



# 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](http://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.

# CREDITS

For John Wiley & Sons, Inc.

EXECUTIVE EDITOR

Carol Long

MARKETING MANAGER

Ashley Zurcher

PRODUCTION MANAGER

Tim Tate

PRODUCTION EDITOR

Daniel Scribner

VICE PRESIDENT AND  
EXECUTIVE GROUP PUBLISHER

Richard Swadley

VICE PRESIDENT AND  
EXECUTIVE PUBLISHER

Barry Pruett

ASSOCIATE PUBLISHER

Jim Minatel

PRODUCTION COORDINATOR,  
COVER

Katie Crocker

AUTHOR

Jon Duckett

COVER DESIGNER

Emme Stone

DESIGN AND LAYOUT

Jon Duckett

Emme Stone

TECHNICAL EDITOR

Chris Mills

TECHNICAL REVIEWERS

Andy Stone

Angela Shimell

Donna Watson

Martin Callanan

Rob Jacoby

Tony Berry

PHOTOGRAPHY

John Stewardson

[johnstewardson.com](http://johnstewardson.com)

ADDITIONAL PHOTOGRAPHY

Hesperian

Joe Robertson

[flickr.com/photos/mindfire](http://flickr.com/photos/mindfire)

Jules Clancy

[thestonesoup.com](http://thestonesoup.com)

Kylie Gusset

[gusset.net](http://gusset.net)

Michael Stillwell

[beebo.org](http://beebo.org)

**Try out and download all of the code for this book online at:**  
[\*\*http://www.htmlandcssbook.com/code/\*\*](http://www.htmlandcssbook.com/code/)

# 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



# INTRODUCTION

- ▶ About this book
- ▶ How the web works
- ▶ Learning from other pages

Firstly, thank you for picking up this book. It has been written with two very different types of people in mind:

- Those who want to learn how to design and build websites from scratch
- Anyone who has a website (that may be built using a content management system, blogging software, or an e-commerce platform) and wants more control over the appearance of their pages

The only things you need in order to use this book are a computer with a web browser and a text editor (such as Notepad, which comes with Windows, orTextEdit, which comes with Macs).



**Introduction** pages come at the beginning of each chapter. They introduce the key topics you will learn about.

**Background** pages appear on white; they explain the context of the topics covered that are discussed in each chapter.

**Example** pages put together the topics you have learned and demonstrate how they can be applied in each.

**Reference** pages introduce key pieces of HTML & CSS code. The HTML code is shown in blue and CSS code is shown in pink.

**Diagram** and infographics pages are shown on a dark background. They provide a simple, visual reference to topics discussed.

**Summary** pages come at the end of each chapter. They remind you of the key topics that were covered in each chapter.

# IS IT HARD TO LEARN?

Many books that teach HTML and CSS resemble dull manuals. To make it easier for you to learn, we threw away the traditional template used by publishers and redesigned this book from scratch.

At work, when people look at my screen and see it full of code, it's not unusual to get a comment about it looking very complicated or how clever I must be to understand it. The truth is, it's not that hard to learn how to write web pages and read the code used to create them; you certainly don't have to be a "programmer."

Understanding HTML and CSS can help anyone who works with the web; designers can create more attractive and usable sites, website editors can create better content, marketers can communicate with their audience more effectively, and managers can commission better sites and get the best out of their teams.

I've focussed on the code you need to use 90% of the time and omitted the code that you would rarely see even if writing websites is your full time job. By the end of the book, if you come across the other 10% you will be able to Google it to find out what it means quickly and easily.

I have also added practical information on topics I am commonly asked about, such as how to prepare images, audio and video for the web, how to approach the design and build of a new site, how to improve your rankings in search engines (SEO), and how to use Google Analytics to learn about visitors to your site.

# THE STRUCTURE OF THIS BOOK

In order to teach you about creating web pages, this book is divided into three sections:

## 1: HTML

We will spend the first chapter looking at how HTML is used to create web pages. You will see that you start by writing down the words you want to appear on your page. You then add tags or elements to the words so that the browser knows what is a heading, where a paragraph begins and ends, and so on.

The rest of this section introduces the tags you have at your disposal to create web pages, grouped into chapters on: text, lists, links, images, tables, forms, video audio and flash, and miscellaneous elements.

I should warn you that the examples in the first nine chapters are not exciting to look at, yet they are the foundation of every web page. The following chapters on CSS will show you how to make your pages look a lot more interesting.

## 2: CSS

We start this section with a chapter that explains how CSS uses rules to enable you to control the styling and layout of web pages. We then go on to look at the wide variety of CSS properties you can use in your CSS rules. These properties generally fall into one of two categories:

**Presentation:** How to control things like the color of text, the fonts you want to use and the size of those fonts, how to add background colors to pages (or parts of a page), and how to add background images.

