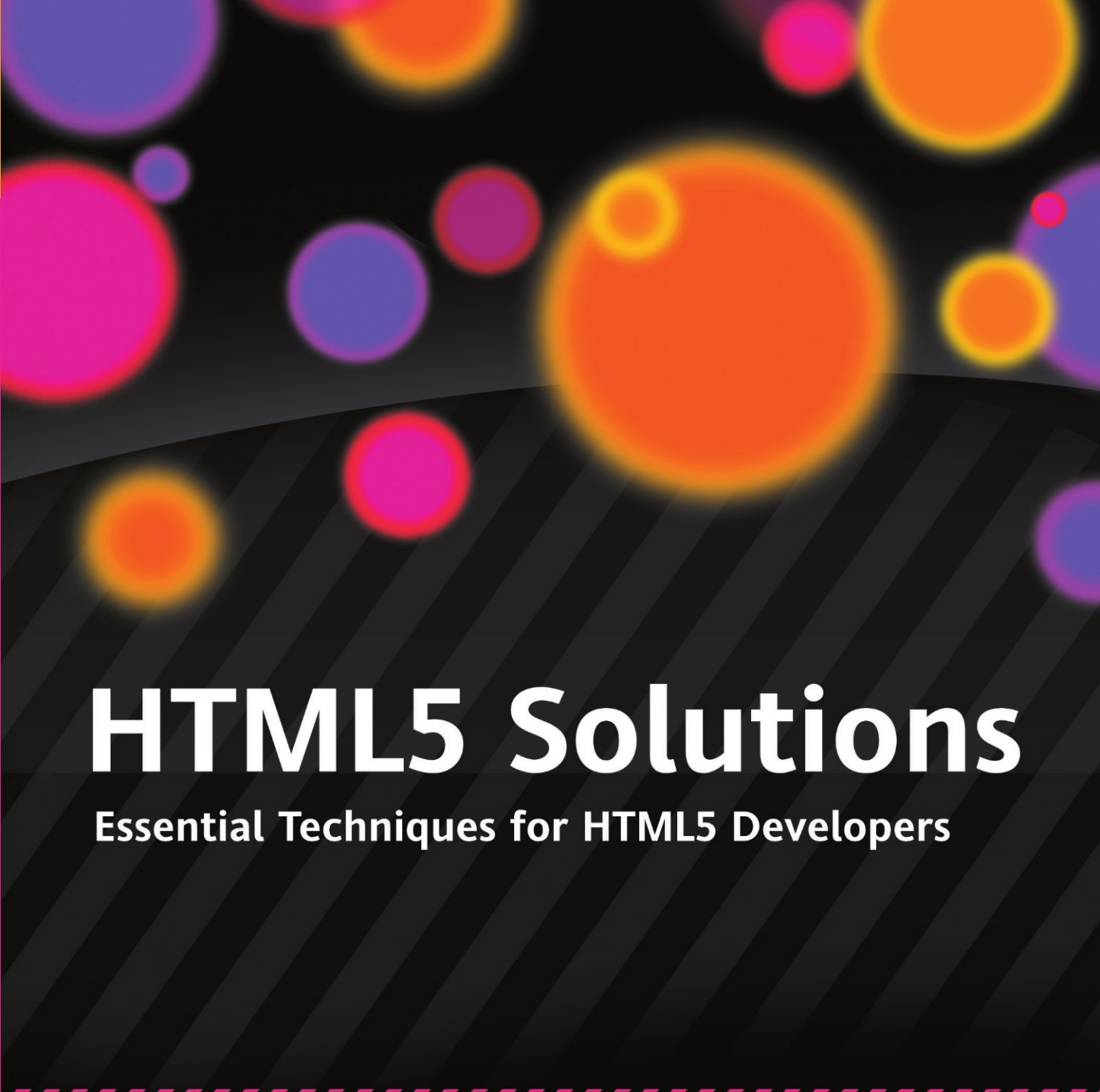


HTML5 Solutions

Essential Techniques for HTML5 Developers

MARCO CASARIO, PETER ELST,
CHARLES BROWN, NATHALIE WORMSER,
AND CYRIL HANQUEZ

friendsof
an Apress® company

HTML5 Solutions: Essential Techniques for HTML5 Developers

Marco Casario, Peter Elst, Charles Brown, Nathalie Wormser, and Cyril Hanquez

HTML5 SOLUTIONS: ESSENTIAL TECHNIQUES FOR HTML5 DEVELOPERS

ISBN-13 (pbk): 978-1-4302-3386-2

ISBN-13 (electronic): 978-1-4302-3387-9

Distributed to the book trade worldwide by Springer Science+Business Media LLC, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com.

For information on translations, please e-mail rights@apress.com or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at www.apress.com/bulk-sales

The source code for this book is freely available to readers at www.friendsofed.com in the Downloads section.

Credits

President and Publisher: Paul Manning

Lead Editor: Ben Renow-Clarke

Technical Reviewer: Jason Nadon

Editorial Board: Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Jonathan Gennick, Jonathan Hassell, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Frank Pohlmann, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Coordinating Editor: Jennifer L. Blackwell

Copy Editor: Seth Kline

Compositor: Mary Sudul

Indexer: Toma Mulligan

Artist: April Milne

Cover Artist: Anna Ishchenko

Cover Designer: Corné van Dooren

To the memory of my grandmother Maria

—Marco Casario

In loving memory of those who continue to inspire me

—Peter Elst

To S.P.

—Nathalie Wormser

To my wife for her love and support and my lovely children for their patience

—Cyril Hanquez

Contents at a Glance

Contents

About the Authors

Marco Casario has been passionate about informatics since he was little more than a child and used to program games in BASIC for the Commodore 64 before dedicating himself, while still very young, to innovative projects for the Web using HTML, JavaScript, Flash, and Director.

