download the source code, view and report errata about the book, and continue to learn and ask questions about RWD. You can also find Jennifer Kyrnin online at http://htmljenn.com, and she welcomes questions and comments.
This page intentionally left blank
Responsive web design doesn’t add any new HTML tags or attributes: You simply write your HTML so that it is well-formed, valid, and semantic.

The best responsive sites are sites that also use progressive enhancement, as you learned in Hour 4, “Progressive Enhancement.” These sites keep the HTML, CSS, and JavaScript separate so that the pages are easier to maintain and load more quickly.

In this hour you will learn the basics of HTML so that you can put together a decent website and make it responsive. However, there is more to HTML than you can learn in just one hour, so if you don’t know HTML at all, you should consider a book like *Sams Teach Yourself HTML and CSS in 24 Hours* by Julie C. Meloni.

**Using HTML5**

HTML5 is the most recent version of HTML and provides the most assistance to web designers who want to use progressive enhancement and RWD. While you can build RWD sites using other versions of HTML, it’s best to stay as up-to-date as possible. This book uses HTML5 code samples.

**Tags Every Page Should Contain**

There are several HTML tags that every web page should contain. These tags may not be required for valid HTML, but they provide information
about the page to the browser to make them easier to use. Every web page should contain these tags:

- `<!doctype>`
- `<html>`
- `<head>`
- `<meta charset>`
- `<title>`
- `<body>`

If your web page contains these elements, it contains the minimum HTML required to start building a responsive page. Listing 5.1 provides a standard template you can use to start any web page. Note that the tags listed above have both starting and ending tags as well as some attributes.

**LISTING 5.1  A Basic HTML Template**

```html
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Untitled Document</title>
  </head>
  <body>
  </body>
</html>
```

The first line in the code is the doctype. The doctype `<!doctype html>` tells the browser that this is HTML5.

Then you place the opening `<html>` tag. Be sure to surround the entire document with this element by including the `</html>` tag at the very end. This tells the browser that this is an HTML document, and also it tells the browser where it should expect to see HTML tags.

Valid HTML keeps all the content inside the `<html>` and `</html>` tags, but if content slips out, the browser will still parse it. Some content management systems append content to the top or bottom of the HTML, without placing it inside these tags. This content will still be parsed and displayed by browsers, but it could cause problems. Keep your content inside the `<html>` and `</html>` tags, and you’ll be safe.

The `<head>` and `</head>` tags contain details about the web page that may not be visible to the browser—things like metadata, the title, and
any links to CSS and JavaScript. The basic template in Listing 5.1 includes a `<meta charset>` tag and a `<title>` tag.

The `<meta charset>` tag is an important one for keeping your web pages secure. It should always be the very first tag in the head of your HTML documents. It tells the browser what character set the page uses. If you don’t define the character set, or if you define it later in the document, you open up your site to hackers who can manipulate the character set and inject malicious code into the site.

Once you’ve defined the head information of your document, all you have left is the `<body>` element. This is where all the visible content goes. It contains what most people consider the meat of the web page. Anything that you type inside the `<body>` and `</body>` tags will show in the browser window.

**Basic Tags for Web Content**

While the tags already listed in this lesson are all you need to create a website, the site would be very plain and hard to read if you used only these tags. While there are dozens of HTML tags you can use (I list the new HTML5 elements at http://www.html5in24hours.com/2012/04/new-html5-elements/), there are only a few you need to know about to start creating a decent web page:

- `<h1>`, `<h2>`, and `<h3>`
- `<p>` and `<br>`
- `<a>`
- `<strong>` and `<em>`
- `<img>`, `<audio>`, and `<video>`

Most web pages start with a headline, and in HTML you use the headline tags `<h1>`, `<h2>`, and `<h3>` to define them. They are numbered, and you should use them in order, with `<h1>` first through `<h6>` last. Most sites don’t go more than two or three levels deep, so you really only need to know `<h2>` and `<h3>`.

Once you have a headline, you can start adding content in paragraphs by using the `<p>` tag. Browsers typically display a paragraph as a block of text separated by some space. But if you want to drop down just one line of text, you use the `<br>` tag. This is a singleton tag and does not require a closing tag. You will sometimes see it written as `<br/>`, with the additional slash. This is a remnant from XHTML. You don’t need the closing slash, and most browsers recognize the tag either way.
You can put these tags together to create a web page. Figure 5.1 shows how a simple web page would look, and Listing 5.2 is the HTML for creating that web page. As you can see, the HTML is still very plain. But because content is the most important part of your website, you need to make sure it is visible and correct first.

The \texttt{<a>} tag provides a link to another document. You define a link by using the \texttt{href=""} attribute. Then any text or image that is inside the \texttt{<a>} element will be clickable in the browser and will take the customer to the new location.

The \texttt{<strong>} and \texttt{<em>} tags let you provide some emphasis to the text in your paragraphs. In most browsers the \texttt{<strong>} tag makes the text bold and \texttt{<em>} makes the text italic. But you will be able to define how you want those to display in your style sheets. You’ll learn more about that in Hour 6, “Basic CSS.”

The last three tags let you add multimedia into your web pages. The \texttt{<img>} tag adds an image to the page, using the \texttt{src=""} attribute to define the location of the image on your web server. You can add sound files by using the \texttt{<audio>} tag and videos by using the \texttt{<video>} tag. I cover these in more detail in Hour 16, “Videos and Other Media in RWD.”

---

**CAUTION**

**Keep Your Tags Closed**

Several HTML elements don’t require closing tags, like the \texttt{<p>} element for paragraphs. While the closing \texttt{</p>} is not required, it’s a good idea to use it anyway. This helps keep your HTML clean and prevents less compliant browsers from getting confused. It’s a small price to pay to ensure that your content is seen by the largest number of people.
LISTING 5.2  Adding Some Content to the Template

```html
<!doctype html>
<html>
<head>
<meta charset="UTF-8">
<title>Dandylions</title>
</head>

<body>
<h1>Dandylions</h1>
<h2>Not Your Mother's Weed</h2>
<p><strong><em>Taraxacum</em></strong> <em>/təˈræksəkəm/</em> is a large genus of flowering plants in the family Asteraceae. They are native to Eurasia and North and South America, and two species, T. officinale and T. erythrospermum, are found as weeds worldwide. Both species are edible in their entirety. The common name <strong>dandelion</strong> (/<em>/'dændiˌlaɪn/ dan-di-ly-n, from French dent-de-lion, meaning "lion's tooth") is given to members of the genus, and like other members of the Asteraceae family, they have very small flowers collected together into a composite flower head. Each single flower in a head is called a floret. Many <em>Taraxacum</em> species produce seeds asexually by apomixis, where the seeds are produced without pollination, resulting in offspring that are genetically identical to the parent plant.</p>
</body>
</html>
```

**HTML for Layout**

You should also consider learning about some of the tags that are typically used for layout and style. There are two in particular that you should know:

- `<div>`
- `<span>`

These HTML tags provide no semantic meaning to your content and should be used to add hooks for CSS and JavaScript.

The `<div>` element acts like a container element, and you can place large blocks of text in it. It is called a block element, and most browsers add a line both before and after, similar to a paragraph.

Most people use `<div>` to create layout sections and then style those sections with CSS. Listing 5.3 shows how you might add some sections to your HTML so that you can style them later.
LISTING 5.3  HTML with Some Divisions

```html
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Dandylions</title>
</head>
<body>
  <div id="main">
    <h1>Dandylions</h1>
    <div id="article">
      <h2>Not Your Mother's Weed</h2>
      <p><strong><em>Taraxacum</em></strong> /tə'ræksəkəm/ is a large genus of flowering plants in the family Asteraceae. They are native to Eurasia and North and South America, and two species, <em>T. officinale</em> and <em>T. erythrospermum</em>, are found as weeds worldwide. Both species are edible in their entirety. The common name <strong>dandelion</strong> (<em>'dændi|ⅼən</em>, from French dent-de-lion, meaning "lion's tooth") is given to members of the genus, and like other members of the Asteraceae family, they have very small flowers collected together into a composite flower head. Each single flower in a head is called a floret. Many <em>Taraxacum</em> species produce seeds asexually by apomixis, where the seeds are produced without pollination, resulting in offspring that are genetically identical to the parent plant.</p>
    </div>
    <div id="sidebar">
      <img src="images/dandy.jpg" width="400" height="300" alt="" />
    </div>
  </div>
  <div id="footer">
  </div>
</body>
</html>
```

The `<span>` element works inside paragraphs to select small blocks of content. It is called an *inline* element. It does not add line breaks, and it allows you to mark up individual words in the content. Listing 5.4 shows a block of text with some words called out with the `<span>` tag.

LISTING 5.4  A Block of Text with Highlighted Words

```html
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Dandylions</title>
</head>
<body>
  <div id="article">
    <p><span>Not Your Mother's Weed</span> <strong><em>Taraxacum</em></strong> /tə'ræksəkəm/ is a large genus of flowering plants in the family Asteraceae. They are native to Eurasia and North and South America, and two species, <em>T. officinale</em> and <em>T. erythrospermum</em>, are found as weeds worldwide. Both species are edible in their entirety. The common name <strong>dandelion</strong> (<em>'dændi|ⅼən</em>, from French dent-de-lion, meaning "lion's tooth") is given to members of the genus, and like other members of the Asteraceae family, they have very small flowers collected together into a composite flower head. Each single flower in a head is called a floret. Many <em>Taraxacum</em> species produce seeds asexually by apomixis, where the seeds are produced without pollination, resulting in offspring that are genetically identical to the parent plant.</p>
  </div>
</body>
</html>
```
<div id="main">
  <h1>Dandylions</h1>
  <div id="article">
    <h2>Not Your Mother's Weed</h2>
    <p><span class="pronounce"><strong><em>Taraxacum</em></strong> is a large genus of flowering plants in the family Asteraceae. They are native to Eurasia and North and South America, and two species, <em>T. officinale</em> and <em>T. erythrospermum</em>, are found as weeds worldwide. Both species are edible in their entirety. The common name is given to members of the genus, and like other members of the Asteraceae family, they have very small flowers collected together into a composite flower head. Each single flower in a head is called a floret. Many species produce seeds asexually by apomixis, where the seeds are produced without pollination, resulting in offspring that are genetically identical to the parent plant.</p>
  </div>
  <div id="sidebar">
    <img src="images/dandy.jpg" width="400" height="300" alt="" />
  </div>
  <div id="footer">
  </div>
</div>

---

**Some Useful Attributes**

Nearly every tag in HTML has attributes. These are keywords that are defined within a tag and give the browser more information about that tag. You have already used attributes with the `<img src="">` tag and the `<a href="">` tag. But there are a couple more attributes that you should know about:

- `id`
- `class`

You can add these attributes to any HTML tag in your document to provide additional information about that element.
You use the `id` attribute to give an element a unique name. The `id` must be unique to the page it is on. In other words, there can be only one. But you can give every single tag on your page a unique `id`

You use `id` to identify an element. You can then link to that element by using the pound sign (`#`) in your URL, followed by the `id` value you assign. For example, if you had an element `<div id="main">`, you could then write a link to that element by typing `<a href="#main">link to main</a>`.

