



HTML&CSS

design and build websites

JON DUCKETT

HTML & CSS

Design and Build Websites

JON DUCKETT



JOHN WILEY & SONS, INC.

HTML & CSS

DESIGN AND BUILD WEBSITES

Published by
John Wiley & Sons, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

©2011 by John Wiley & Sons, Inc., Indianapolis, Indiana
ISBN: 978-1-118-00818-8
Manufactured in the United States of America
Published simultaneously in Canada
10 9 8 7 6 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or website may provide or recommendations it may make. Further, readers should be aware that Internet websites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Control Number: 2011932082

Trademarks: Wiley and the Wiley logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

CONTENTS

Introduction	2
Chapter 1: Structure	12
Chapter 2: Text	40
Chapter 3: Lists	62
Chapter 4: Links	74
Chapter 5: Images	94
Chapter 6: Tables	126
Chapter 7: Forms	144
Chapter 8: Extra Markup	176
Chapter 9: Flash, Video & Audio	200
Chapter 10: Introducing CSS	226
Chapter 11: Color	246
Chapter 12: Text	264
Chapter 13: Boxes	300
Chapter 14: Lists, Tables & Forms	330
Chapter 15: Layout	358
Chapter 16: Images	406
Chapter 17: HTML5 Layout	428
Chapter 18: Process & Design	452
Chapter 19: Practical Information	476
Index	493

HOW WEBSITES ARE CREATED

All websites use HTML and CSS, but content management systems, blogging software, and e-commerce platforms often add a few more technologies into the mix.

WHAT YOU SEE

When you are looking at a website, it is most likely that your browser will be receiving HTML and CSS from the web server that hosts the site. The web browser interprets the HTML and CSS code to create the page that you see.

Most web pages also include extra content such as images, audio, video, or animations and this book will teach you how to prepare them for use on the web and then how to insert them into your web pages.

Some sites also send JavaScript or Flash to your browser, and you will see how to add JavaScript and Flash in your web pages. Both of these technologies are advanced topics that you can go on to learn more about once you have mastered HTML and CSS, if you want to.

HOW IT IS CREATED

Small websites are often written just using HTML and CSS.

Larger websites — in particular those that are updated regularly and use a content management system (CMS), blogging tools, or e-commerce software — often make use of more complex technologies on the web server, but these technologies are actually used to produce HTML and CSS that is then sent to the browser. So, if your site uses these technologies, you will be able to use your new HTML and CSS knowledge to take more control over how your site looks.

Larger, more complex sites like these may use a database to store data, and programming languages such as PHP, ASP.Net, Java, or Ruby on the web server, but you do not need to know these technologies to improve what the user sees. The skills you'll learn in this book should be enough to help you on that road.

HTML5 & CSS3

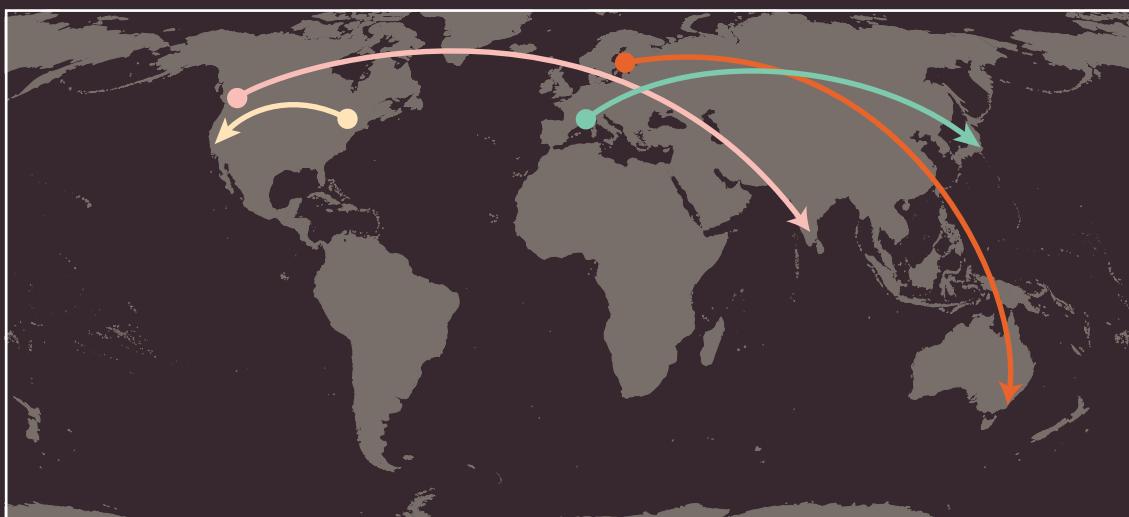
Since the web was first created there have been several versions of HTML and CSS — each intended to be an improvement on the previous version.

At the time of writing this book, HTML5 & CSS3 were still being developed. Although they had not been finalized, many browsers were already supporting some features of these languages and a lot of people were using the latest code on their websites. I have therefore chosen to teach you these latest versions.

Because HTML5 and CSS3 build on previous versions of these languages, learning these means you will also be able to understand the earlier versions of them. I have added clear notes when the code is new and also when it might not work in older browsers.

HOW THE WEB WORKS

When you visit a website, the web server hosting that site could be anywhere in the world. In order for you to find the location of the web server, your browser will first connect to a Domain Name System (DNS) server.



On this page you can see examples that demonstrate how the web server that hosts the website you are visiting can be anywhere in the world. It is the DNS servers that tell your browser how to find the website.

- A user in Barcelona visits sony.jp in Tokyo
- A user in New York visits google.com in San Francisco
- A user in Stockholm visits qantas.com.au in Sydney
- A user in Vancouver visits airindia.in in Bangalore

On the right you can see what happens when a web user in England wants to view the website of the Louvre art gallery in France which is located at www.louvre.fr. Firstly, the browser in Cambridge contacts a DNS server in London. The DNS server then tells the browser the location of the web server hosting the site in Paris.



1
When you connect to the web, you do so via an Internet Service Provider (ISP). You type a domain name or web address into your browser to visit a site; for example: google.com, bbc.co.uk, microsoft.com.

2
Your computer contacts a network of servers called Domain Name System (DNS) servers. These act like phone books; they tell your computer the IP address associated with the requested domain name. An IP address is a number of up to 12 digits separated by periods / full stops. Every device connected to the web has a unique IP address; it is like the phone number for that computer.

3
The unique number that the DNS server returns to your computer allows your browser to contact the web server that hosts the website you requested. A web server is a computer that is constantly connected to the web, and is set up especially to send web pages to users.

4
The web server then sends the page you requested back to your web browser.

1

STRUCTURE

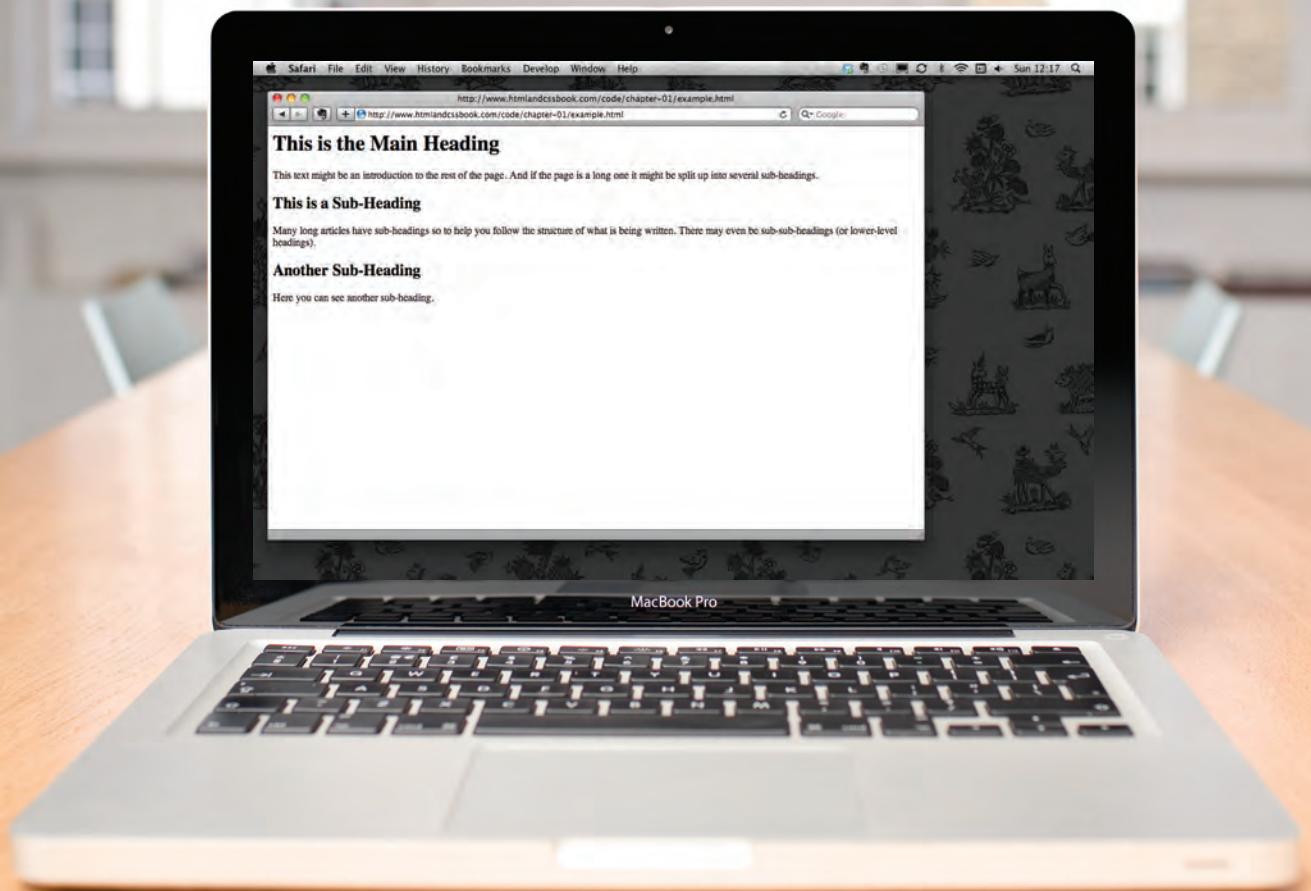
- ▶ Understanding structure
- ▶ Learning about markup
- ▶ Tags and elements

We come across all kinds of documents every day of our lives. Newspapers, insurance forms, shop catalogues... the list goes on.

Many web pages act like electronic versions of these documents. For example, newspapers show the same stories in print as they do on websites; you can apply for insurance over the web; and stores have online catalogs and e-commerce facilities.

In all kinds of documents, structure is very important in helping readers to understand the messages you are trying to convey and to navigate around the document. So, in order to learn how to write web pages, it is very important to understand how to structure documents. In this chapter you will:

- See how HTML describes the structure of a web page
- Learn how tags or elements are added to your document
- Write your first web page



HOW PAGES USE STRUCTURE

Think about the stories you read in a newspaper: for each story, there will be a headline, some text, and possibly some images. If the article is a long piece, there may be subheadings that split the story into separate sections or quotes from those involved. Structure helps readers understand the stories in the newspaper.

The structure is very similar when a news story is viewed online (although it may also feature audio or video). This is illustrated on the right with a copy of a newspaper alongside the corresponding article on its website.

Now think about a very different type of document — an insurance form. Insurance forms often have headings for different sections, and each section contains a list of questions with areas for you to fill in details or checkboxes to tick. Again, the structure is very similar online.

Read more on
MediaGuardian.co.uk

Digital economy or bust
Part 33. In which the team turn up the volume with inside track on The X Factor - and get a glimpse of the future

Coming up this week
Monday: Shortlists for Student Media Awards announced
Wednesday to Friday: Coverage of the RTS Cambridge Convention

Interview Rio Caraeff

Vevo revolutionary

Universal's former mobile chief is leading the music industry's fight to shake up online video. He reveals his frustration with MTV, and says why no one need own music if his site succeeds. Interview by **Mark Sweeney**

If Rio Caraeff succeeds, perhaps only diehard fans will need to own music. His online music video site, part-owned by the two largest record companies, aims to have the same impact as MTV and to be an answer to YouTube. Chuck those goals in with that of making the industry less dependent on the purchase of recordings, and for Caraeff there is clearly plenty to do.