In 2001, he began to collaborate with Macromedia. Since that year, he has produced and headed a long series of presentations, conferences, and articles, which you can find listed in detail in his blog (`casario.blogs.com`).

In 2005, Marco has founded Comtaste (`www.comtaste.com`), a company dedicated to exploring new frontiers in Rich Internet and Mobile Applications and the convergence between the Web and the world of mobile devices. Now his focus is on User Experience (UX) aspects in order to provide users with a valuable experience. He also is a proponent of cloud computing with Google Apps APIs and the Google App Engine.

Marco is founder of the biggest worldwide Flash Lite UG and of `www.augitaly.com`, a reference point for the Italian community of Adobe users in which he serves in the role of Channel Manager for the section dedicated to Flex (`www.augitaly.com/flexgala`.). Recently, he became Community Manager for the Google Technology User Group (`www.gtug.it`) in Milan.

Marco is an Adobe Certified Instructor for Flex 4, LCDS 3, and AIR (ACI), and an Adobe Certified Expert for the Livecycle Platform, Flash, and Dreamweaver. He is also a SCRUM Master.

Marco is author of several books including HTML5 Solutions (Apress); Flex 4 Cookbook (O'Reilly); Professional Flash Catalyst (Wrox); AIR Cookbook (O'Reilly); and Flex 4 Solutions (FriendsOfED), Advanced AIR Applications (FriendsOfED), The Essential Guide to AIR with Flash CS4 (FriendsOfED), and Flex Solutions: Essential Techniques for Flex 3 developers (FriendsOfED).

Some of his speaking engagements include international conferences such as FlashOnTheBeach, AJAXWorld Conference, O'Reilly Web 2.0 Summit, FITC, Adobe MAX, FATC New York, FlexCamp, 360Flex, TAC Singapore, MultiMania Belgium, Adobe CEM, and many others.

Peter Elst is a Belgian freelance IT consultant and Founding Partner of Project Cocoon based in Pondicherry, South India. As a respected member of the online community, Peter has spoken at various international industry events and has had his work published in leading journals.

Charles Brown is one of the most noted authors, consultants, and trainers in the IT industry today. His books on Flex have received strong critical acclaim and are used worldwide as a teaching tool.

In addition to his work in the IT industry, Charles is also a noted concert pianist, organist, and guitarist appearing in major concert centers worldwide. He began his musical studies at age 4, and he went on to study with famed pianist Vladimir Horowitz. At age 14, he made his debut with Leonard Bernstein and went on to study at the famed Juilliard School. Eventually he went to Paris to study with the 20th century legend, Igor Stravinsky. While working with Stravinsky, Charles developed a close friendship with one of the most powerful artistic forces of the 20th century: Pablo Picasso. What he learned about creativity from Picasso he uses today in his writings and training work.

Charles is a certified Adobe trainer who is in heavy demand worldwide. He frequently speaks at major conferences such as MAX and NAB. His website can be found at `www.charlesebrown.net`. You can find Charles on Facebook.

Nathalie Wormser is a freelance web developer who is passionate about emerging multimedia technologies and games. She is the co-founder of Project Cocoon Multimedia, a development and web design company based in Pondicherry, South India.

Cyril Hanquez is a consultant focusing on ColdFusion, Rich Internet and Mobile Applications. He has worked in the IT industry for about 15 years, mainly on European Union institutions projects in Belgium. He is the co-manager of the local ColdFusion User Group where he speaks frequently. Happily married and a proud father of two, Cyril is blogging at `cyrilhanquez.com/blog/`. When he is not working, you can find Cyril playing games or wandering around some popular social networks as “Fitzchev.”.”….

About the Technical Reviewer

Jason Nadon has over ten years' experience working with complex web applications. From content delivery, architecting, optimizing and marketing down to coding in a mix of PHP, Java, ASP.NET, and CFML. Jason holds several industry certifications and is currently employed at a global information company as an IT Infrastructure Manager. In his spare time, Jason enjoys reading and keeping up with technology trends, photography, and songwriting. You can check him out at: `www.thinknadon.com`.

About the Cover Image Artist

Corné van Dooren designed the front cover image for this book. After taking a brief hiatus from friends of ED to create a new design for the Foundation series, Corné worked at combining technological and organic forms with the results now appearing on this and other book covers.

Corné spent his childhood drawing on everything at hand, and then he began exploring the infinite world of multimedia—His journey of discovery hasn't stopped since! His mantra has always been, "The only limit to multimedia is the imagination," a saying that keeps him constantly moving forward.

Corné works for many international clients, writes features for multimedia magazines, reviews and tests software, authors multimedia studies, and works on many other friends of ED books. You can see more of his work and contact him through his website at: `www.cornevandooren.com`.

Acknowledgments

It happens all the time. During the writing of a book, I often have the feeling that I will never reach the end. It is only with the help and support of many people who tirelessly work behind the scenes that the book is ready on time and in good form.

Here I want to thank them all:

I would like to thank for their hard work of my coauthors: Charles, Peter, Nathalie, and Cyril. I want to thank Ben Renow-Clarke, Jennifer Blackwell, Matthew Moodie Dominic Shakeshaft, and all the Friends of ED team for giving me the opportunity and the support to write and improve this book. Their guidance and input throughout the development of this book was essential. It's awesome and incredible how their work in coordinating the editing effort with authors across different continents and time zones made collaboration so easy.

Also, special thanks are due to my technical editor, Jason Nadon, and my developmental editor, Gary Schwartz, who contributed to making the content and the examples easy to understand and follow.