You can also use the `id` attribute as a hook for styles and scripts. Because it must be unique on the page, you know that when you attach a style to that `id`, you will affect only one element. This attribute also makes the style rule more specific, which means it’s more likely to be applied. You will learn more about CSS specificity in Hour 6.

Like the `id` attribute, the `class` attribute allows you to apply styles and scripts to an element. But it does not have to be unique on the page. This means you can apply a class to multiple elements on the page, and any style rules that are written to that class will be applied to all the elements. For instance, say you want some of your `<h1>` headlines to be red, but others should remain the default color. You could give the red headlines a special class that you would style as red in your CSS: `<h1 class="highlight">`.

One of the nicest things about using classes is that you aren’t limited to just one. You can include multiple class names on any element to add styles to the element or hook up with your scripts. To add a second class to an element, simply separate the classes with a space, like this: `<h1 class="highlight fancy">`.

### Clean Code

When you’re building HTML for RWD sites, you should always strive to keep it as clean and clear as possible. The technical term for this is *well-formed*. Well-formed HTML follows these guidelines:

- There is a doctype at the top of the document.
- Tags should nest correctly, inside to outside; for example, `<b><i>text</i></b>` is correct, and `<b><i>text</i></b>` is incorrect.
Don’t Forget Semantic Elements

Attributes with spaces in their values should be quoted using single or double quotation marks.

Comments are not allowed inside tags.

You should escape special characters used in HTML, such as ampersand (&amp;), a less-than sign (&lt;), and a greater-than sign (&gt;).

Well-formed XHTML involves additional rules, such as always closing every tag and using a closing slash in singleton tags, including an XML declaration, and quoting all attributes. But if you are using HTML5, the rules are not as strict.

The other part of clean code is using only the elements and attributes you need and nothing more. Try to consider what your site needs and limit your HTML to only those elements.

Think about what the content is when you are writing HTML. A paragraph should be in the <p> element. For a list, you should use one of the list elements. You get the picture. In the next section, you will learn about some of the new HTML5 elements that help define your content right in the HTML, using semantics.

**Don’t Forget Semantic Elements**

Semantic elements are elements that describe the content. They provide more information to the browser without requiring any extra attributes.

**Standard Semantic HTML Elements**

A number of semantic HTML elements have been in use for years—in some cases since HTML level 1. Here are some semantic HTML elements that are valid in HTML5 and earlier versions of HTML:

- `<abbr>` defines abbreviations and acronyms
- `<cite>` defines citations, such as for quotations
- `<code>` defines a code reference
- `<q>` defines a short, inline quotation

**CAUTION**

*Don’t Catch “Divitis”*

The most commonly overused HTML element is the `<div>` element. It can be tempting to surround elements with lots of `<div>` tags so that you have lots of things to hook into for your layout and scripts. But most of the time, you can remove these and attach styles directly to the specific elements you want styled. The only exception to this is container elements. Some browsers still have trouble applying container styles to the `<body>` element, and placing a `<div id="container"></div>` around your entire page can help.

I have also seen sites use the `<div>` tag instead of paragraph tags, with a class="paragraph" attribute on the `<div>`. This is ridiculous. HTML provides different elements for a reason, and you should use them. And as you'll see in the next section, there are even more elements you can use to mark up your content. You’re not limited to using only `<div>`.
New HTML5 Semantic Elements

HTML5 adds a lot of new elements that you can use to mark up your content semantically. The most useful ones are the sectioning elements:

- `<article>`
- `<aside>`
- `<nav>`
- `<section>`

These elements define areas of the content that are commonly found on web pages. Most web pages have a main article that defines the page (`<article>`); there is usually sidebar information for either the article, the page, or the entire site (`<aside>`); navigation is critical (`<nav>`); and many web pages are divided into separate sections with different semantic meanings that don’t fall into the other categories (`<section>`).

There are also two new elements that are not technically sectioning elements. However, they are semantic and help divide a page:

- `<header>`
- `<footer>`

You can add a header or footer to any of the sectioning elements listed above, you can add them to an entire page, or you can add them to both. Listings 5.5 and 5.6 show you how a page written in HTML 4.01 with `<div>` tags as the divisions can be adjusted to use HTML5 and sectioning elements.
<title>Dandylions</title>
<body>
<div id="main">
    <h1>Dandylions</h1>
    <div id="article">
        <h2>Not Your Mother's Weed</h2>
        <p><span class="pronounce"><strong><em>Taraxacum</em></strong>/tærəˈkæskʌm/ is a large genus of flowering plants in the family Asteraceae. They are native to Eurasia and North and South America, and two species, <em>T. officinale</em> and <em>T. erythrospermum</em>, are found as weeds worldwide. Both species are edible in their entirety. The common name <strong>dandelion</strong>/'dændiˌlaŋ, from French dent-de-lion, meaning "lion's tooth") is given to members of the genus, and like other members of the Asteraceae family, they have very small flowers collected together into a composite flower head. Each single flower in a head is called a floret. Many <em>Taraxacum</em> species produce seeds asexually by apomixis, where the seeds are produced without pollination, resulting in offspring that are genetically identical to the parent plant.</p>
    </div>
    <div id="sidebar">
        <img src="images/dandy.jpg" width="400" height="300" alt="" />
    </div>
    <div id="footer">
    </div>
</div>
</body>
</html>

The HTML in Listing 5.5 has a headline, a main article with a sub-head titling it, a sidebar, and a footer. All these are defined in <div> elements. Listing 5.6 has these same elements, but it also uses HTML5 sectioning elements to define the sections semantically.

LISTING 5.6  The Converted HTML5 Document

<doctype html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Dandylions</title>
    </head>
    <body>
        <div id="main">
            <header>
                <h1>Dandylions</h1>
            </header>
        </div>
    </body>
</html>
Notice that `<div id="main">` is in both documents. This gives you a hook for styles and scripts on the entire web page.

**Considering Microformats**

When you’re adding semantics into your web pages, you should consider using microformats to add even more meaning, when appropriate. Microformats use human-readable text inside the HTML (usually in the `class` attribute of an element) to define the contents.

These are some commonly used microformats:

- hCalendar marks up events for putting in a calendar.
- hCard marks up contact information for address books.
- hRecipe marks up recipes for making food.
- hReview marks up reviews of books, movies, and more.
Microformats add semantic information about the elements, and this information is already being used in certain situations. Figure 5.2 shows a Google search for reviews of the movie *Ender’s Game*. The second and third results show “rich snippets,” including information like the star rating.

Google and Bing are both using these types of rich snippets to enhance their search results, and most of the data they are using to get it is semantically marked-up HTML using microformats. You can learn more about how to use microformats in my book *Sams Teach Yourself HTML5 Mobile Application Development in 24 Hours*.

By writing semantic HTML, you give more information to user agents to use to display the information correctly. For example, if a screen reader sees the `<article>` element, it knows that this is the main point of the page, and it will read it aloud before reading anything in an `<aside>` element. Plus, as web pages get more and more sophisticated, what the user agents do with them gains sophistication. For instance, in the future, your semantically marked-up recipe could tell a web-ready refrigerator what time to alert the robot butler to start the roast.

**Validating Your HTML**

It is tempting, once you’ve finished writing the HTML, to start immediately on the CSS. After all, plain HTML is very ugly, and making it pretty with CSS is the fun part! But you should not skip validation. By making sure your pages are valid HTML, you ensure that they are accessible and that your CSS and scripts will work as you intend.

Luckily validating your HTML is easy and takes only a moment. Simply go to [http://validator.w3.org](http://validator.w3.org) and fill in the URL of the page. If it’s not live yet, you can validate your HTML by uploading a file or pasting it in as direct input. If you’ve been careful, you should see a green indicator, as in Figure 5.3, that tells you the document was successfully checked as HTML5.
Summary

This hour you’ve learned the basics of HTML and how to create content so that it is ready to be converted to responsive design. You’ve learned about some HTML5 tags for content, layout, and page setup. Table 5.1 describes these tags and Table 5.2 describes the attributes. You’ve also learned how to create clean HTML and how to validate it.

This hour you’ve also learned how semantic HTML can help you create content with meaning. You’ve learned some HTML5 elements that are semantic as well as how to use microformats to add classes to your HTML to give the content even more semantic meaning. Table 5.3 lists the microformats and what they do.

### TABLE 5.1 HTML Elements Covered in Hour 5

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;a&gt;</td>
<td>A link or an anchor</td>
</tr>
<tr>
<td>&lt;abbr&gt;</td>
<td>An abbreviation or acronym</td>
</tr>
<tr>
<td>&lt;article&gt;</td>
<td>An article or a basic block of content that could be syndicated</td>
</tr>
<tr>
<td>&lt;aside&gt;</td>
<td>A related block of content</td>
</tr>
<tr>
<td>&lt;audio&gt;</td>
<td>A sound or audio file</td>
</tr>
<tr>
<td>&lt;body&gt;</td>
<td>The body of an HTML document</td>
</tr>
<tr>
<td>Element</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>&lt;br&gt;</td>
<td>Line break</td>
</tr>
<tr>
<td>&lt;cite&gt;</td>
<td>A citation</td>
</tr>
<tr>
<td>&lt;code&gt;</td>
<td>A code reference</td>
</tr>
<tr>
<td>&lt;div&gt;</td>
<td>A division of the HTML that doesn’t have semantic meaning</td>
</tr>
<tr>
<td>&lt;!doctype&gt;</td>
<td>The document type declaration</td>
</tr>
<tr>
<td>&lt;em&gt;</td>
<td>Emphasis</td>
</tr>
<tr>
<td>&lt;footer&gt;</td>
<td>The footer portion of a page or section</td>
</tr>
<tr>
<td>&lt;h1&gt; &lt;h2&gt; &lt;h3&gt;</td>
<td>Headline levels</td>
</tr>
<tr>
<td>&lt;head&gt;</td>
<td>The head or metadata of an HTML document</td>
</tr>
<tr>
<td>&lt;header&gt;</td>
<td>The header portion of a page or section</td>
</tr>
<tr>
<td>&lt;html&gt;</td>
<td>An HTML container element</td>
</tr>
<tr>
<td>&lt;img&gt;</td>
<td>An image or a graphic</td>
</tr>
<tr>
<td>&lt;meta charset&gt;</td>
<td>A meta character set declaration</td>
</tr>
<tr>
<td>&lt;nav&gt;</td>
<td>A navigation section</td>
</tr>
<tr>
<td>&lt;p&gt;</td>
<td>A paragraph</td>
</tr>
<tr>
<td>&lt;q&gt;</td>
<td>A short inline quotation</td>
</tr>
<tr>
<td>&lt;section&gt;</td>
<td>A semantically distinct section of content</td>
</tr>
<tr>
<td>&lt;span&gt;</td>
<td>An inline span of content</td>
</tr>
<tr>
<td>&lt;strong&gt;</td>
<td>Strong emphasis</td>
</tr>
<tr>
<td>&lt;title&gt;</td>
<td>The title of a web page</td>
</tr>
<tr>
<td>&lt;video&gt;</td>
<td>A video file</td>
</tr>
</tbody>
</table>

**TABLE 5.2** HTML Attributes Covered in Hour 5

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>A unique identifier</td>
</tr>
<tr>
<td>class</td>
<td>A classification attribute, which need not be unique</td>
</tr>
</tbody>
</table>

**TABLE 5.3** Microformats Covered in Hour 5

<table>
<thead>
<tr>
<th>Microformat</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hCalendar</td>
<td>Marks up events and dates for use in calendars</td>
</tr>
<tr>
<td>hCard</td>
<td>Marks up contact information</td>
</tr>
<tr>
<td>hRecipe</td>
<td>Marks up recipes</td>
</tr>
<tr>
<td>hReview</td>
<td>Marks up product and service reviews</td>
</tr>
</tbody>
</table>
Workshop

The workshop contains quiz questions to help you process what you’ve learned in this lesson. Try to answer all the questions before you read the answers.