Caraeff is the youthful chief executive of Vevo, which launched in late 2009 with the backing of three of the four major groups, Sony Music, Universal Music and EMI - who is taking the venture international with its partners in the UK and continental Europe. "Sex, music and sports are the only entertainment categories on the planet that people love that can build audiences of 100 million people," he says. "I'm in the business of connecting billions of people to music," is his modestly stated aim.

With music sales plummeting by \$1.8bn last year, Caraeff's mission is clear. "We wouldn't have created Vevo if we didn't need it," he says. "The industry felt it was necessary for MTV to be the first job platform, if YouTube (was), there would have been no need. We have invested tens of millions to be responsible for our own destiny. We can't sit back and say 'It's Apple or whoever' figures this

Vevo's relationship with Google, the owner of the world's most visited sharing site, YouTube, is clearly critical. Michael Grade called the company a "parasite" and Sir Martin Sorrell described it as a "freedom". Despite the lack of dissatisfaction, the music industry has historically had with players in the digital space, Caraeff prefers to characterise Vevo's dealings with YouTube as symbiotic - almost a matter of independence.

"We said 'let's figure out how to work with them,'" he explains. "There is no digital version of music videos on YouTube, there were thousands before, the official versions are only available from us. They don't threaten us. You can't compete with us. We can't upload any video in the world, we're not trying to compete." Caraeff points out that 50% of Vevo's traffic comes from YouTube search, and that the free from Vevo is a list of videos that users might like to watch that appear on the side of the YouTube web page when a user is viewing clips.

Free access

Vevo's business model is all about providing music videos that fans can access free, funded by advertising - or to put it another way, it is a free-to-access alternative to owning songs. "I believe the future is access, not ownership, not iTunes as it is today," he says. "We're not trying to sell people music, we're trying to sell them the same amount of people that want to buy music. We are about providing access: it is the only scalable model for the music industry; the question is, how do you do that without losing money?"

Which raises the question of how well Vevo is actually doing. Caraeff doesn't want to give away too much, but basically he says they are "doing really well" - "hundreds of millions of dollars" in revenue, although there are hosting costs to pay. More than half of gross revenue goes to content owners - the label, artist and publisher. The remaining 40% is kept by Vevo or paid to partners such as YouTube. He says that Vevo is "significantly ahead" of its original business plan - about 40% ahead in fact - and that it is reaching a break-even profitability "in the very early part of next year".

Yet there are problems. Caraeff's business is dependent on advertising, and he is frustrated by the low rates it gets. He is also worried about the lack of around music content. His contention is that advertisers treat music content as inferior and that Vevo's role is to "own" the premium content and not let it be poached by it as a premium product. Think the free-to-access equivalent of BSkyB owning Premier League football.

"The audience that loves music is vast and promising; it should be treated as



Video vexations ... Rio Caraeff says 'if MTV was doing a
We are about access; it is the only scalable model for the music industry; the question is, how do you do that and make money?'

Curriculum vitae

Age 35

Education Did not go to university. "I started my first company at 18."

Career 2004 vice-president, Peter Jones 2005 general manager, Universal Music Group vice-president, Universal division, responsible and new technologies, Vevo

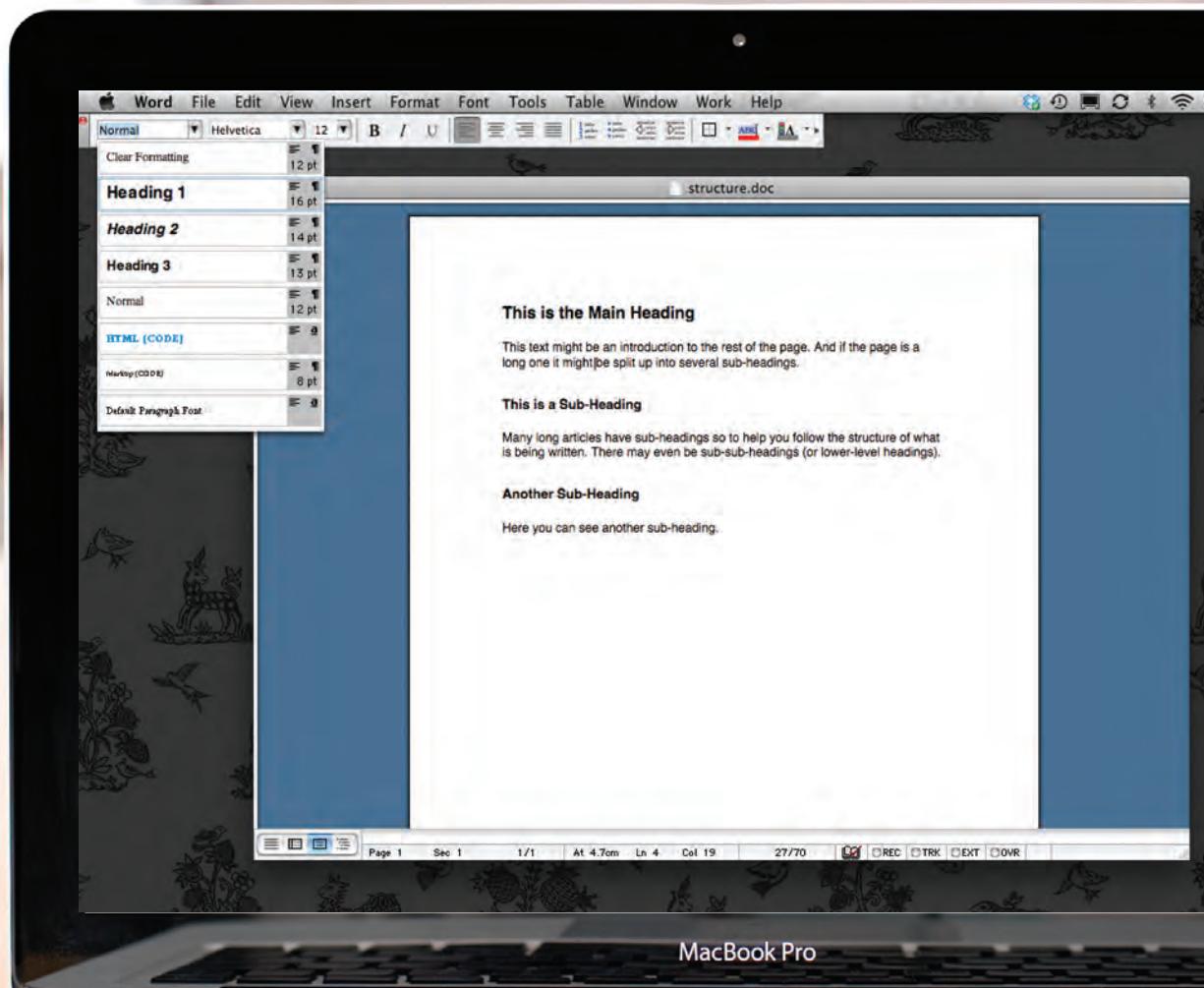


STRUCTURING WORD DOCUMENTS

The use of headings and subheadings in any document often reflects a hierarchy of information. For example, a document might start with a large heading, followed by an introduction or the most important information.

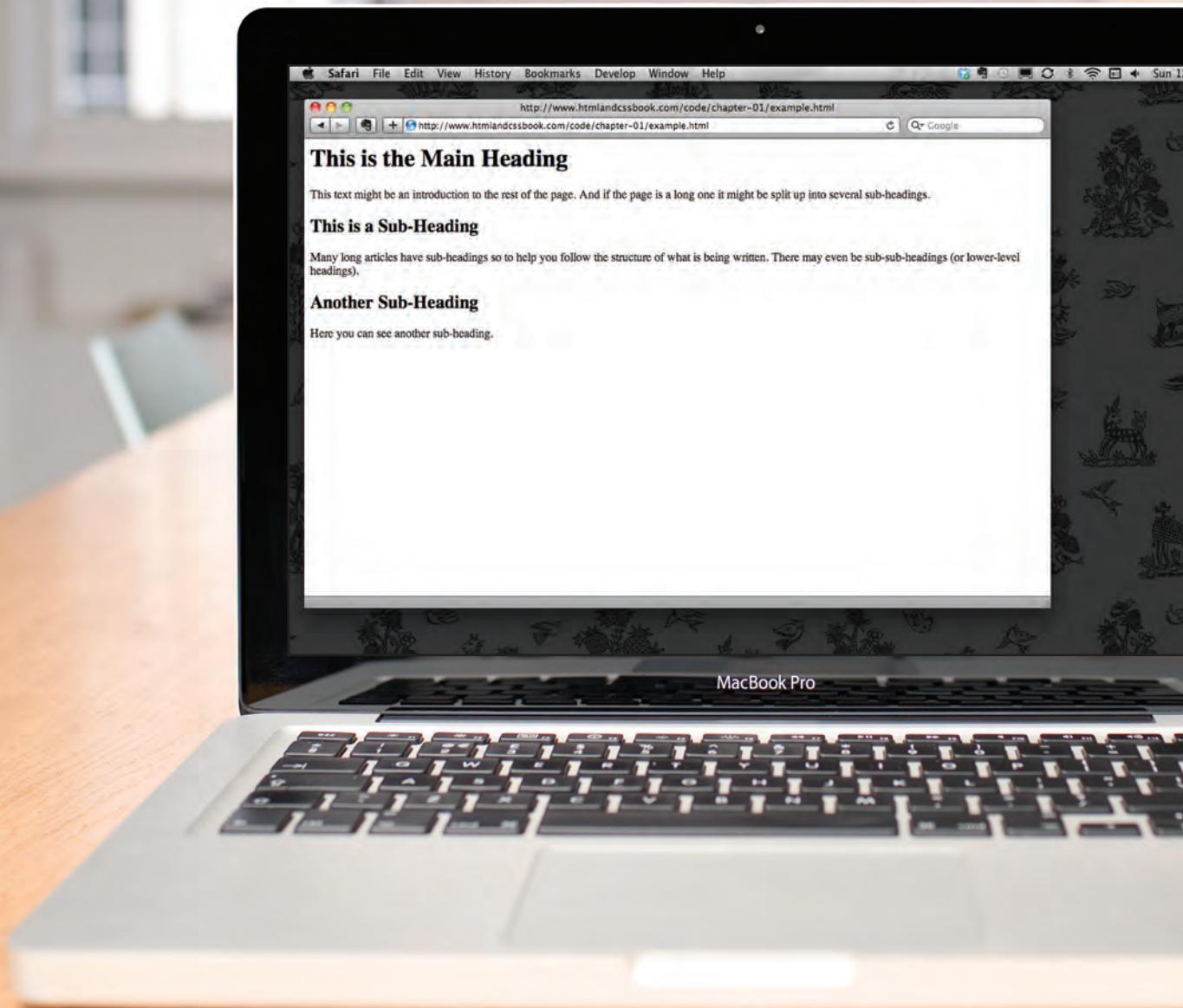
This might be expanded upon under subheadings lower down on the page. When using a word processor to create a document, we separate out the text to give it structure. Each topic might have a new paragraph, and each section can have a heading to describe what it covers.

On the right, you can see a simple document in Microsoft Word. The different styles for the document, such as different levels of heading, are shown in the drop down box. If you regularly use Word, you might have also used the formatting toolbar or palette to do this.



MacBook Pro

On the previous page you saw how structure was added to a Word document to make it easier to understand. We use structure in the same way when writing web pages.





HTML DESCRIBES THE STRUCTURE OF PAGES

In the browser window you can see a web page that features exactly the same content as the Word document you met on the page 18. To describe the structure of a web page, we add code to the words we want to appear on the page.

You can see the HTML code for this page below. Don't worry about what the code means yet. We start to look at it in more detail on the next page. Note that the HTML code is in blue, and the text you see on screen is in black.

```
<html>
  <body>
    <h1>This is the Main Heading</h1>
    <p>This text might be an introduction to the rest of
       the page. And if the page is a long one it might
       be split up into several sub-headings.<p>
    <h2>This is a Sub-Heading</h2>
    <p>Many long articles have sub-headings so to help
       you follow the structure of what is being written.
       There may even be sub-sub-headings (or lower-level
       headings).</p>
    <h2>Another Sub-Heading</h2>
    <p>Here you can see another sub-heading.</p>
  </body>
</html>
```

The HTML code (in blue) is made up of characters that live inside angled brackets — these are called HTML **elements**. Elements are usually made up of two **tags**: an opening tag and a closing tag. (The closing tag has an extra forward slash in it.) Each HTML element tells the browser something about the information that sits between its opening and closing tags.