And, of course, thank you to my Mom for having always pushed me to improve myself and to see beyond the surface of things.

To my brother Alessio for understanding why sometimes I did not have enough time for him.

To Katia for her patience with all the weekend and night hours spent working on this book in the past several months.

This book is significantly better because of these great people.

Introduction

The development of Hypertext Markup Language stopped in 1999 with its final version, n.4, made by the World Wide Web Consortium (W3C). Technology, however, has not stood still in the meantime: the W3C also worked on interesting projects such as the generic Standard Generalized Markup Language (SGML) to XML, as well as on new markup languages such as Scalable Vector Graphics (SVG), XForms, and MathML.

Web browser vendors, on the other hand, preferred to concentrate on new functions for their programs, whereas web developers started to learn CSS and the JavaScript language to build their applications on frameworks that use Asynchronous JavaScript + XML (AJAX).

However, things have changed, and recently HTML has been brought back to life thanks to the work of the companies such as Apple, Google, Opera Software, and the Mozilla Foundation, who collaborated (under the name of WhatWG, the Web Hypertext Application Technology Working Group) on the development of an updated and enhanced version of the old HTML.

Following this major interest, the W3C began to work on a new version of HTML, called HTML5, taking on the official name of Web Applications 1.0 and introducing new structural elements to HTML that have not been seen before.

The new elements introduced by HTML5 tend to bridge the gap between structure, defined by the markup; rendering characteristics, defined by styling directives; and the content of a web page, defined by the text itself. Furthermore, HTML5 introduced a native open standard to deliver multimedia content such as audio and video, collaboration APIs, local storage, geolocation APIs, and much more.

In this practically-oriented book, we wanted to provide a series of solutions to common problems faced by people approaching the new language. You will therefore find a lot of ready-to-use code that you can build on in your web applications.

Who is this book for?

No matters if you're a designer or a developer, this book is aimed at anybody who wants to start using HTML5 right now.

HTML5 Solutions is, in fact, intended for readers who want to take their knowledge further with quick-fire solutions to common problems and best practice techniques to improve their HTML5 skills. The book is full of solutions with real world examples and code to support you as you enter the world of HTML5 development.

Conventions used in this book

This book uses several of conventions that are worth noting. The following terms are used throughout this book:

- *HTML* refers to both the HTML and XHTML languages.
- Unless otherwise stated, *CSS* relates to the CSS 2.1 specification.
- *Modern browsers* are considered to be the latest versions of Firefox, Safari, Chrome, and Opera, along with IE 7 and above.

It is assumed that all the HTML examples in this book are nested within the `<body>` of a valid document, while the CSS is contained in an external style sheet. Occasionally, HTML and CSS have been placed in the same code example for brevity.

Sometimes code won't fit on a single line in a book. Where this happens, I've used an arrow to break the line.

With these formalities out of the way, let’s get started.

What you need

To follow and create the examples shown in this book you'll need a simple text editor. TextMate, UltraEdit, and Notepad++ are just some examples of powerful text editors with code support.

My advice is to use one of the following tools that will allow you to improve the productivity of your coding activities:

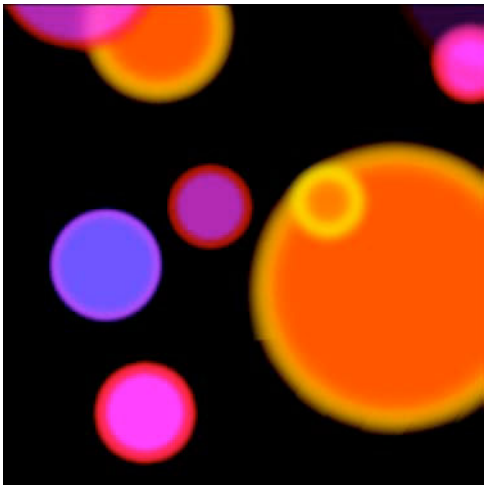
Google Web Toolkit Incubator project supports some features of HTML5 through classes like GWTCanvas. It's completely free and it can be downloaded from this `uhttp://code.google.com/p/google-web-toolkit-incubator/`

The HTML5 pack extension for Dreamweaver CS 5. It enhances Dreamweaver CS5 adding complete support to HTML5. You can download a free trial from the Adobe website `http://www.adobe.com/support/dreamweaver/`

Questions and Contacts

Please direct any technical questions or comments about the book to `m.casario@comtaste.com`.

For more information about other HTML5 and CSS books, see our website: `www.friendsofed.com`.



Chapter 1

HTML5 Page Structures

In 2004, a group of developers from Apple, Opera, and Mozilla were unhappy with the direction that HTML and XHTML were heading. In response, they formed a group called the Web Hypertext Application Technology Working Group (WHATWG). They published their first proposals in 2005 under the name Web Applications 1.0. In 2006, the World Wide Web Consortium (W3C) decided to support WHATWG officially rather than to continue developing XHTML. In 2007, the new specification was republished by the W3C under the name HTML5.

While it was thought that the final specifications would not be published until 2022, that timeline is now being reconsidered. In 2009–2010, there was an explosion of interest in HTML5 and, as a result, an increasing number of browsers and devices were introduced that support it.

This first chapter will introduce many of the new structures within the HTML5 specification. In addition, it will examine those devices that will support the new HTML5 structures.

Solution 1-1: Creating a DOCTYPE in HTML5

Because there are several versions of HTML, a browser requires a DOCTYPE type to tell it what version is in use and how to render it properly.

In this solution, you will learn to form a DOCTYPE for HTML5 properly.

What's involved

In a traditional HTML or XHTML document, the DOCTYPE tag might look as follows:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

There are many variations of the DOCTYPE.