Q&A

Q. This hour seems focused on writing HTML from scratch, but I like using a WYSIWYG editor. Will that work?

A. The best editors give you access to the HTML so that you can make sure there aren’t extra tags or attributes that you don’t want. But as long as you can edit the HTML directly, you shouldn’t have a problem using a visual editor rather than a text editor. Hour 21, “Tools for Creating Responsive Web Designs,” discusses some other tools, including editors, that you can use to create responsive designs.

Q. My system uses XHTML. Can I create RWD with that?

A. As long as you can add style sheets to your HTML or XHTML, you can use it to build RWD.

Q. You mention microformats, but I already use RDF or Microdata. Is that okay?

A. Using RDF and using Microdata are both acceptable ways to add semantics to your HTML. Microdata is a standard proposed by the W3C, and RDF is a more complicated system for defining metadata of all kinds.

Quiz

1. Is this well-formed HTML:
   <strong><em>My dog is</em></strong> big?

2. Is it better to use a <p> or a <div> tag when marking up a paragraph?

3. Is the <header> element semantic?

Answers

1. No, this is not well-formed HTML because the <strong> and <em> tags do not nest correctly. Designers commonly make the error of closing the outermost tag first. Here’s the correct version of this HTML:
   <em><strong>My dog is</strong> big</em>. 

2. When you’re marking up a paragraph, you should use the `<p>` tag because it defines paragraphs.

3. The `<header>` element is semantic because it describes the content as being the header section of a web page, section, or element.

**Exercise**

Go through the HTML for the site you are evaluating. Convert it as much as possible to semantic HTML5. Remove unnecessary tags and then validate it with an HTML5 validator. If the validator finds errors, fix them until the page is valid.
This page intentionally left blank
SYMBOLS

@import lines, placement in style sheets, 188
@import rule (CSS), 77
@media.info, media query tools, 295
<html> tag (HTML5), 56, 69
<a> tag (HTML5), 58, 68
<abbr> tag (HTML5), 63, 68
<article> tag (HTML5), 64, 68
<aside> tag (HTML5), 64, 68
<figure> tag (HTML5), 58, 68
<body> tag (HTML5), 56-57, 68
<br> tag (HTML5), 57, 69
<code> tag (HTML5), 63, 69
<datalist> tag (HTML5), 248, 253-254, 258
<div> tag (HTML5), 59-63, 69
<em> tag (HTML5), 58, 69
<footer> tag (HTML5), 64, 69
<h1> tag (HTML5), 57, 69
<h2> tag (HTML5), 57, 69
<h3> tag (HTML5), 57, 69
<head> tag (HTML5), 56, 69
<header> tag (HTML5), 64, 69
<html> tag (HTML5), 56, 69
<img> tag (HTML5), 56, 69
<input> tag (HTML5), 245-248, 252-254, 258
<label> tag (HTML5), 248
<meta charset> tag (HTML5), 56-57, 69
<meta name=viewport> tag (HTML5), 120-122
<nav> tag (HTML5), 64, 69
<p> tag (HTML5), 57, 69
<q> tag (HTML5), 63, 69
<section> tag (HTML5), 64, 69
<span> tag (HTML5), 59-61, 69
<strong> tag (HTML5), 58, 69
<title> tag (HTML5), 56-57, 69
<video> tag (HTML5), 58, 69, 215-216
!important keyword and cascade (CSS), 89
;(semicolon), CSS rules, 74

NUMBERS

51Degrees, device detection scripts, 311

A

A List Apart magazine, 334
RWD, 11
Switchy McLayout, 8
absolute font sizing, 193-194
absolute positioning, 22-23
left property, 21, 33
position property, 21, 33
position: relative property, 24
top property, 21, 33
Adaptive Images website image tools, 296
adaptive websites, 8
Adobe Edge Inspect website, testing tools, 299
advertising, 110
layout tables and, 237-240
RESS and, 329
RWD, problems with, 283-286
AGPL v3 (Affero General Public License), WURFL licensing concerns, 313
all media type (CSS), 126
analytics (websites)
  Google Analytics, 108, 111
  planning a responsive website, 107-108
and media query expression (CSS), 129
Andersson-Wise Architects website, 334
Android SDK emulator and website tests, 272
Apache Mobile Filter, device detection scripts, 311
Apple devices, portrait mode, 142
aspect-ratio media feature (CSS), 128-130
attribute selector (CSS), 75
attributes (HTML5)
  autocomplete, 248
  autofocus, 248
  class attribute, 62, 69
  id attribute, 61-62, 69
  novalidate, 249
  pattern, 249-250
  placeholder text, 248
  required, 249
autocomplete form attribute (HTML5), 248
autofocus form attribute (HTML5), 248
Axure website, wireframe tools, 293
B
back to top links and navigation, 177
background images
  changing, 205
  table-based layouts, 20
background-color property (CSS), 80, 91
Balsamiq website, wireframe tools, 293
banners (clickable) and tablets, 42
BBC News website, 342
Berners-Lee, Sir Tim, 48
best practices
  breakpoints, 336-338
  bullet points (numbered lists), 341
  content, 340-342
  costs, managing, 342-343
  flexible designs, 338-339
  fonts, 339
  images, 339-341
  layouts, 338
  links, 341
  minification, 339
  numbered lists (bullet points), 341
  user experience, 334-335
  video, 339
Blackberry emulator and website tests, 272
Bootstrap website, frameworks, 294
border property (CSS), 85, 91
borders
  columns, 159
  table-based layouts, 19
boxes, sizing, 173
braille media type (CSS), 126
break-after property (CSS), 162-163
break-before property (CSS), 162-163
break-inside property (CSS), 162-163
breakpoints (CSS). See also media queries (CSS)
  best practices, 136, 142, 336-338
  defining, 135-136
  media queries for larger screens, 141
  styles for all devices, 137-139
  styles for small screens, 139-140
fixed-width layouts, 151
limiting, 281
multiple breakpoints, 136-137
optimal breakpoints, 141-144
recommendations, 143-144
retina devices, 143
starting breakpoints, 143-144
testing
  device tests, 266-270
  HTML, 268-270
  resizing browser to test breakpoints, 266
widths of popular devices, 141-142
**browsers**

- cell phones (basic), 38
- CSS, browser prefixes, 50
- detection scripts, 30
  - mobile detection scripts, 31
  - object detection scripts, 31-32
- fonts, sizing for different browser widths, 188-189
- graded browser support (Yahoo!), 48
- mobile browsers, 8-11
- Opera Mini browser, 38
- prefixes, 27, 50
- tables, rearranging data according to browser, 229-231

**testing websites**

- local tests, 264
- resizing browser to test breakpoints, 266
- setting up a regimen, 265
- updating browsers, 264
- using multiple browsers, 264
- unsupported new features, 11
- updating, 264

**bullet points (numbered lists), best practices, 341**

- Bushell, David, 236
- buttons (toggle), 177

---

**C**

- cameras, 283
- Can I Use website, testing tools, 299

---

**Carnegie Mellon University and IoT, 118**

**cascade (CSS), 49**

- !important keyword, 89
- defining, 88-89
- media queries, fluid (liquid) layouts, 156
- overwriting styles, 141

**case-sensitivity, HTML5, 56**

**CDN (Content Delivery Network), 279**

**cell phones**

- basic cell phones
  - browsers, 38
  - features of, 37
  - support for, 38-39
- handheld media type (CSS), 126
- smartphones
  - form fields, 40
  - resolution, 39-40
  - support for, 40

**cells (tables)**

- layouts, 16
- resizing, 227-229

**centering elements via CSS, 84**

**ch measurement unit (CSS), 195**

**Champeon, Steven, 10**

**child selector (CSS), 74**

**City of Snohomish, Washington website, 343**

**class attribute (HTML5), 62, 69**

**class selector (CSS), 74**

**clean coding**

- HTML5, 62-63, 98
- XHTML, 63

**clear property (CSS), 26, 33, 85, 91**

**clickable banners and tablets, 42**

**client devices, building/testing to, 282**

**closing HTML5 tags, 58**

**Coca Cola and IoT, 118**

**coding**

- HTML5, clean code guidelines, 62-63, 98
- XHTML, clean code guidelines, 63

**CoffeeCup Responsive Layout Maker, editor tools, 302**

**CoffeeCup Web Form Builder, 298**

**Collison, Simon, 338**

**color**

- column-rule-color property (CSS), 161-163

**CSS**

- background-color property, 80, 91
- border colors, 85
- color media feature, 128-130
- color property, 80, 91
- color-index media feature, 128-130