HTML USES ELEMENTS TO DESCRIBE THE STRUCTURE OF PAGES

Let's look closer at the code from the last page.

There are several different elements. Each element has an opening tag and a closing tag.

CODE

```
<html>
  <body>
    <h1>This is the Main Heading</h1>
    <p>This text might be an introduction to the rest of
       the page. And if the page is a long one it might
       be split up into several sub-headings.</p>
    <h2>This is a Sub-Heading</h2>
    <p>Many long articles have sub-headings so to help
       you follow the structure of what is being written.
       There may even be sub-sub-headings (or lower-level
       headings).</p>
    <h2>Another Sub-Heading</h2>
    <p>Here you can see another sub-heading.</p>
  </body>
</html>
```

Tags act like containers. They tell you something about the information that lies between their opening and closing tags.

DESCRIPTION

The opening `<html>` tag indicates that anything between it and a closing `</html>` tag is HTML code.

The `<body>` tag indicates that anything between it and the closing `</body>` tag should be shown inside the main browser window.

Words between `<h1>` and `</h1>` are a main heading.

A paragraph of text appears between these `<p>` and `</p>` tags.

Words between `<h2>` and `</h2>` form a sub-heading.

Here is another paragraph between opening `<p>` and closing `</p>` tags.

Another sub-heading inside `<h2>` and `</h2>` tags.

Another paragraph inside `<p>` and `</p>` tags.

The closing `</body>` tag indicates the end of what should appear in the main browser window.

The closing `</html>` tag indicates that it is the end of the HTML code.

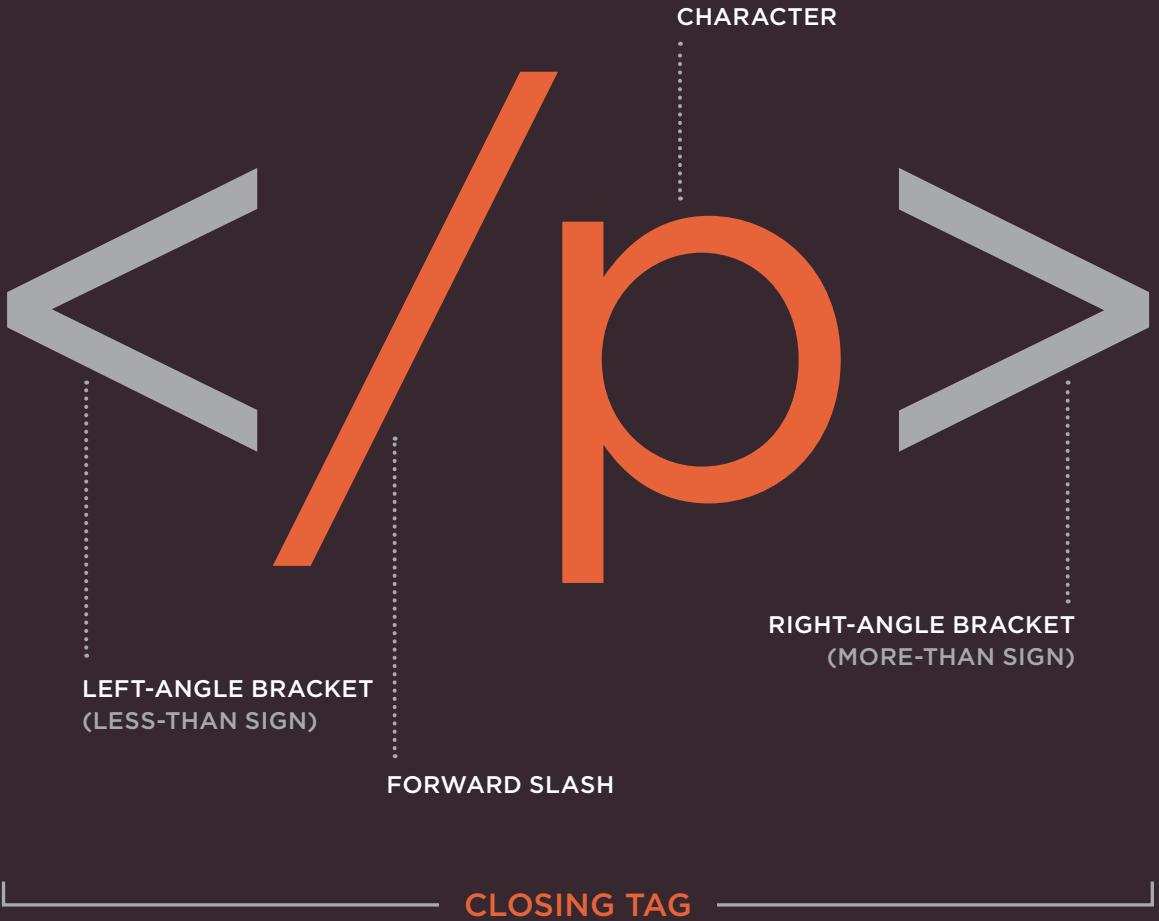
A CLOSER LOOK AT TAGS



The characters in the brackets indicate the tag's purpose.

For example, in the tags above the p stands for paragraph.

The closing tag has a forward slash after the the < symbol.



The terms "tag" and "element" are often used interchangeably.

Strictly speaking, however, an element comprises the opening

tag and the closing tag and any content that lies between them.

ATTRIBUTES TELL US MORE ABOUT ELEMENTS

Attributes provide additional information about the contents of an element. They appear on the opening tag of the element and are made up of two parts: a **name** and a **value**, separated by an equals sign.



The attribute **name** indicates what kind of extra information you are supplying about the element's content. It should be written in lowercase.

The **value** is the information or setting for the attribute. It should be placed in double quotes. Different attributes can have different values.

Here an attribute called `lang` is used to indicate the language used in this element. The value of this attribute on this page specifies it is in US English.

HTML5 allows you to use uppercase attribute names and omit the quotemarks, but this is not recommended.



The majority of attributes can only be used on certain elements, although a few attributes (such as `lang`) can appear on any element.

Most attribute values are either pre-defined or follow a stipulated format. We will look at the permitted values as we introduce each new attribute.

The value of the `lang` attribute is an abbreviated way of specifying which language is used inside the element that all browsers understand.

BODY, HEAD & TITLE

<body>

You met the `<body>` element in the first example we created. Everything inside this element is shown inside the main browser window.

<head>

Before the `<body>` element you will often see a `<head>` element. This contains information *about* the page (rather than information that is shown within the main part of the browser window that is highlighted in blue on the opposite page). You will usually find a `<title>` element inside the `<head>` element.

<title>

The contents of the `<title>` element are either shown in the top of the browser, above where you usually type in the URL of the page you want to visit, or on the tab for that page (if your browser uses tabs to allow you to view multiple pages at the same time).

/chapter-01/body-head-title.html

HTML

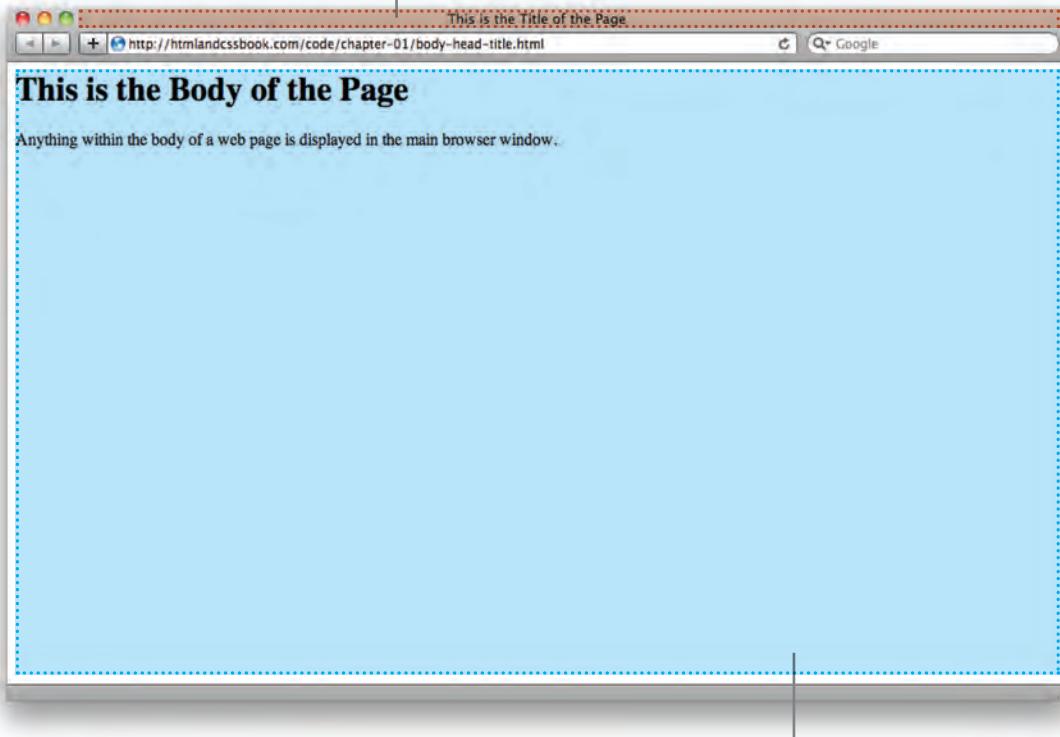
```
<html>
  <head>
    <title>This is the Title of the Page</title>
  </head>
  <body>
    <h1>This is the Body of the Page</h1>
    <p>Anything within the body of a web page is
       displayed in the main browser window.</p>
  </body>
</html>
```

RESULT

This is the Body of the Page

Anything within the body of a web page is displayed in the main browser window.

Anything written between the <title> tags will appear in the title bar (or tabs) at the top of the browser window, highlighted in orange here.



Anything written between the <body> tags will appear in the main browser window, highlighted in blue here.

You may know that HTML stands for HyperText Markup Language. The HyperText part refers to the fact that HTML allows you to create links that allow visitors to move from one

page to another quickly and easily. A markup language allows you to annotate text, and these annotations provide additional meaning to the contents of a document. If you think of a web

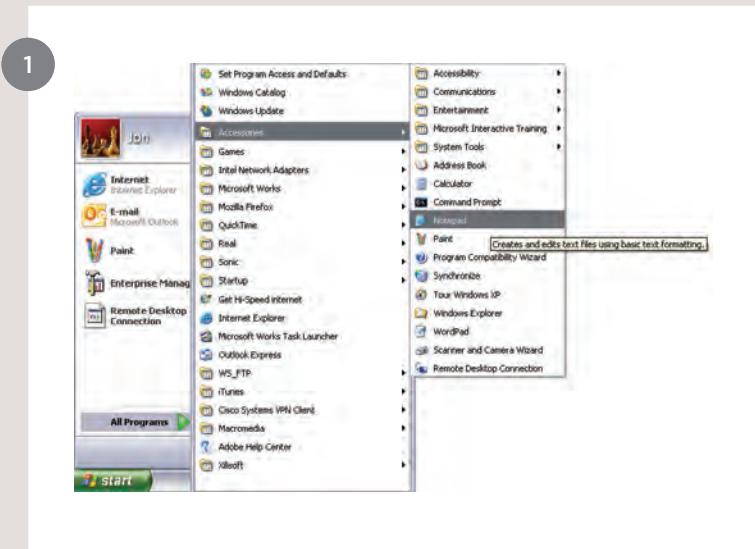
page, we add code around the original text we want to display and the browser then uses the code to display the page correctly. So the tags we add are the markup.