HTML5 simplifies the DOCTYPE to:

```
<!DOCTYPE html>
```

How to build it

1. Open the HTML or text editor of your choice. For the examples shown in this chapter, we use Dreamweaver CS5. Do not use a word processor because that could embed extra characters not recognized by HTML.
2. If necessary, start a new HTML document and give it the name and location of your choice.

If you use an HTML editor like Dreamweaver, you might get code that looks as follows:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"➥
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
     <head>
          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
          <title>Untitled Document</title>
     </head>
      <body>
     </body>
</html>
```

If your code looks a little different from the above, do not worry about that for now.

3. Change the DOCTYPE tag as follows:

```
<!DOCTYPE html>
```

Expert tips

Do not leave any spaces before the DOCTYPE tag. A space could cause errors in browser rendering of the HTML5 code.

Solution 1-2: Creating a character encoding declaration in HTML5

Different languages use different character sets, or charsets. This tag declares which character set is to be used. The most common charset used by most languages is UTF-8.

In this solution, you will learn how to format the charset in HTML5 properly.

What's involved

In most HTML documents, you see the following tag at the beginning:

```
 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

HTML5 has now simplified this tag to:

```
<meta charset="UTF-8" />
```

How to build it

Under the `<!DOCTYPE html>` tag shown in Solution 1-1, type the following:

```
<meta charset = "UTF-8" />
```

Expert tips

While UTF-8 will work in most instances, a lot of developers have found that using ISO-8859-1 as the charset gives even more flexibility. Another charset, UTF-16, sometimes results in wrong characters and, in some cases, applications operating improperly.

Solution 1-3: Dividing a document into sections

In HTML, the only real way to subdivide a document into distinct sections is to use the `<div>` tag. HTML5 presents some new options.

In this solution, you will learn how to use the new HTML5 tags to create distinct document sections. In the subsequent solutions, we will discuss other structural division elements.

What's involved

The HTML `<div>` tag successfully divides the document into sections. But the word `<div>` has very little meaning in identifying the parts of a document. HTML5 provides several new structural elements that will divide the document into meaningful sections.

The first of these elements is the `<section></section>` tag. This element represents any logical division of the document. This could mean product descriptions, chapters, discussions, and so forth. While its

functionality is similar to the tag, it provides a more descriptive and content-sensitive way of dividing the document.

When creating a section in HTML5, as when you used the tag in HTML, you can use either the id or class attributes. Since both of these attributes can be applied to any HTML5 element, they are referred to as global attributes. Each id must be unique, as in HTML, and class can be used multiple times to call predefined scripting or formatting.

All HTML5 elements have three types of attributes: global, which is common to all elements; element-specific, which applies only to this element, and event handler content attributes, which will be triggered depending on content within the document. Many of these will be discussed as you progress throughout this book.

How to build it

Let's say you were creating a document about making cheesecakes. The following represents a typical use for the <section></section> elements.

```
<section id="mixing">
     <h2>The proper way to mix ingredients</h2>
     <p>When using a stand-mixer, it is important that you do not over-mix the
ingredients<.../p>
</section>
<section id="baking">
     <h2>Proper baking techniques</h2>
     <p> It is important that you bake your cheesecake using a lot of moisture in the
oven…</p>
</section>
```

Expert tips

The purpose of the <section></section> element and the subsequent structural elements shown in this chapter is not to replace the HTML tag. If you are dividing your document into logical document sections, use the <section></section> element or one of the structural elements. However, if you are dividing the document only for purposes of formatting, then the tag is appropriate to use.

Solution 1-4: Making parts of the document distributable

Increasingly, it is important to make all or part of the contents of a page distributable. For instance, forum discussion, blogs, reader comments, and so on could all be candidates for distribution or syndication.

In this solution, we will discuss the new HTML5 element, <article></article>, which makes accomplishing this much easier than with traditional HTML.

What's involved

The purpose of this structural tag is not to serve as another way to divide your document into sections. Rather, it is used to identify the portions of the document that you want to be independent and distributable from the rest of the document.

Since the `<article></article>` element is independent, it can have its own sections and subdivisions.

You can make any element distributable by surrounding it with the `<article></article>` element.

How to build it

Using the example shown in Solution 1-3, you can make the cheesecake instructions distributable as follows.

```
<article>
     <section id="mixing">
          <h2>The proper way to mix ingredients</h2>
          <p>When using a stand-mixer, it is important that you do not over mix the
ingredients…</p>
     </section>
     <section id="baking">
          <h2>Proper baking techniques</h2>
          <p> It is important that you bake your cheesecake using a lot of moisture in the
oven…</p>
     </section>
</article>
```

Expert tips

Treat the `<article></article>` element as an independent document and not as part of a larger document. That way, when it is distributed, it will be fully readable and understandable.

Solution 1-5: Creating an aside

If want to create a side discussion in traditional HTML, you use `<div>` tags and correct use of Cascading Style Sheets (CSS) for proper positioning. HTML5 makes the process easier by providing a new structural element, `<aside></aside>`. Like the `<section>` element, it provides a more descriptive way of sectioning the document.

In this solution, you will learn how to use the `<aside></aside>` element.

What's involved

Often, you might want to create what is commonly called a sidebar discussion. Figure 1-1 shows an example of accomplishing this with the `<aside></aside>` tag.