**columns**

- column-count property (CSS), 27, 33, 161-163
- column-fill property (CSS), 162-163
- column-gap property (CSS), 27, 33
- column-min-width property (CSS), 27, 33
- column-rule-color property (CSS), 27, 34, 161-163
column-rule property (CSS), 27, 33, 162-163
column-rule-style property (CSS), 27, 34, 161-163
column-rule-width property (CSS), 27, 34, 161-163
column-span property (CSS), 27, 34, 162-163
column-width-policy property (CSS), 27, 33
column-width property (CSS), 27, 33, 161-163
columns property (CSS), 161-163
layouts, 158
   borders, 159
   CSS3, 161-162
   floats, 158-161
   three-column layouts, 159-161
two-column layouts, 159, 162
   width, 159-161
multicolumnn layouts (CSS), 27-29
table-based layouts, 19-20
two-column layouts, creating in CSS, 86-88
comma-separated list media query expression (CSS), 129
comments, adding in CSS, 76
contact forms
   creating, 254-258
   online tools, 298
containers (images), 201-204
content, best practices, 340-342
converting
   HTML 4.01 web pages to HTML5, 64-66
   images to retina-ready images, 209
cookies and data storage, RESS, 328
costs
   managing (best practices), 342-343
   negative impact of RWD, 282
Crafty Responsive Tables, 236
cropping images, 205-207
CSS (Cascading Style Sheets), 73
   @import lines, placement of, 188
   !important keyword and cascade, 89
   borders, color, 85oxes, sizing, 173
breakpoints. See also CSS, media queries
   best practices, 136, 142, 336-338
defining, 135-137
defining media queries for larger screens, 141
defining styles for every device, 137-139
defining styles for small screens, 139-140
fixed-width layouts, 151
   limiting, 281
multiple breakpoints, 136-137
optimal breakpoints, 141-144
recommendations, 143-144
retina devices, 143
starting breakpoints, 144
testing, 266-270
widths of popular devices, 141-142
browser prefixes, 50
cascade, 49!
   !important keyword, 89
defining, 88-89
fluid (liquid) layouts, 156
media queries, 156
overwriting styles, 141
ch measurement unit and sizing fonts, 195
color
   background-color property, 80, 91
   borders, 85
color property, 80, 91
comments, adding, 76
CSS 1, 7
CSS 2, 7
CSS3, 8, 336
   break-after property, 162-163
   break-before property, 162-163
   break-inside property, 162-163
   browser prefixes and column properties, 27
column-count property, 27, 33, 161-163
column-fill property, 162-163
column-gap property, 27, 33, 161-163
column-min-width property, 27, 33
column properties and browser prefixes, 27
column-rule property, 27, 33, 162-163
column-rule-color property, 27, 34, 161-163
column-rule-style property, 27, 34, 161-163
column-rule-width property, 27, 34, 161-163
column-span property, 27, 34, 162-163
column-width-policy property, 27, 33
column-width property, 27, 33, 161-163
columns property, 161-163
fonts and CSS3 measurement units, 195
four-column layouts, 27-29
layouts, 161-162
testing, 299
development of, 7-8
em measurement unit and sizing fonts, 188, 193
expressions (media queries)
and expression, 129
comma-separated list expression, 129
not expression, 130
only expression, 130
floats, columns in layouts, 158-161
fonts
adjusting in web pages, 80-83
font faces, 78
font-family property, 78, 91
font-size property, 79, 91
font stacks, 78
font-style property, 80, 91
font-weight property, 79, 91
importing, 185
loading, 185-187
fonts, sizing, 191-192
ch measurement unit, 195
CSS3 measurement units, 195
different browser widths, 188-189
em measurement unit, 188, 193
line-height property, 189
margin-bottom property, 190
margin-top property, 190
relative font sizing, 188
rem measurement unit, 195
vh measurement unit, 195
vmax measurement unit, 195
vmin measurement unit, 195
vw measurement unit, 195
forms
contact forms, 255-257
responsive forms, 255-257
images
background images, 205
cropping, 205-207
icons, 210
retina-ready images, 208-210
layouts
border property, 85, 91
break-after property, 162-163
break-before property, 162-163
break-inside property, 162-163
centering elements, 84
clear property, 85, 91
column-count property, 161-163
column-fill property, 162-163
column-gap property, 161-163
column-rule property, 162-163
column-rule-color property, 161-163
column-rule-style property, 161-163
column-rule-width property, 161-163
column-span property, 161-163
column-width property, 161-163
columns property, 161-163
CSS3, 161-162
fixed-width layouts, 151
fluid (liquid) layouts, 153
height property, 85, 91
margin property, 83-84, 91
padding property, 84, 91
three-column layouts, 160-161
two-column layouts, 86-88
width property, 85, 91
line-height property, 189
margin-bottom property, 190
margin-top property, 190
media features
  aspect-ratio, 128-130
color, 128-130
color-index, 128-130
device-aspect-ratio, 128-130
device-height, 128, 131
device-width, 128-131, 142
grid, 128, 131
height, 128, 131
monochrome, 128, 131
orientation, 128, 131
resolution, 128, 131
scan, 128, 131
width, 128-129, 131
media queries. See also
  breakpoints
  adding, 127
cascade, 156
  CSS Media Queries Test
  website, 295
  defining, 125
defining styles for larger
  screens, 141
  displaying video from
different sources, 218
  expressions, 129-130
  fixed-width layouts, 152
  fluid (liquid) layouts,
  155-156
loading fonts, 185-187
media features, 128-131
media types, 126-128
online tools, 295
retina-ready images, 209
select menus, 173
server requests and speed,
  128
testing, 295
media types
  all media type, 126
  best practices, 127-128
  braille media type, 126
  embossed media type, 126
  handheld media type, 126
  print media type, 126-127
  projection media type, 126
  screen media type, 126
  speech media type, 126
  tty media type, 126
tv media type, 126
minification, 298-299
navigation
  repositioning navigation bar,
  171-173
  resizing navigation bar,
  170-171
  select menus, 173-176
positioning layouts
  absolute positioning, 21-24
clear property, 26, 33
column-count property, 27, 33
column-gap property, 27, 33
column-min-width property,
  27, 33
column-rule property, 27, 33
column-rule-color property,
  27, 34
column-rule-style property,
  27, 34
column-rule-width property,
  27, 34
column-span property, 27, 34
column-width property, 27, 33
column-width-policy property,
  27, 33
floats, 24-26, 33
four-column layouts, 27-29
left property, 21, 33
multicolumn layouts, 27-29
position property, 21, 33
position: relative property,
  24
top property, 21, 33
precedence
  cascade and precedence,
  88-89
determining, 90
progressive enhancement,
  49-50
properties
  background-color property,
  80, 91
border property, 85, 91
break-after property, 162-163
break-before property, 162-163
break-inside property,
  162-163
clear property, 85, 91
color property, 80, 91
column-count property, 161-163
column-fill property, 162-163
column-gap property, 161-163
column-rule property, 162-163
column-rule-color property, 161-163
column-rule-style property, 161-163
column-rule-width property, 161-163
column-span property, 162-163
column-width property, 161-163
columns property, 161-163
font-family property, 78, 91
font-size property, 79, 91
font-style property, 80, 91
font-weight property, 79, 91
height property, 85, 91
margin property, 83-84, 91
padding property, 84, 91
width property, 85, 91
rem measurement unit and sizing fonts, 195
resets, 190
RESS and, 323-326
retina-ready images, 208-210
rules
   @import rule, 77
   precedence and cascade, 88-89
   semicolon (;), 74
   specificity, 89
   syntax, 74
   writing, 73
RWD, history of, 7-8
select menus, 173-176
selectors
   attribute selector, 75
   child selector, 74
   class selector, 74
   descendant selector, 74
element selector, 74
   ID selector, 74
   universal selector, 75
server requests and speed, 128
specificity, defining, 89-90
style sheets
   embedded style sheets, 76-77
   external style sheets, 77-78
   importing, 77
   linking to/from, 77
   managing, 78
   printing, 126-127
   web page optimization, 279
styles
   importance of styling everything, 141
   inline styles, 76-77
tables
   creating tables out of other tags, 239-240
   hiding data, 232-234
   layout tables and advertising, 237-238
   rearranging data according to browser, 230-231
   resizing, 228
   scrolling tables, 235
   unrecognized properties, 50
vh measurement unit and sizing fonts, 195
video
   displaying from different sources, 218
   HTML5 and, 216-221
   video backgrounds, 218-219
   YouTube video, 220-221
vmax measurement unit and sizing fonts, 195
vmin measurement unit and sizing fonts, 195
vw measurement unit and sizing fonts, 195
web page optimization, 279

data storage and cookies, RESS and, 328
degradation (graceful), 9-10, 47-48
descendant selector (CSS), 74
design changes, limiting, 281
design patterns, navigation, 168-169, 294-295
   alternative link locations to navigation menu, 178
   back to top links, 177
   fly-out menus, 177
   hidden navigation, 177-178
   navigation bar
      repositioning, 171-173
      resizing, 170-171
   select menus, 173-176
   toggle buttons, 177
DeviceAtlas, device detection scripts, 311
device-height media feature (CSS), 128, 131
device-width media feature (CSS), 128-131, 142
DNS lookups, web page optimization, 280
DOM elements (unobtrusive JavaScript), adding behavior to, 96
downloading
images, performance, 207
slow downloads and RWD impatient customers, 277
optimizing performance, 278-280
Dreamweaver as a web editor, 302
drop-down menus in forms, 252

E

editors
online editors, 302-303
professional design editors, 302
elastic layouts, 157
element selector (CSS), 74
em measurement unit (CSS), relative font sizing, 188, 193
email
design patterns, 295
e-mail types and login forms, 248
embedding
style sheets (CSS), 76-77
video, 215-216, 220
embossed media type (CSS), 126
emulators, website tests, 272-273
enhancement (progressive), 10, 47-48
benefits of, 50-51
content
content layer and HTML, 49
separating content from presentation and functionality, 48
styling content via CSS, 49-50
defining, 47-48
graded browser support (Yahoo!), 48
interactivity, 50
expenses, managing
best practices, 342-343
negative impact of RWD, 282
experience (users), best practices, 334-335
expressions (CSS media queries)
and expression, 129
comma-separated list expression, 129
not expression, 130
only expression, 130
external style sheets (CSS), 77-78
eyes and retina devices, 42
breakpoints (CSS), 143
building retina-ready websites, 43
planning for a responsive website, 110
reasons for building retina-ready websites, 43
retina-ready images, 207-210
video, 218

F

fallback fonts, 185
Family Crafts website, 338
fault tolerance and graceful degradation, 9-10, 47-48
feature detection scripts, 307
Modernizr
development version of, 312
geolocation support script, 312-313
HTML5 and, 312
installing, 311-312
Internet Explorer 8 and, 312
using, 312-313
website, 310-311
polyfills, 310
user agent strings, 308-309
Filament Group, 236
Firefox Responsive Design View, testing tools, 299-300
FitText font tool, 296
fixed-width layouts, 149, 154, 159
benefits of, 151
breakpoints, 151
CSS, 151
drawbacks of, 152
HTML3, 149-150
media queries, 152
flexible designs, best practices, 338-339
flexible-width images, 201-204

floats (CSS), 24, 26
columns in layouts, 158-161
float property (CSS), 25, 33
flow (normal) in HTML web pages, 25
fluid (liquid) layouts, 152, 159
benefits of, 153
cascade, 156
CSS, 153
drawbacks of, 154
fixed widths, 154
media queries, 155-156
percentage widths, 153
fly-out menus, 177
fonts
best practices, 339
choosing, 183
CSS
adjusting fonts on web pages, 80-83
font faces, 78
font stacks, 78
font-family property, 78, 91
font-size property, 79, 91
font-style property, 80, 91
font-weight property, 79, 91
CSS3 measurement units, 195
fallback fonts, 185
font stacks, 185
Google Fonts, 183
importing, 185
loading, 185-187
online tools, 296
sizing, 191-192
absolute font sizing, 193-194

ch measurement unit (CSS), 195
customer control of browser font size, 188
eM measurement unit (CSS), 188, 193
guidelines, 190
line-height property (CSS), 189
margin-bottom property (CSS), 190
margin-top property (CSS), 190
relative font sizing, 188, 193-194
rem measurement unit (CSS), 195
sizes for different browser widths, 188-189
small devices, 190
vh measurement unit (CSS), 195
vmax measurement unit (CSS), 195
vmin measurement unit (CSS), 195
vw measurement unit (CSS), 195
testing, 296
Food Sense website, 339
FooTable table tool, 298
Foresight.js image tool, retina devices, 297
forms
allowing customer input, 252-254
combining form fields, 252
contact forms, 254-258, 298
design guidelines for smaller
forms

devices
allowing customer input, 252-254
combining form fields, 252
drop-down menus, 252
removing extraneous page elements, 251
removing form fields, 252
simplicity, 251
drop-down menus, 252
form fields, 40
hiding passwords, 250