CREATING A WEB PAGE ON A PC

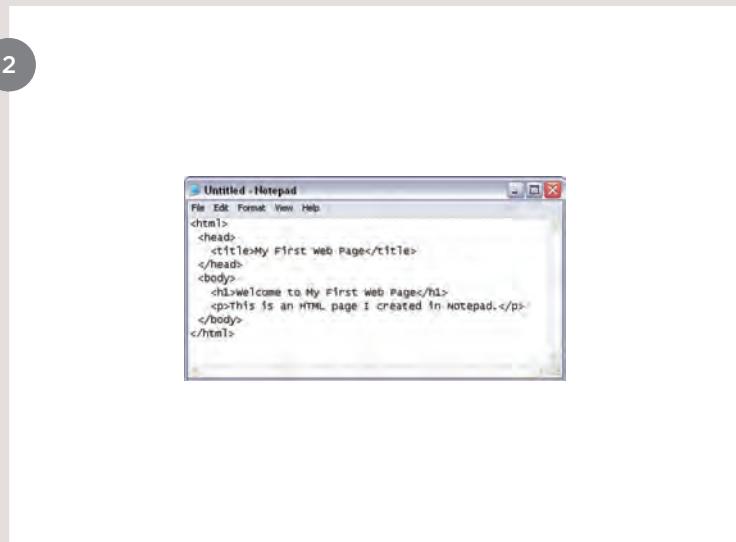
To create your first web page on a PC, start up Notepad. You can find this by going to:

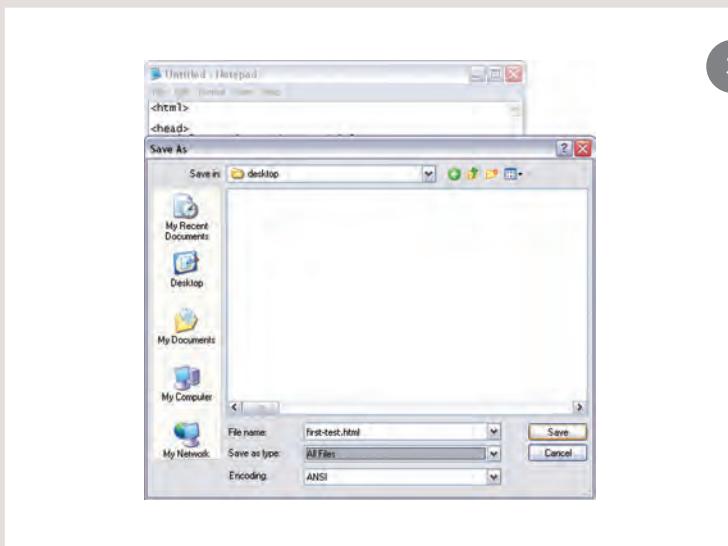
Start
All Programs (or Programs)
Accessories
Notepad

You might also like to download a free editor called Notepad++ from notepad-plus-plus.org.



Type the code shown on the right.





3

Go to the File menu and select **Save as...**. You will need to save the file somewhere you can remember. If you like, you could create a folder for any examples that you try out from this book.

Save this file as **first-test.html**. Make sure that the **Save as type** drop down has **All Files** selected.



4

Start your web browser. Go to the **File** menu and select **Open**. Browse to the file that you just created, select it and click on the **Open** button. The result should look something like the screen shot to the left.

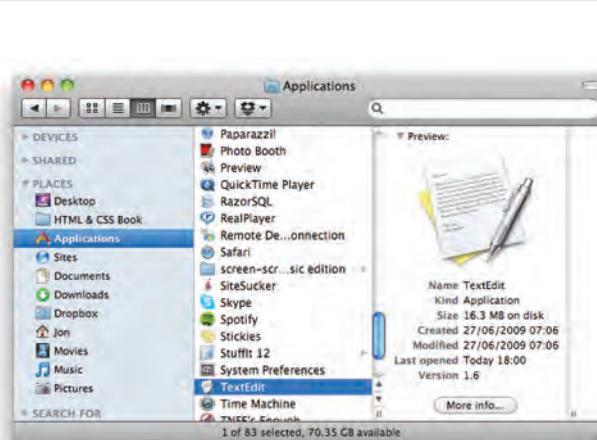
If it doesn't look like this, find the file you just created on your computer and make sure that it has the file extension **.html** (if it is **.txt** then you need to go back to Notepad and save the file again, but this time put quote marks around the name "**first-test.html**").

CREATING A WEB PAGE ON A MAC

To create your first web page on a Mac, start up TextEdit. This should be in your **Applications** folder.

You might also like to download a free text editor for creating web pages called TextWrangler which is available from barebones.com.

1

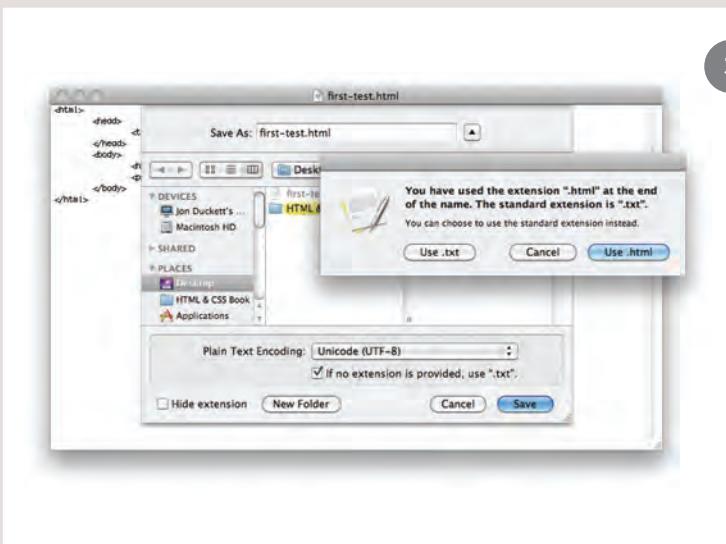


Type the code shown on the right.

2

A screenshot of a TextEdit window titled 'first-test.html'. The window displays the following HTML code:

```
<html>
  <head>
    <title>My First Web Page</title>
  </head>
  <body>
    <h1>Welcome to My First Web Page</h1>
    <p>This is an HTML page that I created in TextEdit.</p>
  </body>
</html>
```



3

Now go to the **File** menu and select **Save as...**. You will need to save the file somewhere you can remember.

If you like, you could create a folder for any examples that you try out from this book. Save this file as `first-test.html`. You will probably see a window like the screen shot to the left.

You want to select the **Use .html** button.



4

Next, start your web browser, go to the **File** menu, and select **Open**. You should browse to the file that you just created, select it and click on the **Open** button. The result should look like the screen shot to the left.

If it doesn't look like this, you might need to change one of the settings in TextEdit. Go to the TextEdit menu and select **Preferences**. Then on the preferences for **Open and Save**, tick the box that says **Ignore rich text commands in HTML files**.

Now try to save the file again.

LOOKING AT HOW OTHER SITES ARE BUILT

When the web was first taking off, one of the most common ways to learn about HTML and discover new tips and techniques was to look at the source code that made up web pages.

These days there are many more books and online tutorials that teach HTML, but you can still look at the code that a web server sends to you. To try this out for yourself, simply go to the sample code for this chapter, at www.htmlandcssbook.com/code/ and click on the link called "View Source."

Once you have opened this page, you can look for the **View** menu in your browser, and select the option that says **Source** or **View source**. (The title changes depending on what browser you are using.)

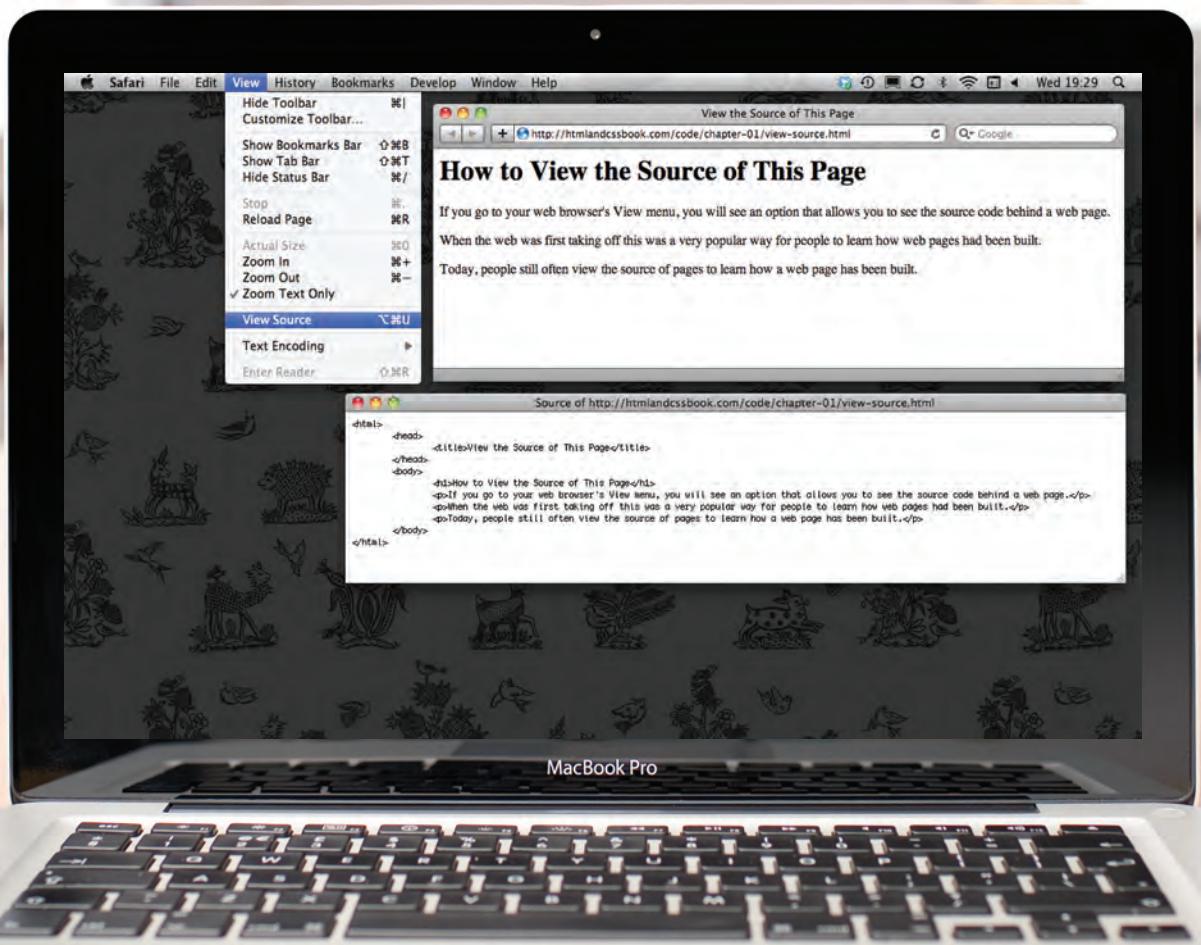
You should see a new window appear, and it will contain the source code that was used to create this page.

You can see this result in the photograph on the right. The page you see is the window at the top; the code is below.

At first this code might look complicated but don't be discouraged. By the time you have finished the next chapter of this book, you will be able to understand it.

All of the examples for this book are on the website, and you can use this simple technique on any of the example pages to see how they work.

You can also download all of the code for this book from the same website by clicking on the "Download" link.



SUMMARY

STRUCTURE

- ▶ HTML pages are text documents.
- ▶ HTML uses tags (characters that sit inside angled brackets) to give the information they surround special meaning.
- ▶ Tags are often referred to as elements.
- ▶ Tags usually come in pairs. The opening tag denotes the start of a piece of content; the closing tag denotes the end.
- ▶ Opening tags can carry attributes, which tell us more about the content of that element.
- ▶ Attributes require a name and a value.
- ▶ To learn HTML you need to know what tags are available for you to use, what they do, and where they can go.

2

TEXT

- ▶ Headings and paragraphs
- ▶ Bold, italic, emphasis
- ▶ Structural and semantic markup

When creating a web page, you add tags (known as markup) to the contents of the page. These tags provide extra meaning and allow browsers to show users the appropriate structure for the page.

In this chapter we focus on how to add markup to the text that appears on your pages. You will learn about:

- **Structural markup:** the elements that you can use to describe both headings and paragraphs
- **Semantic markup:** which provides extra information; such as where emphasis is placed in a sentence, that something you have written is a quotation (and who said it), the meaning of acronyms, and so on



HEADINGS

<h1>
<h2>
<h3>
<h4>
<h5>
<h6>

HTML has six "levels" of headings:

<h1> is used for main headings

<h2> is used for subheadings

If there are further sections under the subheadings then the <h3> element is used, and so on...