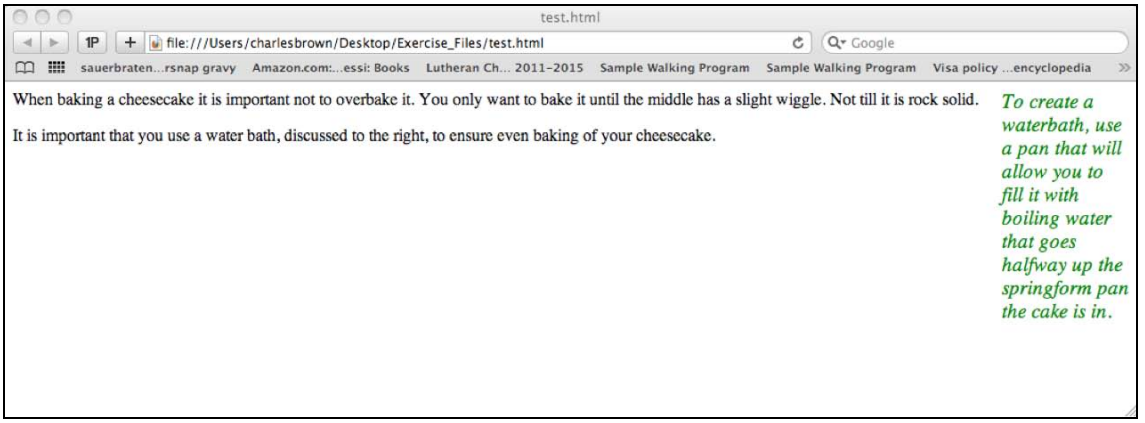


Figure 1-1. Using the <aside></aside> element

Of course, it could have been accomplished with the use of the element. The use of the <aside></aside> element, however, provides for a much more meaningful structural description.

How to build it

the example shown in the previous section was built was accomplished with the following code:

```
<aside style="font-size:larger;font-style:italic;color:blue;float:right;width:120px;">
     To create a water bath, use a pan that will allow you to fill it with boiling water that
goes halfway up the springform pan in which the cake is placed.
</aside>
<p>
     When baking a cheesecake, it is important not to over bake it. You only want to bake it
until the middle has a slight wiggle, not until it is rock solid.
</p>
<p>
     It is important that you use a water bath, discussed at the right, to ensure even baking
of your cheesecake.
</p>
```

Expert tips

Placement of the <aside></aside> element is critical. In the above example, it is placed as part of the main document. However, if you want the sidebar to be specific to a certain section, then you must place it within that section. This is especially important if you are using the <article></article> element so that the sidebar publishes with the rest of the related material.

Solution 1-6: Creating a header

Use the structural <header></header> element to create a document or section header. It can also contain <h1> to <h6>; although, as you will see later in this chapter, they are better served by the <hgroup></hgroup> element. You can also use it to help place logos and a section's table of contents.

In this solution, you will see an example of using the <header></header> element.

What's involved

The <header></header> element is an easy way to create introductions to both the document and to sections. Figure 1-2 shows the <header></header> element added to our example.

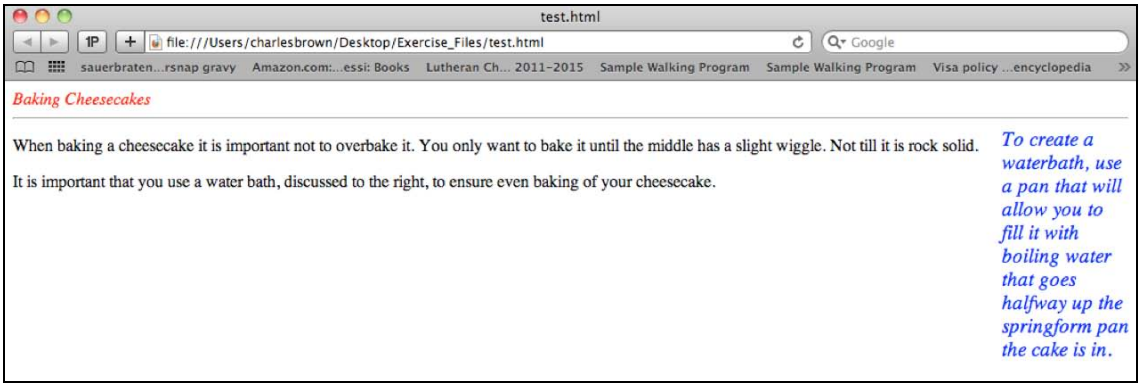


Figure 1-2. Using the <header></header> element

Notice that in this example, the <hr> element is used to draw the horizontal line. This is not a requirement of any kind.

How to build it

the following code is used to create the document shown in Figure 1-2.

```
<header>
     <span style="color:red;font-style:italic;">
          Baking Cheesecakes</span>
     <hr>
</header>
<aside style="font-size:larger;font-style:italic;color:blue;float:right;width:120px;">
     To create a water bath, use a pan that will allow you to fill it with boiling water that
goes halfway up the spring form pan in which the cake is placed.
</aside>
<p>
     When baking a cheesecake, it is important not to over bake it. You only want to bake it
until the middle has a slight wiggle, not until it is rock solid.
</p>
```

```
<p>
     It is important that you use a water bath, discussed at the right, to ensure even baking
of your cheesecake.
</p>
```

You can use the <header></header> structural element as an introduction to most of the other structural elements in the document.

Expert tips

You cannot use the <header></header> element within the <footer>, <address>, or other <header> elements. If you do, the result will be improper rendering.

Solution 1-7: Grouping <h1> to <h6> elements

In some cases, you might want to group the <h1> to <h6> elements together. As an example, you may want to create a section title that uses an <h1> element. Then, right under that, place a subtitle that uses the <h2> element. In HTML5, you can group the <h1> and <h2> elements in a new structural element called <hgroup>.

In this solution, you will see an example of using the <hgroup></hgroup> element.

What's involved

The structural <hgroup></hgroup> gives you the ability to show that you are grouping headings (<h1> to <h6>) together for needs such as alternative titles and subheadings.