HTML5
<form data form tag, 248, 253-254, 258
<input form tag, 245-248, 252-254, 258
<label form tag, 248
autocomplete attribute, 248
autofocus attribute, 248
login form, 247, 250
novalidate attribute, 249
pattern attribute, 249-250
placeholder text attribute, 248
required attribute, 249
validating forms without scripting, 249-250
login forms, 247, 250
online tools, 298
removing
extraneous page elements, 251
form fields, 252
responsive forms, creating, 254-258
simplicity, 251
validating without scripting, 249-250
Formstack Responsive Web Form Builder, 298
Foundation website, frameworks, 294
four-column layouts (CSS), 27-29
frameworks, 293-294

G - H
geolocation, 283, 312-313
GIF, single-pixel GIF tricks, 6-7
Google
Google Analytics, 108, 111
Fonts, 183
graceful degradation, 9-10, 47-48
graded browser support (Yahoo!), 48
graphics, SVG and retina-ready
image creation, 210
grid media feature (CSS), 128, 131
handheld media type (CSS), 126
hCalendar microformat (HTML5), 66, 69
hCard microformat (HTML5), 66, 69
htaccess, retina device tools, 297
HTML3 (HyperText Markup Language version 3)
fixed-width layouts, 149-150
fonts
importing, 185
loading, 185-187
sizing, 194
images, 199
containers, 201-204
cropping, 205-207
flexible-width images, 201-204
links, mobile friendly
navigation, 167
navigation, 169, 173-175
select menus, 173-175
tables
hiding data, 232-234
rearranging data according to browser, 229-230
resizing cells, 227-228
scrolling tables, 235
HTML 4.01 (HyperText Markup Language version 4.01), converting web pages to HTML5, 64-66

HTML5 (HyperText Markup Language version 5), 55, 336
  absolute positioning layouts, 22-24
  attributes
    autocomplete, 248
    autofocus, 248
    class attribute, 62, 69
    id attribute, 61-62, 69
    novalidate, 249
    pattern, 249-250
    placeholder text, 248
    required, 249
  breakpoints, testing, 268-270
  case-sensitivity, 56
  clean code guidelines, 62-63, 98
  detection scripts, 31-32
  forms
    <datalist> form tag, 248, 253-254, 258
    <input> form tag, 245-248, 252-254, 258
    <label> form tag, 248
    autocomplete attribute, 248
    autofocus attribute, 248
    contact forms, 254-255
    form fields, 40
    login form, 247, 250
    novalidate attribute, 249
    pattern attribute, 249-250
    placeholder text attribute, 248
    required attribute, 249
  responsive forms, 254-255
  validating forms without scripting, 249-250
  frameworks, HTML5 Boilerplate website, 294
  hCalendar microformat, 66, 69
  hCard microformat, 66, 69
  highlighting text, 60-61
  hRecipe microformat, 66, 69
  hReview microformat, 66, 69
  HTML 4.01 web pages, converting, 64-66
  images
    image tools, 297
    scaling images, 280
    table-based layouts, 20
    web page optimization, 280
  JavaScript (unobtrusive), 96
  layout tags, 59-61
  microformats, 66, 69
  Modernizr feature detection tools and, 312
  normal flow in web pages, 25
  progressive enhancement, 49
  recommendation for, 49
  RESS and, 321-323, 326
  semantic elements, 49, 335
    new elements, 64-66
    standard elements, 63, 98
  style tags, 59-61
  table-based layouts, 17-18
    borders, 19
    columns, 19-20
    defining cells, 16
  images, 20
  rows, 19-20
  tags
    <!doctype> tag, 56, 69
    <a> tag, 58, 68
    <abbr> tag, 63, 68
    <article> tag, 64, 68
    <aside> tag, 64, 68
    <audio> tag, 58, 68
    <body> tag, 56-57, 68
    <br> tag, 57, 69
    <cite> tag, 63, 69
    <code> tag, 63, 69
    <div> tag, 59-63, 69
    <em> tag, 58, 69
    <footer> tag, 64, 69
    <h1> tag, 57, 69
    <h2> tag, 57, 69
    <h3> tag, 57, 69
    <head> tag, 56, 69
    <header> tag, 64, 69
    <html> tag, 56, 69
    <img> tag, 58, 69
    <meta charset> tag, 56-57, 69
    <meta name=viewport> tag, 120-122
    <nav> tag, 64, 69
    <p> tag, 57, 69
    <q> tag, 63, 69
    <section> tag, 64, 69
    <span> tag, 59-61, 69
    <strong> tag, 58, 69
    <title> tag, 56-57, 69
    <video> tag, 58, 69
  basic web content tags, 57-59
  class attribute, 62, 69
closing, 58
id attribute, 61-62, 69
initial-scale attribute, 121-122
layout tags, 59-61
maximum-scale attribute, 121-122
minimum-scale attribute, 121-122
overused tags, 63
required tags, 56
style tags, 59-61
user-scalable attribute, 121-122
viewport meta tag, 268-270
width attribute, 120-122
templates
adding content to, 58-59
basic template, 56
testing
breakpoints, 268-270
online tools, 299
websites, 265, 268-270
text, highlighting, 60-61
type attribute, 98
validating, 67
video
<video> tag, 215-216
background video, 218-219
CSS and, 216-221
embedding, 215-216
YouTube video, 220-221
web pages
normal flow, 25
optimization, 280
websites, testing, 265, 268-270
well-formed codes, 62-63, 98
well-formed HTML, 49
wireframe tools, Moquups
website, 293
HTTP requests, minimizing, 278
hybrid layouts, 157
IcoMoon website, font and image
tools, 296
icons, creating from retina-ready
images, 210
id attribute (HTML5), 61-62, 69
ID selector (CSS), 74
if statements (unobtrusive
JavaScript), 99
images, 199
background images, changing, 205
best practices, 341
changing displayed images, 204-205
containers, 201-204
converting to retina-ready
images, 209
cropping, 205-207
displaying, RESS, 328
download speeds, 207
resizing, 328, 339
resolution, 208
RESS and, 328
retina-ready images, 207
converting standard images
to, 209
creating icons, 210
creating images via SVG, 210
CSS media queries, 209
defining, 208
guidelines, 208
JavaScript, 210
optimizing, 208
resolution, 208
scaling images, web page
optimization, 280
single-pixel GIF tricks, 6-7
small web pages, best
practices, 339
table-based layouts, 20
updating on-the-fly, 208
importing
fonts, 185
style sheets, 77
initial-scale attribute (HTML5 tags), 121-122
inline styles (CSS), 76-77
installing
  Modernizr, 311-312
  WURFL device detection scripts, 313
interactivity and unobtrusive JavaScript, 50
Internet Explorer 8 and Modernizr feature detection tools, 312
Internet Images website, 341
iOS SDK emulator and website tests, 272
IoT (Internet of Things), 118
iPad Peek, website tests, 272
iPhone
  breakpoint tests, 266-267
  links, viewing, 167
  tables, viewing, 225
  website tests, 266-267
Irish, Paul, 173
Is This Retina? website, 297

JavaScript
detection scripts, 30, 307
  browser detection scripts, 30
  mobile detection scripts, 31
feature detection tools, 310-313
images
  image tools, 296
  retina-ready images, 210
JavaScript Compressor tool, 299
  minification, 298-299
Modernizr
development version of, 312
  geolocation support script, 312-313
  HTML5 and, 312
  installing, 311-312
  Internet Explorer 8 and, 312
  using, 312-313
  website, 310
  polyfills, 310
  progressive enhancement, 50
RESS and, 329
retina device tools, 297
testing
  online tools, 299
  websites, 264
unobtrusive JavaScript, 50
  adding behavior to DOM elements, 96
  defining, 95
  document.getElementById method, 100
  if statements, 99
  implementing, 98-100
  onclick events, 100
  onmouseover events, 100
  optimizing, 99
  security, 100
  separating structure from behavior, 95-96
  usability, 97
web page optimization, 279
jQuery
  font tools, 296
  FooTable table tool, 298
forms
  contact forms, 257-258
  responsive forms, 257-258
image tools, 297
navigation, 174
retina device tools, 297
select menus, 174
Stacktable.js table tool, 298

K - L
languages, Multi-Language Responsive Contact Form website, 298
layouts
  borders, 159
  columns
    floats, 158-161
    three-column layouts, 159-161
  two-column layouts, 159, 162
  width, 159-161
CSS layouts
  absolute layouts, 21-24
  border property, 85, 91
  break-after property, 162-163
  break-before property, 162-163
  break-inside property, 162-163
centering elements, 84
clear property, 26, 33, 85, 91
column-count property, 27, 33, 161-163
column-fill property, 162-163
column-gap property, 27, 33, 161-163
column-min-width property, 27, 33
column-rule property, 27, 33, 161-163
column-rule-color property, 27, 34, 161-163
column-rule-style property, 27, 34, 161-163
column-rule-width property, 27, 34, 161-163
column-span property, 27, 34, 162-163
column-width property, 27, 33, 161-163
columns property, 161-163
floats, 24-26, 33
four-column layouts, 27-29
height property, 85, 91
left property, 21, 33
margin property, 83-84, 91
multicolumn layouts, 27-29
padding property, 84, 91
position property, 21, 33
position: relative property, 24
top property, 21, 33
two-column layouts, 86-88
width property, 85, 91
defining, 147-148
editor tools, 302
elastic layouts, 157
fixed-width layouts, 159
benefits of, 151
breakpoints, 151
CSS, 151
drawbacks of, 152
HTML3, 149-150
media queries, 152
flexible designs, best practices, 338
fluid (liquid) layouts, 152, 159
benefits of, 153
cascade, 156
CSS, 153
drawbacks of, 154
fixed widths, 154
media queries, 155-156
percentage widths, 153
HTML5 layout tags, 59-61
hybrid layouts, 157
online tools, 302
table-based layouts, 15, 17-18, 158, 236
advertising, 237-240
borders, 19
columns, 19-20
defining cells, 16
images, 20
rows, 19-20
two-column layouts, creating in
CSS, 86-88
wireframes, 147-148
left property (CSS), 21, 33
Lettering.js font tool, 296
line-height property (CSS), 189
linking
back to top links and
navigation, 177
best practices, 341
navigation
alternative link locations to
navigation menu, 178
mobile friendly navigation, 167
style sheets, 77
web pages, id attribute
(HTML5), 62
liquid (fluid) layouts, 152, 159
benefits of, 153
cascade, 156
CSS, 153
drawbacks of, 154
fixed widths, 154
media queries, 155-156
percentage widths, 153
lists, comma-separated list
expression (CSS media queries),
129
loading fonts, 185-187
login forms, 247, 250
lookups (DNS), web page
optimization, 280