Browsers display the contents of headings at different sizes. The contents of an <h1> element is the largest, and the contents of an <h6> element is the smallest. The exact size at which each browser shows the headings can vary slightly. Users can also adjust the size of text in their browser. You will see how to control the size of text, its color, and the fonts used when we come to look at CSS.

chapter-02/headings.html

HTML

```
<h1>This is a Main Heading</h1>
<h2>This is a Level 2 Heading</h2>
<h3>This is a Level 3 Heading</h3>
<h4>This is a Level 4 Heading</h4>
<h5>This is a Level 5 Heading</h5>
<h6>This is a Level 6 Heading</h6>
```

RESULT

This is a Main Heading

This is a Level 2 Heading

This is a Level 3 Heading

This is a Level 4 Heading

This is a Level 5 Heading

This is a Level 6 Heading

PARAGRAPHS

HTML

chapter-02/paragraphs.html

```
<p>A paragraph consists of one or more sentences  
that form a self-contained unit of discourse. The  
start of a paragraph is indicated by a new  
line.</p>  
<p>Text is easier to understand when it is split up  
into units of text. For example, a book may have  
chapters. Chapters can have subheadings. Under  
each heading there will be one or more  
paragraphs.</p>
```

<p>

To create a paragraph, surround the words that make up the paragraph with an opening `<p>` tag and closing `</p>` tag.

By default, a browser will show each paragraph on a new line with some space between it and any subsequent paragraphs.

RESULT

A paragraph consists of one or more sentences that form a self-contained unit of discourse. The start of a paragraph is indicated by a new line.

Text is easier to understand when it is split up into units of text. For example, a book may have chapters. Chapters can have subheadings. Under each heading there will be one or more paragraphs.

BOLD & ITALIC

By enclosing words in the tags `` and `` we can make characters appear bold.

The `` element also represents a section of text that would be presented in a visually different way (for example key words in a paragraph) although the use of the `` element does not imply any additional meaning.

chapter-02/bold.html

HTML

```
<p>This is how we make a word appear bold.</p>
<p>Inside a product description you might see some
key features in bold.</p>
```

This is how we make a word appear **bold**.

RESULT

Inside a product description you might see some **key features** in bold.

<i>

By enclosing words in the tags `<i>` and `</i>` we can make characters appear italic.

The `<i>` element also represents a section of text that would be said in a different way from surrounding content — such as technical terms, names of ships, foreign words, thoughts, or other terms that would usually be italicized.

chapter-02/italic.html

HTML

```
<p>This is how we make a word appear italic</p>
<p>It's a potato Solanum tuberosum.</p>
<p>Captain Cook sailed to Australia on the
Endeavour.</p>
```

This is how we make a word appear *italic*.

RESULT

It's a potato *Solanum tuberosum*.

Captain Cook sailed to Australia on the *Endeavour*.

SUPERSCRIPT & SUBSCRIPT

HTML

chapter-02/superscript-and-subscript.html

```
<p>On the 4<sup>th</sup> of September you will learn  
about E=MC<sup>2</sup>. </p>  
<p>The amount of CO<sub>2</sub> in the atmosphere  
grew by 2ppm in 2009<sub>1</sub>. </p>
```

RESULT

On the 4th of September you will learn about E=MC².

The amount of CO₂ in the atmosphere grew by 2ppm in 2009₁.

<sup>

The `<sup>` element is used to contain characters that should be superscript such as the suffixes of dates or mathematical concepts like raising a number to a power such as 2².

<sub>

The `<sub>` element is used to contain characters that should be subscript. It is commonly used with foot notes or chemical formulas such as H₂O.

WHITE SPACE

In order to make code easier to read, web page authors often add extra spaces or start some elements on new lines.

When the browser comes across two or more spaces next to each other, it only displays one space. Similarly if it comes across a line break, it treats that as a single space too. This is known as **white space collapsing**.

You will often see that web page authors take advantage of white space collapsing to indent their code in order to make it easier to follow.

chapter-02/white-space.html

HTML

```
<p>The moon is drifting away from Earth.</p>
<p>The moon      is drifting away from Earth.</p>
<p>The moon is drifting away from
Earth.</p>
```

RESULT

The moon is drifting away from Earth.

The moon is drifting away from Earth.

The moon is drifting away from Earth.

LINE BREAKS & HORIZONTAL RULES

HTML

chapter-02/line-breaks.html

```
<p>The Earth<br />gets one hundred tons heavier  
every day<br />due to falling space dust.</p>
```

RESULT

The Earth
gets one hundred tons heavier every day
due to falling space dust.

As you have already seen, the browser will automatically show each new paragraph or heading on a new line. But if you wanted to add a line break inside the middle of a paragraph you can use the line break tag
.

HTML

chapter-02/horizontal-rules.html

```
<p>Venus is the only planet that rotates  
clockwise.</p>  
<hr />  
<p>Jupiter is bigger than all the other planets  
combined.</p>
```

RESULT

Venus is the only planet that rotates clockwise.

Jupiter is bigger than all the other planets combined.

<hr />

To create a break between themes — such as a change of topic in a book or a new scene in a play — you can add a horizontal rule between sections using the <hr /> tag.

There are a few elements that do not have any words between an opening and closing tag. They are known as **empty elements** and they are written differently.

An empty element usually has only one tag. Before the closing angled bracket of an empty element there will often be a space and a forward slash character. Some web page authors miss this out but it is a good habit to get into.

VISUAL EDITORS & THEIR CODE VIEWS

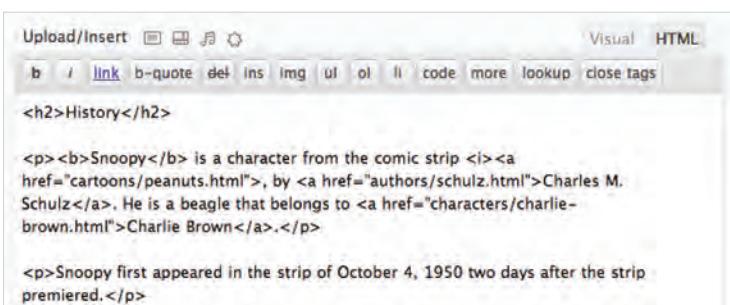
Content management systems and HTML editors such as Dreamweaver usually have two views of the page you are creating: a visual editor and a code view.

Visual editors often resemble word processors. Although each editor will differ slightly, there are some features that are common to most editors that allow you to control the presentation of text.

- Headings are created by highlighting text then using a drop-down box to select a heading.
- Bold and italic text are created by highlighting some text and pressing a **b** or *i* button.
- New paragraphs are created using the return or the enter key.
- Line breaks are created by pressing the shift key and the return key at the same time.
- Horizontal rules are created using a button with a straight line on it.

If you copy and paste text from a program that allows you to format text (such as Word) into a visual editor, it may add extra markup. To prevent this copy the text into a plain text editor first (such as Notepad on a PC orTextEdit on a Mac) and then copy it from that program and paste it into the visual editor.

Code views show you the code created by the visual editor so you can manually edit it, or so you can just enter new code yourself. It is often activated using a button with an icon that says HTML or has angled brackets. White space may be added to the code by the editor to make the code easier to read.



SEMANTIC MARKUP

There are some text elements that are not intended to affect the structure of your web pages, but they do add extra information to the pages — they are known as semantic markup.

In the rest of the chapter you will meet some more elements that will help you when you are adding text to web pages. For example, you are going to meet the `` element that allows you to indicate where emphasis should be placed on selected words and the `<blockquote>` element which indicates that a block of text is a quotation.

Browsers often display the contents of these elements in a different way. For example, the content of the `` element is shown in italics, and a `<blockquote>` is usually indented. But you should not use them to change the way that your text looks; their purpose is to describe the content of your web pages more accurately.

The reason for using these elements is that other programs, such as screen readers or search engines, can use this extra information. For example, the voice of a screen reader may add emphasis to the words inside the `` element, or a search engine might register that your page features a quote if you use the `<blockquote>` element.

STRONG & EMPHASIS

The use of the `` element indicates that its content has strong importance. For example, the words contained in this element might be said with strong emphasis.

By default, browsers will show the contents of a `` element in bold.

chapter-02/strong.html

HTML

```
<p><strong>Beware:</strong> Pickpockets operate in  
this area.</p>  
<p>This toy has many small pieces and is <strong>not  
suitable for children under five years old.  
</strong></p>
```

Beware: Pickpockets operate in this area.

RESULT

This toy has many small pieces and is **not
suitable for children under five years old.**

The `` element indicates emphasis that subtly changes the meaning of a sentence.

By default browsers will show the contents of an `` element in italic.

chapter-02/emphasis.html

HTML

```
<p>I <em>think</em> Ivy was the first.</p>  
<p>I think <em>Ivy</em> was the first.</p>  
<p>I think Ivy was the <em>first</em>.</p>
```

I think Ivy was the first.

RESULT

I think Ivy was the first.

I think Ivy was the first,

QUOTATIONS

HTML

chapter-02/quotations.html

```
<blockquote cite="http://en.wikipedia.org/wiki/  
Winnie-the-Pooh">  
  <p>Did you ever stop to think, and forget to start  
    again?</p>  
</blockquote>  
<p>As A.A. Milne said, <q>Some people talk to  
  animals. Not many listen though. That's the  
  problem.</q></p>
```

RESULT

Did you ever stop to think, and forget
to start again?

As A.A. Milne said, "Some people talk to animals.
Not many listen though. That's the problem."

There are two elements
commonly used for marking up
quotations:

<blockquote>

The `<blockquote>` element is
used for longer quotes that take
up an entire paragraph. Note
how the `<p>` element is still
used inside the `<blockquote>`
element.

Browsers tend to indent the
contents of the `<blockquote>`
element, however you should not
use this element just to indent a
piece of text — rather you should
achieve this effect using CSS.

<q>

The `<q>` element is used for
shorter quotes that sit within
a paragraph. Browsers are
supposed to put quotes around
the `<q>` element, however
Internet Explorer does not —
therefore many people avoid
using the `<q>` element.

Both elements may use the `cite`
attribute to indicate where the
quote is from. Its value should
be a URL that will have more
information about the source of
the quotation.

ABBREVIATIONS & ACRONYMS

<abbr>

If you use an abbreviation or an acronym, then the `<abbr>` element can be used. A title attribute on the opening tag is used to specify the full term.

In HTML 4 there was a separate `<acronym>` element for acronyms. To spell out the full form of the acronym, the title attribute was used (as with the `<abbr>` element above). HTML5 just uses the `<abbr>` element for both abbreviations and acronyms.

chapter-02/abbreviations.html

HTML

```
<p><abbr title="Professor">Prof</abbr> Stephen  
Hawking is a theoretical physicist and  
cosmologist.</p>  
<p><acronym title="National Aeronautics and Space  
Administration">NASA</acronym> do some crazy  
space stuff.</p>
```

RESULT

Prof Stephen Hawking is a theoretical physicist and cosmologist.
NASA do some crazy space stuff.

National Aeronautics and Space
Administration

CITATIONS & DEFINITIONS

HTML

chapter-02/citations.html

```
<p><cite>A Brief History of Time</cite> by Stephen  
Hawking has sold over ten million copies  
worldwide.</p>
```

RESULT

A Brief History of Time by Stephen Hawking has
sold over ten million copies worldwide.

<cite>

When you are referencing a piece of work such as a book, film or research paper, the `<cite>` element can be used to indicate where the citation is from.

In HTML5, `<cite>` should not really be used for a person's name — but it was allowed in HTML 4, so most people are likely to continue to use it.

Browsers will render the content of a `<cite>` element in italics.

HTML

chapter-02/definitions.html

```
<p>A <dfn>black hole</dfn> is a region of space from  
which nothing, not even light, can escape.</p>
```

RESULT

A black hole is a region of space from which
nothing, not even light, can escape.

<dfn>

The first time you explain some new terminology (perhaps an academic concept or some jargon) in a document, it is known as the defining instance of it.

The `<dfn>` element is used to indicate the defining instance of a new term.

Some browsers show the content of the `<dfn>` element in italics. Safari and Chrome do not change its appearance.

AUTHOR DETAILS

<address>

The <address> element has quite a specific use: to contain contact details for the author of the page.

It can contain a physical address, but it does not have to. For example, it may also contain a phone number or email address.