Adding to the example from Solution 1-6, an <hgroup> would appear as shown in Figure 1-3.

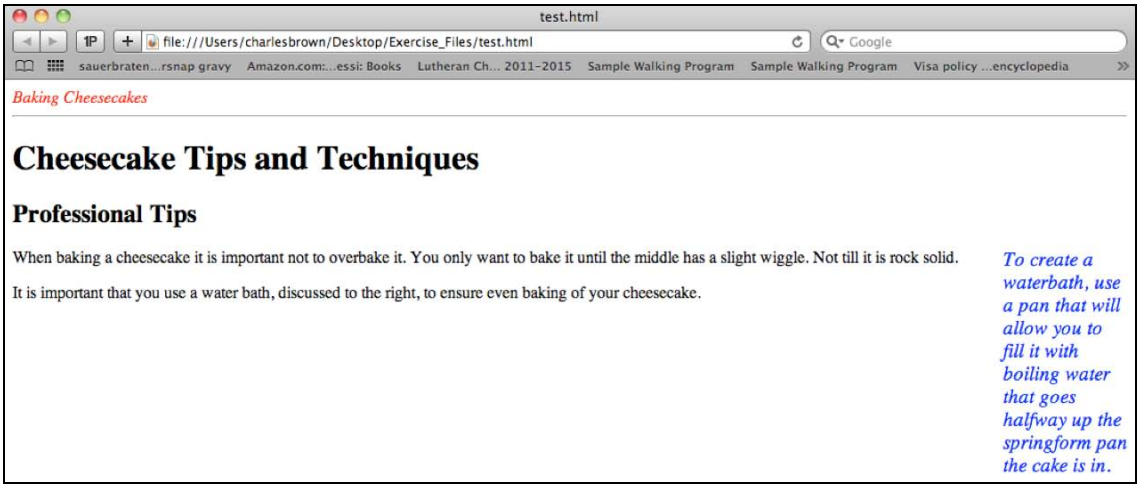


Figure 1-3. Using the <hgroup> </hgroup> element

In this case, <h1> and <h2> headings were used within the <hgroup>.

How to build it

In the example shown in Figure 1-3, the following code is used.:

```
<header>
      <span style="color:red;font-style:italic;">
          Baking Cheesecakes
     </span>
     <hr>
</header>
<hgroup draggable="true">
     <h1>Cheesecake Tips and Techniques</h1>
     <h2>Professional Tips</h2>
</hgroup>
<aside style="font-size:larger;font-style:italic;color:blue;float:right;width:120px;">
     To create a water bath, use a pan that will allow you to fill it with boiling water that
goes halfway up the spring form pan in which the cake is placed.
</aside>
<p>
     When baking a cheesecake, it is important not to over bake it. You only want to bake it
until the middle has a slight wiggle, not until it is rock solid.
</p>
<p>It is important that you use a water bath, discussed at the right, to ensure even baking of
your cheesecake.</p>
```

Notice that we placed the <hgroup> element below the <header></header> element. While this is not required, it is a good practice. You can use the <hgroup> element in any section of the HTML5 document.

Expert tips

In a well-built HTML5 document, the <hgroup></hgroup> element is a great way to tie together various headings and subheadings. This is especially true if you are using the <article></article> element. You are assured then that any headings and their connected subheadings will move as a group.

Solution 1-8: Creating a footer

As the name suggests, the <footer></footer> element will create a footer for the HTML5 document—a structural division of that document. The footer can contain copyright information, author information, citations, privacy policy, and so on.

In this solution, you will examine how the <footer></footer> element works.

What's involved

You can use the structural <footer></footer> element to create footers for an HTML5 document or any divisions within that document.

Building on Solution 1-3, the results of the <footer> element are shown in Figure 1-4.

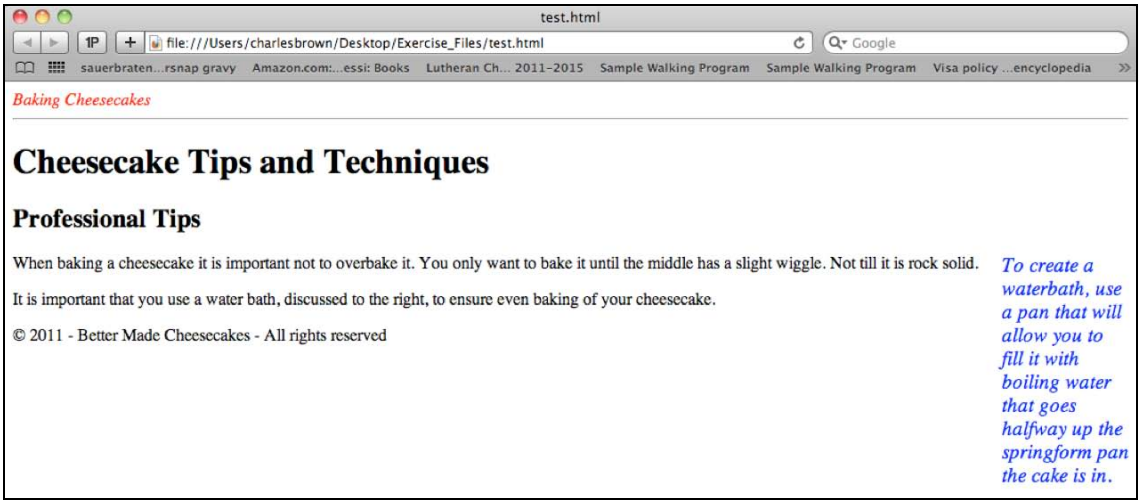


Figure 1-4. Using the <footer></footer> element

The copyright symbol, "&copy," and any text regarding rights and ownership are placed within the footer.