M

Macaw web editor website, 302
Marcotte, Ethan, 11
margin property (CSS), 83-84, 91
margin-bottom property (CSS), 190
margin-top property (CSS), 190
maximum-scale attribute (HTML5
tags), 121-122
media features (CSS)
aspect-ratio, 128-130
color, 128-130
color-index, 128-130
device-aspect-ratio, 128-130
device-height, 128, 131
device-width, 128-131, 142
grid, 128, 131
height, 128, 131
monochrome, 128, 131
orientation, 128, 131
resolution, 128, 131
scan, 128, 131
width, 128-129, 131

**media queries (CSS). See also breakpoints (CSS)**

adding, 127
cascade and fluid (liquid) layouts, 156
defining, 125
expressions, 129-130
fixed-width layouts, 152
fluid (liquid) layouts, 155-156
fonts, loading, 185-187
large screen devices, defining styles for, 141
media features, 128-131
Media Query Snippets website, 295
media types, 126-128
online tools, 295
retina-ready images, 209
select menus, 173
server requests and speed, 128
testing, 295
video, displaying different sources, 218

**media types (CSS)**

all media type, 126
best practices, 127-128
braille media type, 126
embossed media type, 126
handheld media type, 126
print media type, 126-127
projection media type, 126
screen media type, 126
speech media type, 126
tty media type, 126
tv media type, 126

**menus**

don-down menus in forms, 252
fly-out menus, 177
select menus, 173-176

**Meyer, Chris, 190**

**microformats (HTML5), 66, 69**

**minification, 298-299, 339**

**minimum-scale attribute (HTML5 tags), 121-122**

**mobile browsers, 8-11**

**mobile detection scripts, 31**

**mobile devices**

best practices, 334-335
built-in hardware, 283
cameras, 283
cell phones
  basic cell phones, 37-39
  smartphones, 39-40
geolocation, 283
links and mobile friendly navigation, 167
Mobile Device Rentals website, 271

**mobile first thinking, 115**

experimenting with new technologies, 117
focusing on content, 116-117
making a site mobile friendly, 118-121
success of, 117-118

**mobile friendly websites, 118**

user options, 119
zooming, 119-121

**mobile only design, 121**

**navigation (mobile friendly), 166**

horizontal scrolling, 168
hover navigation, 167
links, 167

**offline access, 283**

Popular Pages on Mobile (Google Analytics), 111
retina devices, 42
building retina-ready websites, 43
reasons for building retina-ready websites, 43

RWD, problems with, 282-283
tables
  clickable banners, 42
  form fields, 40
  resolution, 41
  support for, 41-42
user experience, 334-335
zooming, 119-121

**MobiReady website, testing tools, 299**

Mockflow website, wireframe tools, 293
mockups, wireframe tools, 292-293

Modernizr
development version of, 312
geolocation support script, 312-313
HTML5 and, 312
installing, 311-312
Internet Explorer 8 and, 312
using, 312-313
website, 310-311

monochrome media feature (CSS), 128, 131

Moqups website, wireframe tools, 293

mouse
onclick events (unobtrusive JavaScript), 100
onmouseover events (unobtrusive JavaScript), 100

moving navigation bar, 171-173

multicolumn layouts (CSS), 27-29
Multi-Language Responsive Contact Form website, 298

N

navigation
back to top links, 177
design patterns, 168-169
alternative link locations to navigation menu, 178
back to top links, 177
fly-out menus, 177
hidden navigation, 177-178
repositioning navigation bar, 171-173
resizing navigation bar, 170-171
select menus, 173-176
toggle buttons, 177
failures of RWD navigation, 165
fly-out menus, 177
hidden navigation, 177-178
horizontal scrolling, 168
hover navigation, 167
importance of RWD navigation, 165
links
alternative link locations to navigation menu, 178
mobile friendly design, 167
mobile friendly navigation, 166
horizontal scrolling, 168
hover navigation, 167
links, 167
navigation bar
repositioning, 171-173
resizing, 170-171
select menus, 173-176
toggle buttons, 177

negative reaction to RWD, 282-283

NetMarketShare.com website, Opera Mini browser, 38

normal flow in HTML web pages, 25
not media query expression (CSS), 130
novalidate form attribute (HTML5), 249
numbered lists (bullet points), best practices, 341

object detection scripts, 31-32
offline access of mobile devices, 283
onclick events (unobtrusive JavaScript), 100
online editors, 302-303
online resources
design tools
design patterns, 294-295
frameworks, 293-294
sketch sheets, 292
wireframe tools, 292-293
detection scripts, 31
Detect Mobile Browsers website, mobile detection scripts, 31
device detection tools, 311
editors
online editors, 302-303
professional design editors, 302
font tools, 296
form tools, 298
frameworks, 293-294
image tools, 296-297
media query tools, 295
optimization tools, 298-299
performance tools, 298-299
planning tools
design patterns, 294-295
frameworks, 293-294
sketch sheets, 292
wireframe tools, 292-293
planning a responsive website

RESS, 319
advertising and, 329
benefits of, 320
cookies and data storage, 328
CSS for basic RWD websites, 323-326
HTML for basic RWD websites, 321-323, 326
images and, 328
JavaScript and, 329
PHP and, 326-327, 330
server-side device detection, 326-327
uses for, 328-329
WURFL and, 326-329

sketch sheets, 292
sourceforge.net website and WURFL, 31
speed tools, 298-299
table tools, 297
testing tools, 299-301
text tools, 296
type tools, 296
wireframe tools, 292-293

only media query expression (CSS), 130
onmouseover events (unobtrusive JavaScript), 100

Opera browser
breakpoint tests, 266-267
document.getElementById method (unobtrusive JavaScript), 100
emulator and website tests, 272
website tests, 266-267

Opera Mini browser, 38

optimizing
images, retina-ready images, 208
performance, 278
CDN, 279
CSS, 279
DNS lookups, 280
JavaScript, 279
minimizing HTTP requests, 278
scaling images in HTML, 280
scripts, 279
speed tools, 298-299
style sheets, 279
retina-ready images, 208
style sheets (CSS), 78
unobtrusive JavaScript, 99
options (user) and mobile friendly design, 119
orientation media feature (CSS), 128, 131
overused HTML5 tags, 63

performance, optimizing
CDN, 279
CSS, 279
DNS lookups, 280
HTTP requests, minimizing, 278
images
downloading, 207
scaling in HTML, 280
JavaScript, 279
scripts, 279
speed tools, 298-299
style tools, 279

Photoshop, sketch sheets and RWD planning, 292

PHP (Hypertext Preprocessor)
detection scripts
browser detection scripts, 30
mobile detection scripts, 31
image tools, 296
RESS and, 326-327, 330
retina device tools, 297
testing websites, 264
WURFL and, 326-327, 330

Picturefill image tool, 297
Pixmobi Mobile Phone emulator and website tests, 272

placeholder text form attribute (HTML5), 248
plain text in login forms, 248

Planning a responsive website
advertising, 110
analytics, checking, 107-108
content
deciding, 110-111
evaluating, 109-110
planning a responsive website

flexible designs, best practices, 338-339
mobile devices, 115
experimenting with new technologies, 117
focusing on content, 116-117
making a site mobile friendly, 118-121
mobile only design, 121
success of, 117-118
need for planning
evaluating, 106-107
examples of, 105-106
planning tools
design patterns, 294-295
frameworks, 293-294
sketch sheets, 292
wireframe tools, 292-293
retina devices, 110
writing a plan, 112
polyfills, 310
Popular Pages on Mobile (Google Analytics), 111
portrait mode (Apple devices), 142
position property (CSS), 21, 33
position: relative property (CSS), 24
positioning
CSS layouts
absolute positioning, 21-24
clear property, 26, 33
column-count property, 27, 33
column-gap property, 27, 33
column-min-width property, 27, 33
column-rule property, 27, 33
column-rule-color property, 27, 34
column-rule-style property, 27, 34
column-rule-width property, 27, 34
column-span property, 27, 34
column-width property, 27, 33
column-width-policy property, 27, 33
position property, 21, 33
position: relative property, 24
top property, 21, 33
navigation bar, 171-173
precedence (CSS)
cascade and, 88-89
determining precedence, 90
printing
print media type (CSS), 126-127
style sheets (CSS), 126-127
prioritizing table data, 232
Prism emulator and website tests, 272
professional design editors, 302
progressive enhancement, 10
benefits of, 50-51
content
content layer and HTML, 49
separating content from presentation and functionality, 48
styling content via CSS, 49-50
defining, 47-48
graded browser support (Yahoo!), 48
interactivity, 50
Project Pencil website, wireframe tools, 293
projection media type (CSS), 126
properties (CSS)
background-color property, 80, 91
border property, 85, 91
break-after property, 162-163
break-before property, 162-163
break-inside property, 162-163
clear property, 85, 91
color property, 80, 91
column-count property, 161-163
column-fill property, 162-163
column-gap property, 161-163
column-rule property, 162-163
column-rule-color property, 161-163
column-rule-style property, 161-163
column-rule-width property, 161-163
column-span property, 162-163
column-width property, 161-163
columns property, 161-163
font-family property, 78, 91
font-size property, 79, 91
font-style property, 80, 91
font-weight property, 79, 91
height property, 85, 91
line-height property, 189
margin property, 83-84, 91
margin-bottom property, 190
margin-top property, 190
padding property, 84, 91
width property, 85, 91