Browsers often display the content of the <address> element in italics.

You may also be interested in something called the hCard microformat for adding physical address information to your markup.

ONLINE EXTRA:

You can find out more about hCards on the website accompanying this book.

chapter-02/address.html

HTML

```
<address>
  <p><a href="mailto:homer@example.org">
    homer@example.org</a></p>
  <p>742 Evergreen Terrace, Springfield.</p>
</address>
```

homer@example.org

RESULT

742 Evergreen Terrace, Springfield.

CHANGES TO CONTENT

HTML

chapter-02/insert-and-delete.html

```
<p>It was the <del>worst</del> <ins>best</ins> idea  
she had ever had.</p>
```

RESULT

It was the worst best idea she had ever had.

**<ins>
**

The `<ins>` element can be used to show content that has been inserted into a document, while the `` element can show text that has been deleted from it.

The content of a `<ins>` element is usually underlined, while the content of a `` element usually has a line through it.

HTML

chapter-02/strikethrough.html

```
<p>Laptop computer:</p>  
<p><s>Was $995</s></p>  
<p>Now only $375</p>
```

<s>

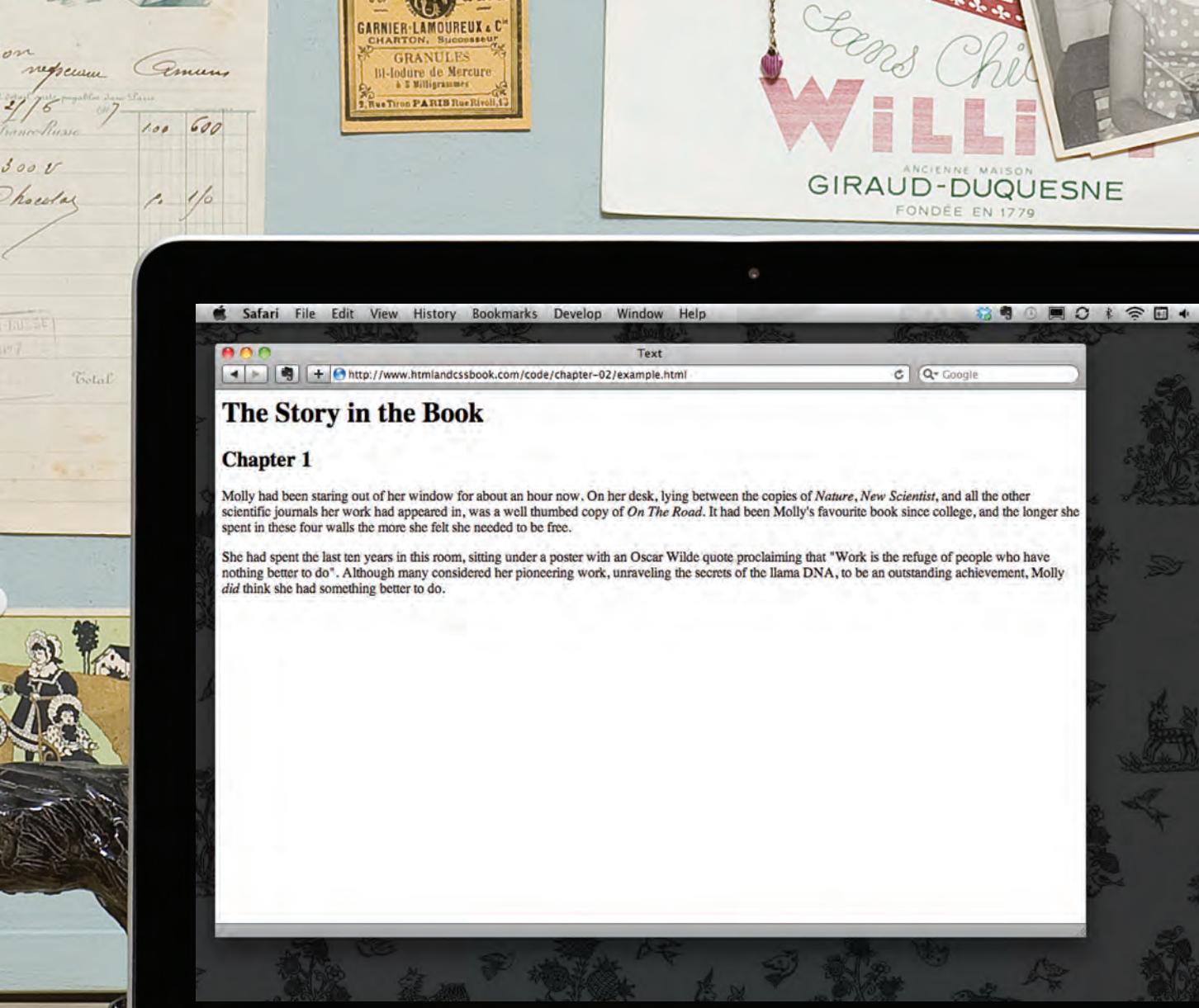
The `<s>` element indicates something that is no longer accurate or relevant (but that should not be deleted).

Visually the content of an `<s>` element will usually be displayed with a line through the center.

RESULT

Laptop computer:
Was \$995
Now only \$375

Older versions of HTML had a `<u>` element for content that was underlined, but this is being phased out.



MacBook Pro

EXAMPLE

TEXT



This is a very simple HTML page that demonstrates text markup.

Structural markup includes elements such as `<h1>`, `<h2>`, and `<p>`. Semantic information is carried in elements such as `<cite>` and ``.

```
<html>
  <head>
    <title>Text</title>
  </head>
  <body>
    <h1>The Story in the Book</h1>
    <h2>Chapter 1</h2>
    <p>Molly had been staring out of her window for about
       an hour now. On her desk, lying between the copies
       of <i>Nature</i>, <i>New Scientist</i>, and all
       the other scientific journals her work had
       appeared in, was a well thumbed copy of <cite>On
       The Road</cite>. It had been Molly's favorite book
       since college, and the longer she spent in these
       four walls the more she felt she needed to be
       free.</p>
    <p>She had spent the last ten years in this room,
       sitting under a poster with an Oscar Wilde quote
       proclaiming that <q>Work is the refuge of
       people who have nothing better to do</q>. Although
       many considered her pioneering work, unraveling
       the secrets of the llama <abbr
       title="Deoxyribonucleic acid">DNA</abbr>, to be an
       outstanding achievement, Molly <em>did</em> think
       she had something better to do.</p>
  </body>
</html>
```


SUMMARY

TEXT

- ▶ HTML elements are used to describe the structure of the page (e.g. headings, subheadings, paragraphs).
- ▶ They also provide semantic information (e.g. where emphasis should be placed, the definition of any acronyms used, when given text is a quotation).

4

LINKS

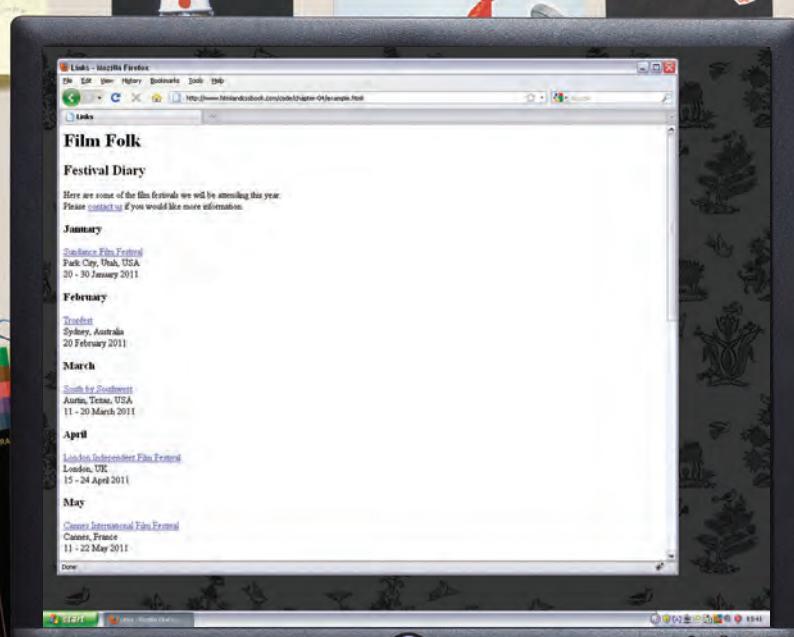
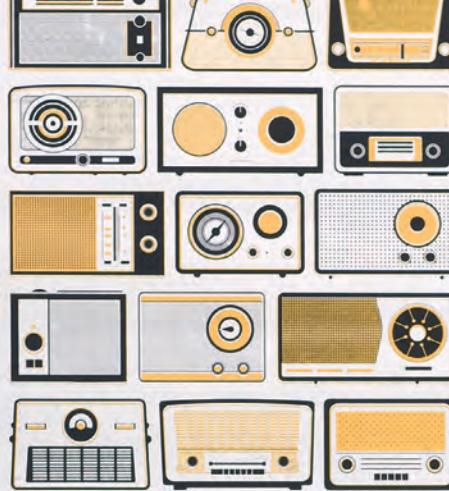
- ▶ Creating links between pages
- ▶ Linking to other sites
- ▶ Email links



Links are the defining feature of the web because they allow you to move from one web page to another — enabling the very idea of browsing or surfing.

You will commonly come across the following types of links:

- Links from one website to another
- Links from one page to another on the same website
- Links from one part of a web page to another part of the same page
- Links that open in a new browser window
- Links that start up your email program and address a new email to someone



WRITING LINKS

Links are created using the `<a>` element. Users can click on anything between the opening `<a>` tag and the closing `` tag. You specify which page you want to link to using the `href` attribute.

The diagram illustrates the structure of an anchor tag (`<a>`). It shows the tag `IMDB` with annotations explaining its components. A bracket under the tag is labeled "OPENING LINK TAG". Another bracket covers the URL "http://www.imdb.com" and the word "IMDB" is labeled "THIS IS THE TEXT THE USER CLICKS ON". A final bracket at the end of the tag is labeled "CLOSING LINK TAG".

THIS IS THE PAGE THE
LINK TAKES YOU TO

THIS IS THE TEXT THE
USER CLICKS ON

OPENING LINK TAG

CLOSING
LINK TAG

```
<a href="http://www.imdb.com">IMDB</a>
```

The text between the opening <a> tag and closing tag is known as link text. Where possible, your link text should explain where visitors will be taken if they click on it (rather than just saying "click here"). Below you can see the link to IMDB that was created on the previous page.

Many people navigate websites by scanning the text for links. Clear link text can help visitors find what they want. This will give them a more positive impression of your site and may encourage them to visit it for longer. (It also helps people using screen reader software.)

To write good link text, you can think of words people might use when searching for the page that you are linking to. (For example, rather than write "places to stay" you could use something more specific such as "hotels in New York.")



IMDB

LINKING TO OTHER SITES

<a>

Links are created using the `<a>` element which has an attribute called `href`. The value of the `href` attribute is the page that you want people to go to when they click on the link.

Users can click on anything that appears between the opening `<a>` tag and the closing `` tag and will be taken to the page specified in the `href` attribute.

When you link to a different website, the value of the `href` attribute will be the full web address for the site, which is known as an **absolute URL**.

Browsers show links in blue with an underline by default.

chapter-04/linking-to-other-sites.html

HTML

```
<p>Movie Reviews:<br/><ul><li><a href="http://www.empireonline.com">Empire</a></li><li><a href="http://www.metacritic.com">Metacritic</a></li><li><a href="http://www.rottentomatoes.com">Rotten Tomatoes</a></li><li><a href="http://www.variety.com">Variety</a></li></ul></p>
```

RESULT

Movie Reviews:

- [Empire](http://www.empireonline.com)
- [Metacritic](http://www.metacritic.com)
- [Rotten Tomatoes](http://www.rottentomatoes.com)
- [Variety](http://www.variety.com)

ABSOLUTE URLs

URL stands for Uniform Resource Locator. Every web page has its own URL. This is the web address that you would type into a browser if you wanted to visit that specific page.

An absolute URL starts with the domain name for that site, and can be followed by the path to a specific page. If no page is specified, the site will display the homepage.