How to build it

In Figure 1-4, the following code is used:

```
<header>
      <span style="color:red;font-style:italic;">
          Baking Cheesecakes
     </span>
     <hr>
</header>
<hgroup draggable="true">
     <h1>Cheesecake Tips and Techniques</h1>
     <h2>Professional Tips</h2>
</hgroup>
<aside style="font-size:larger;font-style:italic;color:blue;float:right;width:120px;">
     To create a water bath, use a pan that will allow you to fill it with boiling water that
goes halfway up the spring form pan in which the cake is placed.
</aside>
<p>
     When baking a cheesecake, it is important not to over bake it. You only want to bake it
until the middle has a slight wiggle, not until it is rock solid.
</p>
<p>It is important that you use a water bath, discussed at the right, to ensure even baking of
your cheesecake.</p>
<footer> &copy; 2011 - Better Made Cheesecakes - All rights reserved </footer>
```

The <footer></footer> element can be used either for the whole HTML5 document, as it is here, or for a structural division within the document.

Expert tips

The `<footer>` element cannot be used within the <header> element or within another `<footer>` element. This would result in improper rendering.

Solution 1-9: Creating navigation in an HTML5 document

Most websites have navigational links. The links, whether they are hyperlinks or buttons of some sort, are usually separated from the rest of the document through the use of a `<div>` element. Again, other than it being a division, it does not identify the section as being specifically used for navigation. In HTML5, you can now identify the section for navigational aids in the markup.

In this section, you will learn about the new structural HTML5 element: `<nav></nav>`.

What's involved

Structural element `<nav></nav>` can be used to create a container to hold navigational elements within the entire HTML5 document or in any divisions within the document.

The result is shown in Figure 1-5.

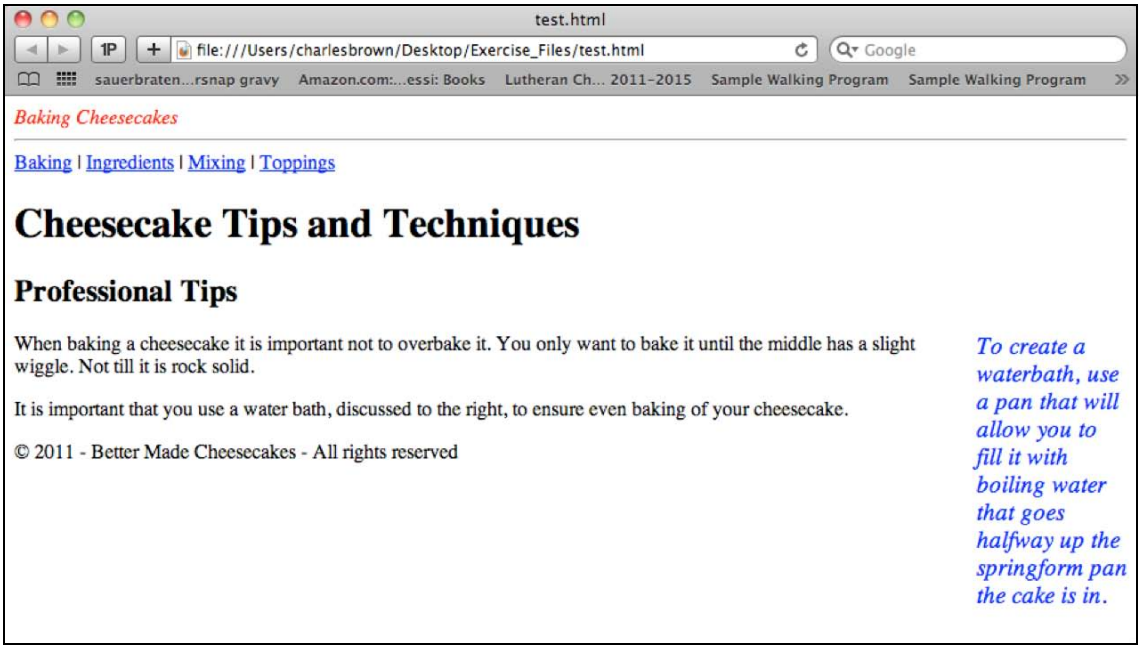


Figure 1-5. Using the `<nav></nav>` element

This gives the navigation section its own markup, which makes it easier to identify.

How to build it

The following code is used for Figure 1-5. The navigational links shown are for illustrative purposes only and are non-functional.

```
<header>
      <span style="color:red;font-style:italic;">
          Baking Cheesecakes
     </span>
     <hr>
</header>
<nav>
     <a href="/Baking/" target="_blank">Baking</a> |
     <a href="/ingredients/" target="_blank">Ingredients</a> |
     <a href="/mixing/" target="_blank">Mixing</a> |
     <a href="/toppings/" target="_blank">Toppings</a>
</nav>
<hgroup draggable="true">
     <h1>Cheesecake Tips and Techniques</h1>
     <h2>Professional Tips</h2>
</hgroup>
<aside style="font-size:larger;font-style:italic;color:blue;float:right;width:120px;">
     To create a water bath, use a pan that will allow you to fill it with boiling water that
goes halfway up the spring form pan in which the cake is placed.
</aside>
<p>
     When baking a cheesecake, it is important not to over bake it. You only want to bake it
until the middle has a slight wiggle, not until it is rock solid.
</p>
<p>It is important that you use a water bath, discussed at the right, to ensure even baking of
your cheesecake.</p>
<footer> &copy; 2011 - Better Made Cheesecakes - All rights reserved </footer>
```

As you can see, the navigational elements of your document now have their own easily identifiable markup. They can be placed either for the whole HTML5 document, as shown here, or for any subdivision of the document.