PXtoEM font tool, 296

Q - R

rearranging table data according to browser, 229-231
relative font sizing, 188, 193-194
relative position property (CSS). See position: relative property (CSS)
rem measurement unit (CSS), 195
renting devices for website testing, 271
required form attribute (HTML5), 249
resets (CSS), 190
resizing
boxes, 173
cells (tables), 227-229
images, 328, 339
navigation bar, 170-171

resolution
resolution media feature (CSS), 128, 131
retina-ready images, 208
smartphones, 39-40
tables, 41

resources (online)
design tools
design patterns, 294-295
frameworks, 293-294
sketch sheets, 292
wireframe tools, 292-293
detection scripts, 31
device detection, 311
editors
online editors, 302-303
professional design editors, 302
font tools, 296
form tools, 298
frameworks, 293-294
image tools, 296-297
media query tools, 295
optimization tools, 298-299
performance tools, 298-299
planning tools
design patterns, 294-295
frameworks, 293-294
sketch sheets, 292
wireframe tools, 292-293
RESS, 319
advertising and, 329
benefits of, 320
cookies and data storage, 328
CSS for basic RWD websites, 323-326
HTML for basic RWD websites, 321-323, 326
images and, 328
JavaScript and, 329
PHP and, 326-327, 330
server-side device detection, 326-327
uses for, 328-329
WURFL and, 326-329
speed tools, 298-299
tables
responsive scripting, 236
table tools, 297
testing tools, 299-300
text tools, 296
type tools, 296
wireframe tools, 292-293
Responsive Design View (Firefox), testing tools, 299-300
Responsive Email Patterns website, design patterns, 295
Responsive Grid System website, frameworks, 294
Responsive Patterns website, design patterns, 295
Responsive Tables 2 website, 236
RESS (Responsive Wed Design + Server Side Components), 319
advertising and, 329
basic RWD websites
CSS, 323-326
HTML, 321-323, 326
benefits of, 320
cookies and data storage, 328
images and, 328
JavaScript and, 329
PHP and, 326-327, 330
server-side device detection, 326-327
uses for, 328-329
WURFL and, 326-329

retina devices, 42
breakpoints (CSS), 143
image tools, 297
planning for a responsive website, 110
retina-ready images, 207
  converting standard images to, 209
  creating icons, 210
  creating images via SVG, 210
CSS media queries, 209
defining, 208
guidelines, 208
JavaScript, 210
Retina Images website, 297
retina-ready websites, 43
video, 218

rows (tables), table-based layouts, 19-20
rules (CSS)
  @import rule, 77
  precedence and cascade, 88-89
  semicolon (;), 74
  specificity, 89
  syntax, 74
  writing, 73

RWD (Responsive Web Design)
  best practices
  breakpoints, 336-338
  bullet points (numbered lists), 341
  content, 340-342
  costs, 342-343
  flexible designs, 338-339
  fonts, 339
  images, 339-341
  layouts, 338
  links, 341
  minification, 339
  mobile devices, 334-335
  numbered lists (bullet points), 341
  simplicity, 335
  video, 339
client devices, building/testing to, 282
defining, 5
design tools
  design patterns, 294-295
  frameworks, 293-294
  sketch sheets, 292
  wireframe tools, 292-293
expertise, importance of, 281
history of, 5, 11
adaptive websites, 8
CSS, 7-8
graceful degradation, 9-10
mobile browsers, 8-10
progressive enhancement, 10
  single-pixel GIF trick, 6-7
importance of, 44

mobile devices
  cell phones (basic), 37-39
  mobile first design, 115-121
  mobile friendly websites, 118-121
  mobile only design, 121
  retina devices, 42-43
  smartphones, 39-40
  tablets, 40-42
need for, 11
planning
  advertising, 110
  checking website analytics, 107-108
  deciding on content, 110-111
  evaluating content, 109-110
  evaluating need for a responsive website, 106-107
  examples for the need for planning, 105-106
  mobile devices, 115-121
  retina devices, 110
  writing a site plan, 112
planning tools
  design patterns, 294-295
  frameworks, 293-294
  sketch sheets, 292
  wireframe tools, 292-293
problems with
  advertising, 283-286
  designer workload, 280-281
  expense, 282
  impatient customers, 277
  mobile site expectations, 282-283
negative reaction to RWD, 282-283
optimizing performance, 278-280
slow downloads, 277-280
user experience, 334-335
RWD Table Patterns website, design patterns, 295

sizing

boxes, 173
cells (tables), 227-229
CSS fonts, 79, 91
fonts, 191-192
absolute font sizing, 193-194
ch measurement unit (CSS), 195
CSS3 measurement units, 195
customer control of browser font size, 188
different browser widths, 188-189
ewm measurement unit (CSS), 188, 193
guidelines, 190
line-height property (CSS), 189
margin-bottom property (CSS), 190
margin-top property (CSS), 190
relative font sizing, 188, 193-194
rem measurement unit (CSS), 195
small devices, 190
vh measurement unit (CSS), 195
vmax measurement unit (CSS), 195
vmin measurement unit (CSS), 195
vw measurement unit (CSS), 195
images, 328, 339
navigation bar, 170-171

tables, responsive scripting resources, 236
testing, online tools, 299
web page optimization, 279

scrolling

horizontal scrolling, mobile friendly navigation, 168
tables, 235

security

passwords, hiding in forms, 250
unobtrusive JavaScript, 100
user agent strings, 308-310

select menus, 173-176
selectors (CSS)
attribute selector, 75
child selector, 74
class selector, 74
descendant selector, 74
element selector, 74
ID selector, 74
universal selector, 75

semantic elements (HTML5), 49, 335
new elements, 64-66
standard elements, 63, 98

servers

CSS requests and speed, 128
server-side device detection and WURFL, 326-327
testing websites, 264

Sharp, Remy, 310

simplicity, user experience, 335

single-pixel GIF trick, 6-7

S

scaling images, web page optimization, 280
scan media feature (CSS), 128, 131
ScientiaMobile, WURFL licensing concerns, 313
screen media type (CSS), 126
scripts
detection scripts
browser detection scripts, 30
mobile detection scripts, 31
object detection scripts, 31-32
device detection scripts, 307, 310-311
online tools, 311
WURFL, 311-315, 326-327
feature detection scripts, 307
Modernizr, 310-313
polyfills, 310
user agent strings, 308-309
geolocation support script and Modernizr, 312-313
minification, 298-299
polyfills, 310

sizing

boxes, 173
cells (tables), 227-229
CSS fonts, 79, 91
fonts, 191-192
absolute font sizing, 193-194
ch measurement unit (CSS), 195
CSS3 measurement units, 195
customer control of browser font size, 188
different browser widths, 188-189
em measurement unit (CSS), 188, 193
guidelines, 190
line-height property (CSS), 189
margin-bottom property (CSS), 190
margin-top property (CSS), 190
relative font sizing, 188, 193-194
rem measurement unit (CSS), 195
small devices, 190
vh measurement unit (CSS), 195
vmax measurement unit (CSS), 195
vmin measurement unit (CSS), 195
vw measurement unit (CSS), 195
images, 328, 339
navigation bar, 170-171

tables, responsive scripting resources, 236
testing, online tools, 299
web page optimization, 279

scrolling

horizontal scrolling, mobile friendly navigation, 168
tables, 235

security

passwords, hiding in forms, 250
unobtrusive JavaScript, 100
user agent strings, 308-310

select menus, 173-176
selectors (CSS)
attribute selector, 75
child selector, 74
class selector, 74
descendant selector, 74
element selector, 74
ID selector, 74
universal selector, 75

semantic elements (HTML5), 49, 335
new elements, 64-66
standard elements, 63, 98

servers

CSS requests and speed, 128
server-side device detection and WURFL, 326-327
testing websites, 264

Sharp, Remy, 310

simplicity, user experience, 335

single-pixel GIF trick, 6-7

S

scaling images, web page optimization, 280
scan media feature (CSS), 128, 131
ScientiaMobile, WURFL licensing concerns, 313
screen media type (CSS), 126
scripts
detection scripts
browser detection scripts, 30
mobile detection scripts, 31
object detection scripts, 31-32
device detection scripts, 307, 310-311
online tools, 311
WURFL, 311-315, 326-327
feature detection scripts, 307
Modernizr, 310-313
polyfills, 310
user agent strings, 308-309
geolocation support script and Modernizr, 312-313
minification, 298-299
polyfills, 310

sizing

boxes, 173
cells (tables), 227-229
CSS fonts, 79, 91
fonts, 191-192
absolute font sizing, 193-194
ch measurement unit (CSS), 195
CSS3 measurement units, 195
customer control of browser font size, 188
different browser widths, 188-189
em measurement unit (CSS), 188, 193
guidelines, 190
line-height property (CSS), 189
margin-bottom property (CSS), 190
margin-top property (CSS), 190
relative font sizing, 188, 193-194
rem measurement unit (CSS), 195
small devices, 190
vh measurement unit (CSS), 195
vmax measurement unit (CSS), 195
vmin measurement unit (CSS), 195
vw measurement unit (CSS), 195
images, 328, 339
navigation bar, 170-171
sketch sheets, RWD planning, 292
slow downloads and RWD
  impatient customers, 277
  optimizing performance, 278
  CDN, 279
  CSS, 279
  DNS lookups, 280
  JavaScript, 279
  minimizing HTTP requests, 278
  scaling images in HTML, 280
  scripts, 279
  style sheets, 279
small web pages, best practices, 338-339
smartphones
  form fields, 40
  handheld media type (CSS), 126
  iPhone
    breakpoint tests, 266-267
    viewing links, 167
    viewing tables, 225
    website tests, 266-267
  resolution, 39-40
  support for, 40
SmartPlanet website, mobile first design, 116
Snohomish Aquatic Center website, 337
sourceforge.net website and WURFL, 31
specificity (CSS), 89-90
speech media type (CSS), 126
speeding up (optimizing) performance
  CDN, 279
  CSS, 279
  DNS lookups, 280
  JavaScript, 279
  minimizing HTTP requests, 278
  scaling images in HTML, 280
  scripts, 279
  style sheets, 279
speed tools, 298-299
spooled user agent strings in Safari, 310
Squarespace online editor website, 302
stacking fonts, 78, 185
Stacktable.js table tool, 298
StatCounter website, mobile first design, 115
storing data via cookies, RESS, 328
style sheets (CSS)
  @import lines, placement of, 188
  cascade, overwriting styles, 141
  embedded style sheets, 76-77
  external style sheets, 77-78
  importing, 77
  linking to/from, 77
  managing, 78
  printing, 126-127
  web page optimization, 279
style tags (HTML5), 59-61
styles (CSS), inline styles, 76-77
SVG (Scalable Vector Graphics) and retina-ready image creation, 210
Switchy McLayout, 8
syntax, CSS rules, 74

T
tables and table-based layouts, 15, 17-18
  advertising, 237-240
  borders, 19
  cells
    defining, 16
    resizing, 227-229
  columns, 19-20
  hiding data, 232-234
  images, 20
  online tools, 297
  prioritizing data, 232
  rearranging data according to browser, 229-231
  responsive scripting resources, 236
  responsiveness of, 226
  rows, 19-20
  scrolling tables, 235
  smaller screens, 225
  tablePress table tool, 298
Tablets
  clickable banners, 42
  form fields, 40
  resolution, 41
  support for, 41-42
tags (HTML5)