LINKING TO OTHER PAGES ON THE SAME SITE

HTML

chapter-04/linking-to-other-pages.html

```
<p>
<ul>
  <li><a href="index.html">Home</a></li>
  <li><a href="about-us.html">About</a></li>
  <li><a href="movies.html">Movies</a></li>
  <li><a href="contact.html">Contact</a></li>
</ul>
</p>
```

RESULT

- [Home](#)
- [About](#)
- [Movies](#)
- [Contact](#)

<a>

When you are linking to other pages within the same site, you do not need to specify the domain name in the URL. You can use a shorthand known as a **relative URL**.

If all the pages of the site are in the same folder, then the value of the `href` attribute is just the name of the file.

If you have different pages of a site in different folders, then you can use a slightly more complex syntax to indicate where the page is in relation to the current page. You will learn more about these on the pages 81-84.

If you look at the download code for each chapter, you will see that the `index.html` file contains links that use relative URLs.

RELATIVE URLs

When linking to other pages within the same site, you can use relative URLs. These are like a shorthand version of absolute URLs because you do not need to specify the domain name.

We will take a closer look at relative URLs on pages 83-84 as there are several helpful shortcuts you can use to write links to other pages on your own website.

Relative URLs help when building a site on your computer because you can create links between pages without having to set up your domain name or hosting.

DIRECTORY STRUCTURE

On larger websites it's a good idea to organize your code by placing the pages for each different section of the site into a new folder. Folders on a website are sometimes referred to as directories.

STRUCTURE

The diagram on the right shows the directory structure for a fictional entertainment listings website called ExampleArts.

The top-level folder is known as the **root** folder. (In this example, the root folder is called **examplearts**.) The root folder contains all of the other files and folders for a website.

Each section of the site is placed in a separate folder; this helps organize the files.

If you are working with a content management system, blogging software, or an e-commerce system, you might not have individual files for each page of the website.

RELATIONSHIPS

The relationship between files and folders on a website is described using the same terminology as a family tree.

In the diagram on the right, you can see some relationships have been drawn in.

The **examplearts** folder is a parent of the **movies**, **music** and **theater** folders. And the **movies**, **music** and **theater** folders are children of the **examplearts** folder.

Instead, these systems often use one template file for each different type of page (such as news articles, blog posts, or products).

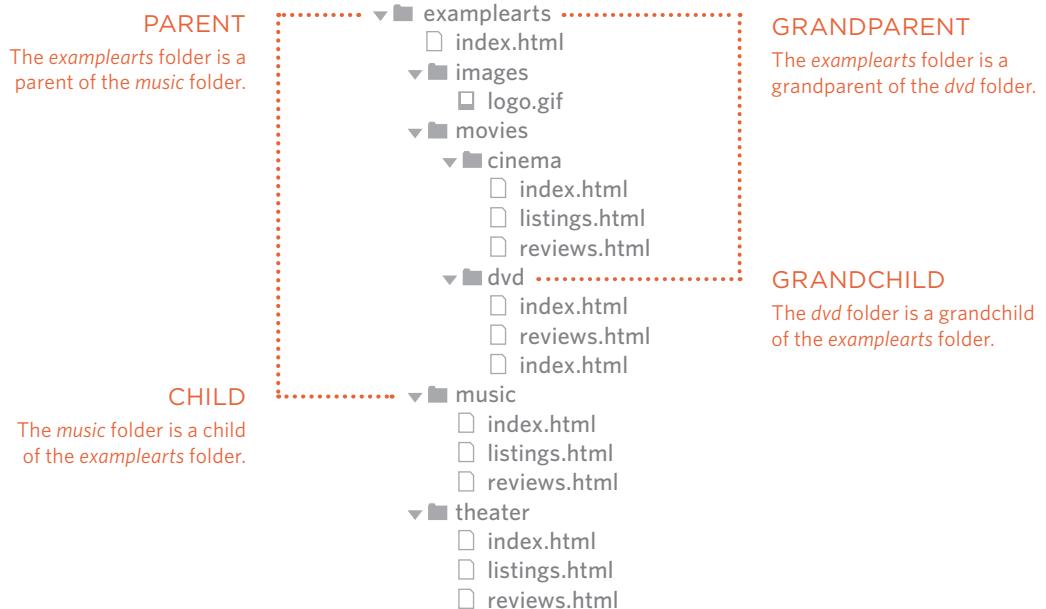
HOME PAGES

The main homepage of a site written in HTML (and the homepages of each section in a child folder) is called **index.html**.

Web servers are usually set up to return the **index.html** file if no file name is specified.

Therefore, if you enter `examplearts.com` it will return `examplearts.com/index.html`, and `examplearts.com/music` will return `examplearts.com/music/index.html`.

Editing the template file would change all of the pages that use that template. Do not change any code that is not HTML or you may break the page.



Every page and every image on a website has a **URL** (or Uniform Resource Locator). The URL is made up of the domain name followed by the **path** to that page or image.

The path to the homepage of this site is `www.examplearts.com/index.html`. The path to the logo for the site is `examplearts.com/images/logo.gif`.

You use URLs when linking to other web pages and when including images in your own site. On the next page, you will meet a shorthand way to link to files on your own site.

The root folder contains:

- A file called *index.html* which is the homepage for the entire site
- Individual folders for the movies, music and theatre sections of the site

Each sub-directory contains:

- A file called *index.html* which is the homepage for that section
- A reviews page called *reviews.html*
- A listings page called *listings.html* (except for the DVD section)

The movies section contains:

- A folder called *cinema*
- A folder called *DVD*.

RELATIVE URLs

Relative URLs can be used when linking to pages within your own website. They provide a shorthand way of telling the browser where to find your files.

When you are linking to a page on your own website, you do not need to specify the domain name. You can use **relative URLs** which are a shorthand way to tell the browser where a page is in relation to the current page.

This is especially helpful when creating a new website or learning about HTML because you can create links between pages when they are only on your personal computer (before you have got a domain name and uploaded them to the web).

Because you do not need to repeat the domain name in each link, they are also quicker to write.

If all of the files in your site are in one folder, you simply use the file name for that page.

If your site is organized into separate folders (or directories), you need to tell the browser how to get from the page it is *currently on* to the page that you are *linking to*.

If you link to the same page from two different pages you might, therefore, need to write two different relative URLs.

These links make use of the same terminology (borrowed from that of family trees) you met on the previous page which introduces directory structure.

RELATIVE LINK TYPE

EXAMPLE (from diagram on previous page)

SAME FOLDER

To link to a file in the same folder, just use the file name. (Nothing else is needed.)

To link to music reviews from the music homepage:
`Reviews`

CHILD FOLDER

For a child folder, use the name of the child folder, followed by a forward slash, then the file name.

To link to music listings from the homepage:
`Listings`

GRANDCHILD FOLDER

Use the name of the child folder, followed by a forward slash, then the name of the grandchild folder, followed by another forward slash, then the file name.

To link to DVD reviews from the homepage:
`Reviews`

PARENT FOLDER

Use `../` to indicate the folder above the current one, then follow it with the file name.

To link to the homepage from the music reviews:
`Home`

GRANDPARENT FOLDER

Repeat the `../` to indicate that you want to go up two folders (rather than one), then follow it with the file name.

To link to the homepage from the DVD reviews:
`Home`

When a website is live (that is, uploaded to a web server) you may see a couple of other techniques used that do not work when the files are on your local computer.

For example, you may see the name of a child folder without the name of a file. In this case the web server will usually try to show the homepage for that section.

A forward slash will return the homepage for the entire site, and a forward slash followed by a file name will return that file providing it is in the root directory.

LINKING TO A SPECIFIC PART OF THE SAME PAGE

At the top of a long page you might want to add a list of contents that links to the corresponding sections lower down. Or you might want to add a link from part way down the page back to the top of it to save users from having to scroll back to the top.

Before you can link to a specific part of a page, you need to identify the points in the page that the link will go to. You do this using the `id` attribute (which can be used on every HTML element). You can see that the `<h1>` and `<h2>` elements in this example have been given `id` attributes that identify those sections of the page.

The value of the `id` attribute should start with a letter or an underscore (not a number or any other character) and, on a single page, no two `id` attributes should have the same value.

To link to an element that uses an `id` attribute you use the `<a>` element again, but the value of the `href` attribute starts with the `#` symbol, followed by the value of the `id` attribute of the element you want to link to. In this example, `` links to the `<h1>` element at the top of the page whose `id` attribute has a value of `top`.

chapter-05/linking-to-a-specific-part.html

HTML

```
<h1 id="top">Film-Making Terms</h1>
<a href="#arc_shot">Arc Shot</a><br />
<a href="#interlude">Interlude</a><br />
<a href="#prologue">Prologue</a><br /><br />
<h2 id="arc_shot">Arc Shot</h2>
<p>A shot in which the subject is photographed by an encircling or moving camera</p>
<h2 id="interlude">Interlude</h2>
<p>A brief, intervening film scene or sequence, not specifically tied to the plot, that appears within a film</p>
<h2 id="prologue">Prologue</h2>
<p>A speech, preface, introduction, or brief scene preceding the the main action or plot of a film; contrast to epilogue</p>
<p><a href="#top">Top</a></p>
```

LINKING TO A SPECIFIC PART OF ANOTHER PAGE

RESULT

Film-Making Terms

[Arc Shot](#)
[Interlude](#)
[Prologue](#)

Arc Shot

A shot in which the subject is photographed by an encircling or moving camera

Interlude

A brief, intervening film scene or sequence, not specifically tied to the plot, that appears within a film

Prologue

A speech, preface, introduction, or brief scene preceding the main action or plot of a film; contrast to epilogue

[Top](#)

If you want to link to a specific part of a different page (whether on your own site or a different website) you can use a similar technique.

As long as the page you are linking to has id attributes that identify specific parts of the page, you can simply add the same syntax to the end of the link for that page.

Therefore, the href attribute will contain the address for the page (either an absolute URL or a relative URL), followed by the # symbol, followed by the value of the id attribute that is used on the element you are linking to.

For example, to link to the bottom of the homepage of the website that accompanies this book, you would write:

```
<a href="http://www.htmlandcssbook.com/#bottom">
```

THE EVOLUTION OF HTML

Since the web was first created, there have been several different versions of HTML.

HTML 4

RELEASED 1997

Each new version was designed to be an improvement on the last (with new elements and attributes added and older code removed).

There have also been several versions of each browser used to view web pages, each of which implements new code. Not all web users, however, have the latest browsers installed on their computers, which means that not everyone will be able to view all of the latest features and markup.

Where you should be particularly aware of browsers not supporting certain features, I have made a note of this (as you have seen with some of the HTML5 elements introduced in the Forms chapter — and as you will see in the CSS chapters).

With the exception of a few elements added in HTML5 (which have been highlighted), the elements you have seen in this book were all available in HTML 4.

Although HTML 4 had some presentational elements to control the appearance of pages, authors are not recommended to use them any more. (Examples include the `<center>` element for centering content on a page, `` for controlling the appearance of text, and `<strike>` to put a line through the text — all of these can be achieved with CSS instead.)

XHTML 1.0

RELEASED 2000

In 1998, a language called XML was published. Its purpose was to allow people to write new markup languages. Since HTML was the most widely used markup language around, it was decided that HTML 4 should be reformulated to follow the rules of XML and it was renamed XHTML. This meant that authors had to follow some new, more strict rules about writing markup. For example:

- Every element needed a closing tag (except for empty elements such as ``).
- Attribute names had to be in lowercase.
- All attributes required a value, and all values were to be placed in double quotes.
- Deprecated elements should no longer be used.
- Every element that was opened inside another element should be closed inside that same element.

HTML5

RELEASED 2000

The examples in this book all follow these strict rules of XML.

One of the key benefits of this change was that XHTML works seamlessly with other programs that are written to create and process XML documents.

It could also be used with other data formats such as Scalable Vector Graphics (SVG) — a graphical language written in XML, MathML (used to mark up mathematical formulae), and CML (used to mark up chemical formulae).

In order to help web page authors move to this new syntax, two main flavors of XHTML 1.0 were created:

- **Strict XHTML 1.0**, where authors had to follow the rules to the letter
- **Transitional XHTML 1.0**, where authors could still use presentational elements (such as `<center>` and ``).

The transitional version of XHTML was created because it allowed authors to continue to follow older practices (with a less strict syntax) and use some of the elements and attributes that were going to be removed from future versions of HTML.