Expert tips

You are not limited to hyperlinks as shown in Figure 1-5. Within the `<nav></nav>` container element, you can put any navigational aids that you wish.

A second tip is to not use a `<nav></nav>` container within the `<footer>` element. While not strictly forbidden, it could result in improper rendering.

Solution 1-10: Inserting figures

It is fairly common to insert photos, illustrations, diagrams, and so on into a web page. Up to now, a developer could just insert an `<img>` element wherever it was needed. Now you can use markup to designate where the figures should be placed using the new `<figure></figure>` element in HTML5.

In this solution, you will see an example of using the `<figure></figure>` element.

What's involved

The structural element `<figure></figure>` can be used to create a container to hold illustrative elements within the HTML5 document or in any divisions within the document.

The result is shown in Figure 1-6.

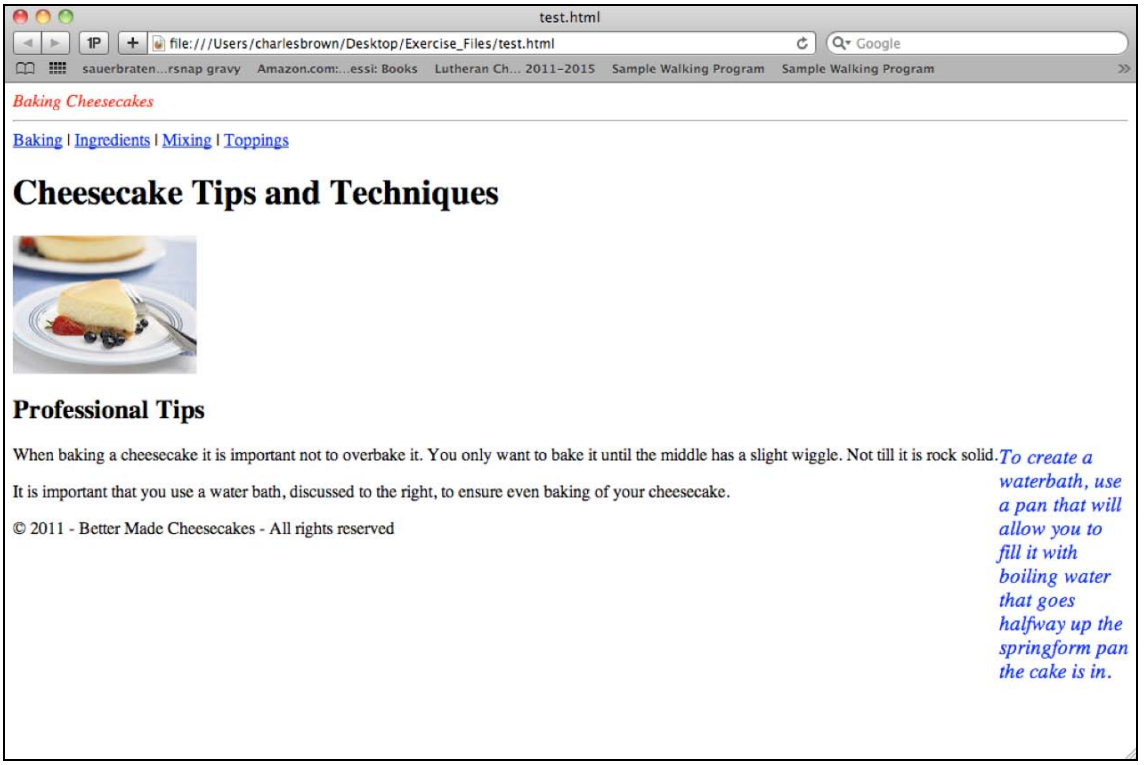


Figure 1-6. Using the `<figure></figure>` element

The illustrative elements of your document now have their own easily identifiable markup. It can be used in any subdivision of the document.

How to build it

The following code is used for Figure 1-6. The navigational links shown are for illustrative purposes only and are non-functional.

```
<header>
      <span style="color:red;font-style:italic;">
          Baking Cheesecakes
     </span>
     <hr>
</header>
<nav>
     <a href="/Baking/" target="_blank">Baking</a> |
     <a href="/ingredients/" target="_blank">Ingredients</a> |
     <a href="/mixing/" target="_blank">Mixing</a> |
     <a href="/toppings/" target="_blank">Toppings</a>
</nav>
<hgroup draggable="true">
     <h1>Cheesecake Tips and Techniques</h1>
    <figure>
        <img src="cheescake.jpg" width="170" height="128" />
    </figure>
     <h2>Professional Tips</h2>
</hgroup>
<aside style="font-size:larger;font-style:italic;color:blue;float:right;width:120px;">
     To create a water bath, use a pan that will allow you to fill it with boiling water that
goes halfway up the spring form pan in which the cake is placed.
</aside>
<p>
     When baking a cheesecake, it is important not to over bake it. You only want to bake it
until the middle has a slight wiggle, not until it is rock solid.
</p>
<p>It is important that you use a water bath, discussed at the right, to ensure even baking of
your cheesecake.</p>
<footer> &copy; 2011 - Better Made Cheesecakes - All rights reserved </footer>
```

It is easy to identify where the photo is located because it now has its own container markup using the `<figure></figure>` element.

Expert tips

Along with the new `<figure>` element comes another new HTML5 element called `<figcaption></figcaption>`. You place this within the `<figure>` element as follows:

```
<figure>
     <img src="cheescake.jpg" width="170" height="128" />
     <figcaption>One of our many cheesecakes</figcaption>
 </figure>
```

If you place the `<figcaption>` element below the picture, as shown in the example above, it will appear to the right. If you place it above, it will appear to the left.