<!doctype> tag, 56, 69
<a> tag, 58, 68
<abbr> tag, 63, 68
<article> tag, 64, 68
<aside> tag, 64, 68
<audio> tag, 58, 68
<body> tag, 56-57, 68
<br> tag, 57, 69
<cite> tag, 63, 69
<code> tag, 63, 69
<div> tag, 59-63, 69
<em> tag, 58, 69
<footer> tag, 64, 69
<h1> tag, 57, 69
<h2> tag, 57, 69
<h3> tag, 57, 69
<header> tag, 56, 69
<header> tag, 64, 69
<html> tag, 56, 69
<img> tag, 58, 69
<meta charset> tag, 56-57, 69
<meta name=viewport> tag, 120-122
<nav> tag, 64, 69
<p> tag, 57, 69
<q> tag, 63, 69
<section> tag, 64, 69
<span> tag, 59-61, 69
<strong> tag, 58, 69
<title> tag, 56-57, 69
<video> tag, 58, 69
basic web content tags, 57-59
class attribute, 62, 69
closing, 58
id attribute, 61-62, 69
initial-scale attribute, 121-122
layout tags, 59-61
maximum-scale attribute, 121-122
minimum-scale attribute, 121-122
overused tags, 63
required tags, 56
style tags, 59-61
user-scalable attribute, 121-122
viewport meta tag, 268-270
width attribute, 120-122
Technology Review website, graceful degradation, 48
templates
HTML5 templates
adding content to, 58-59
basic template, 56
online tools, 303

testing
breakpoints
device tests, 266-270
HTML5, 268-270
browser tests
local tests, 264
resizing browser to test breakpoints, 266
setting up a regimen, 265
updating browsers, 264
using multiple browsers, 264
client devices, 282
CSS3, online tools, 299
emulators, 272-273
fonts, 296
HTML5
breakpoints, 268-270
editors, 265
online tools, 299
JavaScript, 264, 299
limiting, 281
media queries, 295
online tools, 299-301
PHP, 264
text, 296
type, 296
web pages, 281
websites, 263
customer assistance, 273
renting devices, 271
server tests, 264
testing without devices, 271-273
WYSIWYG editor tests, 264
text
best practices, 339
fonts
absolute font sizing, 193-194
ch measurement unit (CSS), 195
choosing, 183
CSS3 measurement units, 195
customer control of browser font size, 188
em measurement unit (CSS), 188, 193
fallback fonts, 185
font stacks, 185
Google Fonts, 183
importing, 185
loading, 185-187
relative font sizing, 188, 193-194
rem measurement unit (CSS), 195
sizing, 188-192
vh measurement unit (CSS), 195
vmax measurement unit (CSS), 195
vmin measurement unit (CSS), 195
vw measurement unit (CSS), 195
highlighting text, HTML5, 60-61
online tools, 296
plain text in login forms, 248
testing, 296
three-column layouts, 159-161
toggle buttons, 177
tools (online)
design tools
development patterns, 294-295
frameworks, 293-294
sketch sheets, 292
wireframe tools, 292-293
device detection tools, 311
editors
online editors, 302-303
professional design editors, 302
font tools, 296
form tools, 298
frameworks, 293-294
image tools, 296-297
media query tools, 295
optimization tools, 298-299
performance tools, 298-299
planning tools
design patterns, 294-295
frameworks, 293-294
sketch sheets, 292
wireframe tools, 292-293
RESS, 319
advertising and, 329
benefits of, 320
cookies and data storage, 328
CSS for basic RWD websites, 323-326
HTML for basic RWD websites, 321-326
images and, 328
JavaScript and, 329
PHP and, 326-327, 330
server-side device detection, 326-327
uses for, 328-329
WURFL and, 326-329
sketch sheets, 292
speed tools, 298-299
table tools, 297
testing tools, 299-301
text tools, 296
type tools, 296
wireframe tools, 292-293
top property (CSS), 21, 33
tty media type (CSS), 126
tv media type (CSS), 126
two-column layouts, 86-88, 159, 162
type. See fonts
universal selector (CSS), 75
unobtrusive JavaScript, 50
defining, 95
document.getElementById method, 100
DOM elements, adding behavior to, 96
if statements, 99
implementing, 98-100
onclick events, 100
onmouseover events, 100
optimizing, 99
security, 100
separating structure from behavior, 95-96
usability, 97
unrecognized properties in CSS, 50
unsupported new features in browsers, 11
updating
browsers, 264
images on-the-fly, 208
usability and unobtrusive JavaScript, 97
user agent strings and feature detection scripts, 308-309
user experience, best practices, 334-335
user options and mobile friendly design, 119
user-scalable attribute (HTML5 tags), 121-122
validating
email types in login forms, 248
forms without scripting, 249-250
HTML5, 67
vh measurement unit (CSS), 195
video
CSS and
displaying video from
different sources, 218
HTML5, 216-221
video backgrounds, 218-219
YouTube video, 220-221
different sources, displaying
video from, 218
embedding, 215-216, 220
HTML5
<video> tag, 215-216
background video, 218-219
CSS and, 216-221
embedding video, 215-216
YouTube video, 220-221
retina devices, 218
small web pages, best
practices, 339
YouTube video, 219-221
viewport meta tag (HTML5), 268-270
vmax measurement unit (CSS), 195
vmin measurement unit (CSS), 195
vw measurement unit (CSS), 195

web editors
online editors, 302-303
professional design editors, 302
web fonts. See fonts
web forms. See forms
web layouts. See layouts
web pages
best practices
flexible designs, 338-339
small web pages, 338-339
clickable banners and tablets, 42
content, best practices,
340-342
design changes, limiting, 281
flexible designs, best practices,
338-339
linking
alternative link locations to
navigation menu, 178
id attribute (HTML5), 62
navigation
alternative link locations to
navigation menu, 178
back to top links, 177
design patterns, 168-178
failures of RWD navigation, 165
fly-out menus, 177
hidden navigation, 177-178
horizontal scrolling, 168
hover navigation, 167
importance of RWD
navigation, 165
mobile friendly navigation, 166-168
repositioning navigation bar, 171-173
resizing navigation bar, 170-171
select menus, 173-176
toggle buttons, 177
normal flow, 25
optimizing performance
CDN, 279
CSS, 279
DNS lookups, 280
JavaScript, 279
minimizing HTTP requests, 278
scaling images in HTML, 280
scripts, 279
style sheets, 279
select menus, 173-176
small web pages, best
practices, 338-339
testing
customer assistance, 273
limiting tests, 281
renting devices, 271
server tests, 264
testing without devices,
271-273
WYSIWYG editor tests, 264
WebOS emulator and website
tests, 272
websites
adaptive websites, 8
Adobe Edge Inspect website,
testing tools, 299
analytics
    Google Analytics, 108
    planning a responsive website, 107-108
Andersson-Wise Architects website, 334
Axure website, wireframe tools, 293
Balsamiq website, wireframe tools, 293
BBC News website, 342
Bootstrap website, frameworks, 294
Can I Use website, testing tools, 299
City of Snohomish, Washington website, 343
CSS Media Queries Test website, 295
design changes, limiting, 281
Design Week Portland website, 335
Detect Mobile Browsers website, mobile detection scripts, 31
Family Crafts website, 338
Food Sense website, 339
Foundation website, frameworks, 294
graceful degradation, 9-10, 47-48
HTML5
    HTML5 Boilerplate website, 294
    semantic HTML, 49
    well-formed HTML, 49
IcoMoon website, font and image tools, 296
Internet Images website, 341
Is This Retina? website, 297
Macaw web editor website, 302
Media Query Snippets website, 295
Mobile Device Rentals website, 271
mobile devices
    mobile first design, 115-121
    mobile friendly websites, 118-121
    mobile only design, 121
MobiReady website, testing tools, 299
Mockflow website, wireframe tools, 293
Modernizr website, 310-311
Moqups website, wireframe tools, 293
Multi-Language Responsive Contact Form website, 298
NetMarketShare.com website, Opera Mini browser, 38
optimizing performance, 278
    CDN, 279
    CSS, 279
    DNS lookups, 280
    JavaScript, 279
    minimizing HTTP requests, 278
    scaling images in HTML, 280
    scripts, 279
    style sheets, 279
Pixmobi Mobile Phone emulator and website tests, 272
planning
    advertising, 110
    checking website analytics, 107-108
deciding on content, 110-111
evaluating content, 109-110
evaluating need for a responsive website, 106-107
examples for the need for planning, 105-106
mobile devices, 115-121
retina devices, 110
writing a site plans, 112
Prism emulator and website tests, 272
progressive enhancement, 10
    benefits of, 50-51
content layer and HTML, 49
content separation from presentation and functionality, 48
CSS and content stylization, 49-50
HTML and content layer, 49
interactivity, 50
styling content via CSS, 49-50
Project Pencil website, wireframe tools, 293
Responsive Email Patterns website, design patterns, 295
Responsive Grid System website, frameworks, 294
Responsive Patterns website, design patterns, 295
Responsive Tables 2 website, 236
retina-ready websites
   building, 43
   Retina Images website, 297
RWD Table Patterns website, design patterns, 295
SmartPlanet website, mobile first design, 116
Snohomish Aquatic Center website, 337
Squarespace online editor website, 302
StatCounter website, mobile first design, 116
Technology Review website, graceful degradation, 48
testing, 263
   breakpoints, 266-270
   browser tests, 264-266
   customer assistance, 273
   emulators, 272-273
   HTML editors, 265
   JavaScript, 264
   PHP, 264
   renting devices, 271
   server tests, 264
   without devices, 271-273
   WYSIWYG editor tests, 264
WebOS emulator and website tests, 272
Weebly online editor website, 303
Windows Phone Developer Tools emulator and website tests, 272
well-formed codes
   HTML5, 49, 62-63, 98
   XHTML, 63
width
   breakpoints (CSS), 141-142, 151
   columns, 159-160
      column-rule-width property (CSS), 161-163
      column-width property (CSS), 161-163
   fixed-width layouts, 159
      benefits of, 151
      breakpoints, 151
      CSS, 151
      drawbacks of, 152
      HTML3, 149-150
      media queries, 152
   flexible-width images, 201-204
   fluid (liquid) layouts, 152, 159
      fixed widths, 154
      percentage widths, 153
   fonts, sizing for different browser widths, 188-189
   layouts, 159-161
   percentage widths, fluid (liquid) layouts, 153
   width attribute (HTML5 tags), 120-122
   width media feature (CSS), 128-131
   width property (CSS), 85, 91
   zooming, 119-121
Zurb Responsive Tables website, 297
Weebly online editor website, 303
Windows Phone Developer Tools emulator and website tests, 272
wireframes (layouts), 147-148, 292-293
Wix online editor website, 303
WordPress, 298
workload (designers), problems with RWD, 280-281
Wroblewski, Luke, 319
WURFL (Wireless Universal Resource FiLe), 31
   freeware/licensing concerns, 313
   installing, 313
   RESS and, 326-329
   server-side device detection, 326-327
   using, 315
   website, 311
WYSIWYG (What You See Is What You Get) editors, testing websites, 264
X - Y - Z
XHTML (Extensible HyperText Markup Language), clean code guidelines, 63
Yahoo! website, graded browser support, 48
YouTube video, 219-221
YSlow analyzer, speed (optimization) tool, 299
zooming, 119-121
Zurb Responsive Tables website, 297
FOR A LIMITED TIME, we are offering readers of *Sams Teach Yourself* books a 50% OFF discount to ANY online Learning Lab through Dec 15, 2015.

Visit [informit.com/learninglabs](http://informit.com/learninglabs) to see available labs, try out full samples, and order today.

- **Read** the complete text of the book online in your web browser
- **Watch** an expert instructor show you how to perform tasks in easy-to-follow videos
- **Try** your hand at coding in an interactive code-editing sandbox in select products
- **Test** yourself with interactive quizzes