There was also a third version of XHTML 1.0 called **XHTML 1.0 Frameset**, which allowed web page authors to partition a browser window into several "frames," each of which would hold a different HTML page. These days, frames are very rarely used and are being phased out.

In HTML5, web page authors do not need to close all tags, and new elements and attributes will be introduced. At the time of writing, the HTML5 specification had not been completed, but the major browser makers had started to implement many of the new features, and web page authors were rapidly adopting the new markup.

Despite the fact that HTML5 is not yet completed, you can safely take advantage of the new features of the language as long as you endeavour to ensure that users with older browsers will be able to view your pages (even though some of the extra features will not be visible to them).

DOCTYPES

Because there have been several versions of HTML, each web page should begin with a DOCTYPE declaration to tell a browser which version of HTML the page is using (although browsers usually display the page even if it is not included). We will therefore be including one in each example for the rest of the book.

As you will see when we come to look at CSS and its box model on page 316, the use of a DOCTYPE can also help the browser to render a page correctly.

Because XHTML was written in XML, you will sometimes see pages that use the XHTML strict DOCTYPE start with the optional XML declaration. Where this is used, it should be the first thing in a document. There must be nothing before it, not even a space.

```
HTML5
<!DOCTYPE html>

HTML 4
<!DOCTYPE html PUBLIC
  "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">

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

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

XML Declaration
<?xml version="1.0" ?>
```

COMMENTS IN HTML

HTML

chapter-08/comments-in-html.html

```
<!-- start of introduction -->
<h1>Current Exhibitions</h1>
<h2>Olafur Eliasson</h2>
<!-- end of introduction -->
<!-- start of main text -->
<p>Olafur Eliasson was born in Copenhagen, Denmark
    in 1967 to Icelandic parents.</p>
<p>He is known for sculptures and large-scale
    installation art employing elemental materials
    such as light, water, and air temperature to
    enhance the viewer's experience.</p>
<!-- end of main text -->
<!--
    <a href="mailto:info@example.org">Contact</a>
-->
```

RESULT

Current Exhibitions

Olafur Eliasson

Olafur Eliasson was born in Copenhagen, Denmark in 1967 to Icelandic parents.

He is known for sculptures and large-scale installation art employing elemental materials such as light, water, and air temperature to enhance the viewer's experience.

<!-- -->

If you want to add a comment to your code that will not be visible in the user's browser, you can add the text between these characters:

<!-- comment goes here -->

It is a good idea to add comments to your code because, no matter how familiar you are with the page at the time of writing it, when you come back to it later (or if someone else needs to look at the code), comments will make it much easier to understand.

Although comments are not visible to users in the main browser window, they can be viewed by anyone who looks at the source code behind the page.

On a long page you will often see comments used to indicate where sections of the page start or end, and to pass on notes to help anyone who is looking at the code understand it.

Comments can also be used around blocks of code to stop that code from being displayed in the browser. In the example on the left, the email link has been commented out.

ID ATTRIBUTE

Every HTML element can carry the id attribute. It is used to uniquely identify that element from other elements on the page. Its value should start with a letter or an underscore (not a number or any other character). It is important that no two elements on the same page have the same value for their id attributes (otherwise the value is no longer unique).

As you will see when you come to look at CSS in the next section, giving an element a unique identity allows you to style it differently than any other instance of the same element on the page. For example, you might want to assign one paragraph within the page (perhaps a paragraph containing a pull quote) a different style than all of the other paragraphs. In the example on the right, the paragraph with the id attribute whose value is `pullquote` is made uppercase using CSS.

If you go on to learn about JavaScript (a language that allows you to add interactivity to your pages), id attributes can be used to allow the script to work with that particular element.

The id attribute is known as a **global attribute** because it can be used on any element.

chapter-08/id-attribute.html

HTML

```
<p>Water and air. So very commonplace are these substances, they hardly attract attention - and yet they vouchsafe our very existence.</p>
<p id="pullquote">Every time I view the sea I feel a calming sense of security, as if visiting my ancestral home; I embark on a voyage of seeing.
</p>
<p>Mystery of mysteries, water and air are right there before us in the sea.</p>
```

RESULT

Water and air. So very commonplace are these substances, they hardly attract attention - and yet they vouchsafe our very existence.

EVERY TIME I VIEW THE SEA I FEEL A CALMING SENSE OF SECURITY, AS IF VISITING MY ANCESTRAL HOME; I EMBARK ON A VOYAGE OF SEEING.

Mystery of mysteries, water and air are right there before us in the sea.

CLASS ATTRIBUTE

HTML

chapter-08/class-attribute.html

```
<p class="important">For a one-year period from  
November 2010, the Marugame Genichiro-Inokuma  
Museum of Contemporary Art (MIMOCA) will host a  
cycle of four Hiroshi Sugimoto exhibitions.</p>  
<p>Each will showcase works by the artist  
thematically contextualized under the headings  
"Science," "Architecture," "History" and  
"Religion" so as to present a comprehensive  
panorama of the artist's oeuvre.</p>  
<p class="important admittance">Hours: 10:00 - 18:00  
(No admittance after 17:30)</p>
```

RESULT

FOR A ONE-YEAR PERIOD FROM NOVEMBER 2010,
THE MARUGAME GENICHIRO-INOKUMA MUSEUM
OF CONTEMPORARY ART (MIMOCA) WILL HOST A
CYCLE OF FOUR HIROSHI SUGIMOTO EXHIBITIONS.

Each will showcase works by the artist thematically
contextualized under the headings "Science," "Architecture,"
"History" and "Religion" so as to present a comprehensive
panorama of the artist's oeuvre.

HOURS: 10:00 - 18:00 (NO ADMITTANCE AFTER 17:30)

By default, using these attributes does not affect the presentation of an element. It will only change their appearance if there is a CSS rule that indicates it should be displayed differently.

In this example, CSS has been applied to make elements with a class attribute whose value is important uppercase, and elements with a class attribute whose value is admittance red.

Every HTML element can also carry a class attribute. Sometimes, rather than uniquely identifying one element within a document, you will want a way to identify several elements as being different from the other elements on the page. For example, you might have some paragraphs of text that contain information that is more important than others and want to distinguish these elements, or you might want to differentiate between links that point to other pages on your own site and links that point to external sites.

To do this you can use the class attribute. Its value should describe the class it belongs to. In the example on the left, key paragraphs have a class attribute whose value is important.

The class attribute on any element can share the same value. So, in this example, the value of important could be used on headings and links, too.

If you would like to indicate that an element belongs to several classes, you can separate class names with a space, as you can see in the third paragraph in the example above.

BLOCK ELEMENTS

Some elements will always appear to start on a new line in the browser window. These are known as **block level** elements.



Examples of block elements are <h1>, <p>, , and .

chapter-08/block-elements.html

HTML

```
<h1>Hiroshi Sugimoto</h1>
<p>The dates for the ORIGIN OF ART exhibition are as follows:</p>
<ul>
  <li>Science: 21 Nov - 20 Feb 2010/11</li>
  <li>Architecture: 6 Mar - 15 May 2011</li>
  <li>History: 29 May - 21 Aug 2011</li>
  <li>Religion: 28 Aug - 6 Nov 2011</li>
</ul>
```

RESULT

Hiroshi Sugimoto

The dates for the ORIGIN OF ART exhibition are as follows:

- Science: 21 Nov - 20 Feb 2010/11
- Architecture: 6 Mar - 15 May 2011
- History: 29 May - 21 Aug 2011
- Religion: 28 Aug - 6 Nov 2011

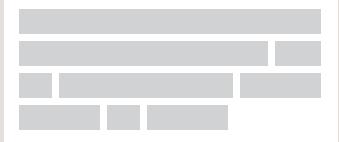
INLINE ELEMENTS

HTML

chapter-08/inline-elements.html

Timed to a single revolution of the planet around the sun at a 23.4 degrees tilt that plays out the rhythm of the seasons, this `Origins of Art` cycle is organized around four themes: `science, architecture, history` and `religion`.

Some elements will always appear to continue on the same line as their neighbouring elements. These are known as **inline** elements.



Examples of inline elements are `<a>`, ``, ``, and ``.

RESULT

Timed to a single revolution of the planet around the sun at a 23.4 degrees tilt that plays out the rhythm of the seasons, this *Origins of Art* cycle is organized around four themes: **science, architecture, history and religion**.

GROUPING TEXT & ELEMENTS IN A BLOCK

<div>

The `<div>` element allows you to group a set of elements together in one block-level box.

For example, you might create a `<div>` element to contain all of the elements for the header of your site (the logo and the navigation), or you might create a `<div>` element to contain comments from visitors.

In a browser, the contents of the `<div>` element will start on a new line, but other than this it will make no difference to the presentation of the page.

Using an `id` or `class` attribute on the `<div>` element, however, means that you can create CSS style rules to indicate how much space the `<div>` element should occupy on the screen and change the appearance of all the elements contained within it.

It can also make it easier to follow your code if you have used `<div>` elements to hold each section of the page.

chapter-08/grouping-block-elements.html

HTML

```
<div id="header">
  
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="biography.html">Biography</a></li>
    <li><a href="works.html">Works</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</div><!-- end of header -->
```

RESULT



- [Home](#)
- [Biography](#)
- [Works](#)
- [Contact](#)

Since there may be several other elements inside a `<div>` element, it can be helpful to add a comment after the closing `</div>` tag.

This allows you to clearly see which opening tag it is supposed to correspond to, as shown at the end of the example here.

GROUPING TEXT & ELEMENTS INLINE

HTML

chapter-08/grouping-inline-elements.html

```
<p>Anish Kapoor won the Turner Prize in 1991 and  
exhibited at the <span class="gallery">Tate  
Modern</span> gallery in London in 2003.</p>
```

RESULT

Anish Kapoor won the Turner Prize in 1991 and exhibited at the
TATE MODERN gallery in London in 2003.

The `` element acts like an inline equivalent of the `<div>` element. It is used to either:

1. Contain a section of text where there is no other suitable element to differentiate it from its surrounding text
2. Contain a number of inline elements

The most common reason why people use `` elements is so that they can control the appearance of the content of these elements using CSS.

You will usually see that a `class` or `id` attribute is used with `` elements:

- To explain the purpose of this `` element
- So that CSS styles can be applied to elements that have specific values for these attributes

ESCAPE CHARACTERS

There are some characters that are used in and reserved by HTML code. (For example, the left and right angled brackets.)

Therefore, if you want these characters to appear on your page you need to use what are termed "escape" characters (also known as escape codes or entity references). For example, to write a left angled bracket, you can use either < or <. For an ampersand, you can use either & or &.

There are also special codes that can be used to show symbols such as copyright and trademark, currency symbols, mathematical characters, and some punctuation marks. For example, if you want to include a copyright symbol on a web page you can use either © or ©.

When using escape characters, it is important to check the page in your browser to ensure that the correct symbol shows up. This is because some fonts do not support all of these characters and you might therefore need to specify a different font for these characters in your CSS code.

ONLINE EXTRA

You can find a more complete list of escape codes in the tools section of the website accompanying this book.

<	Less-than sign < amp;#60;	¢	Cent sign ¢ amp;#162;	'	Left single quote ‘ amp;#8216;
>	Greater-than sign > amp;#62;	£	Pound sign £ amp;#163;	'	Right single quote ’ amp;#8217;
&	Ampersand & amp;#38;	¥	Yen sign ¥ amp;#165;	"	Left double quotes “ amp;#8220;
"	Quotation mark " amp;#34;	€	Euro sign € amp;#8364;	"	Right double quotes ” amp;#8221;
(C)	Copyright symbol © amp;#169;	X	Multiplication sign × amp;#215;		
(R)	Registered trademark ® amp;#174;	÷	Division sign ÷ amp;#247;		
TM	Trademark ™ amp;#8482;				

SUMMARY

EXTRA MARKUP

- ▶ DOCTYPES tell browsers which version of HTML you are using.
- ▶ You can add comments to your code between the <!-- and --> markers.
- ▶ The id and class attributes allow you to identify particular elements.
- ▶ The <div> and elements allow you to group block-level and inline elements together.
- ▶ <iframes> cut windows into your web pages through which other pages can be displayed.
- ▶ The <meta> tag allows you to supply all kinds of information about your web page.
- ▶ Escape characters are used to include special characters in your pages such as <, >, and ©.