**Layout:** How to control where the different elements are positioned on the screen. You will also learn several techniques that professionals use to make their pages more attractive.

## 3: PRACTICAL

We end up with some helpful information that will assist you in building better websites.

We look at some new tags that will be introduced in HTML5 to help describe the structure of your pages. HTML5 is the latest version of HTML (still under development at the time of writing). Before learning about these elements, you need a good grasp of how CSS is used to control the design of web pages. There is a chapter that talks you through a design process that you might like to follow when creating a new website.

Finally, we end up looking at topics that will help you once you have built your site, such as putting it on the web, search engine optimisation (SEO) and using analytics software to track who comes to your site and what they are looking at.

# HOW PEOPLE ACCESS THE WEB

Before we look at the code used to build websites it is important to consider the different ways in which people access the web and clarify some terminology.

## BROWSERS

People access websites using software called a **web browser**. Popular examples include Firefox, Internet Explorer, Safari, Chrome, and Opera.

In order to view a web page, users might type a web address into their browser, follow a link from another site, or use a bookmark.

Software manufacturers regularly release new versions of browsers with new features and supporting new additions to languages. It is important, however, to remember that many computer owners will not be running the latest versions of these browsers. Therefore you cannot rely on all visitors to your site being able to use the latest functionality offered in all browsers.

You will learn how to tell which browsers visitors use to access your website in Chapter 19.

## WEB SERVERS

When you ask your browser for a web page, the request is sent across the Internet to a special computer known as a **web server** which hosts the website.

Web servers are special computers that are constantly connected to the Internet, and are optimized to send web pages out to people who request them.

Some big companies run their own web servers, but it is more common to use the services of a **web hosting** company who charge a fee to host your site.

## DEVICES

People are accessing websites on an increasing range of devices including desktop computers, laptops, tablets, and mobile phones. It is important to remember that various devices have different screen sizes and some have faster connections to the web than others.

## SCREEN READERS

**Screen readers** are programs that read out the contents of a computer screen to a user. They are commonly used by people with visual impairments.

In the same way that many countries have legislations that require public buildings to be accessible to those with disabilities, many laws have also been passed that require websites be accessible to those with disabilities.

Throughout this book you will see several references to screen readers. These notes will help ensure that the sites you create are accessible to people who use such software.

It is interesting to note that technologies similar to those employed by screen readers are also being used in other areas where people are unable to read a screen, such as when they are driving or jogging.

# 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

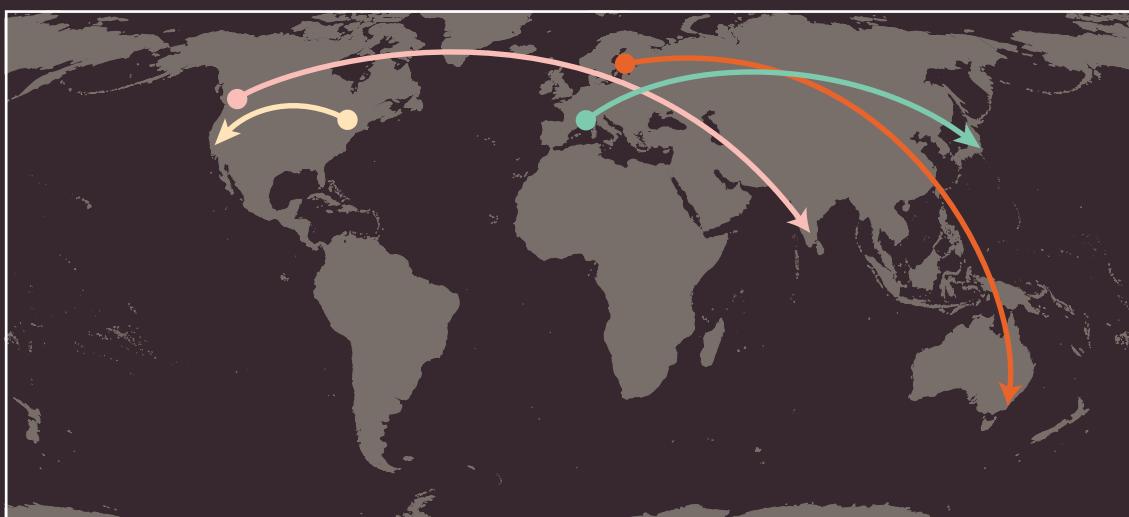
Since the web was first created there have been several versions of HTML and CSS — each intended to be an improvement on the previous version.

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](http://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

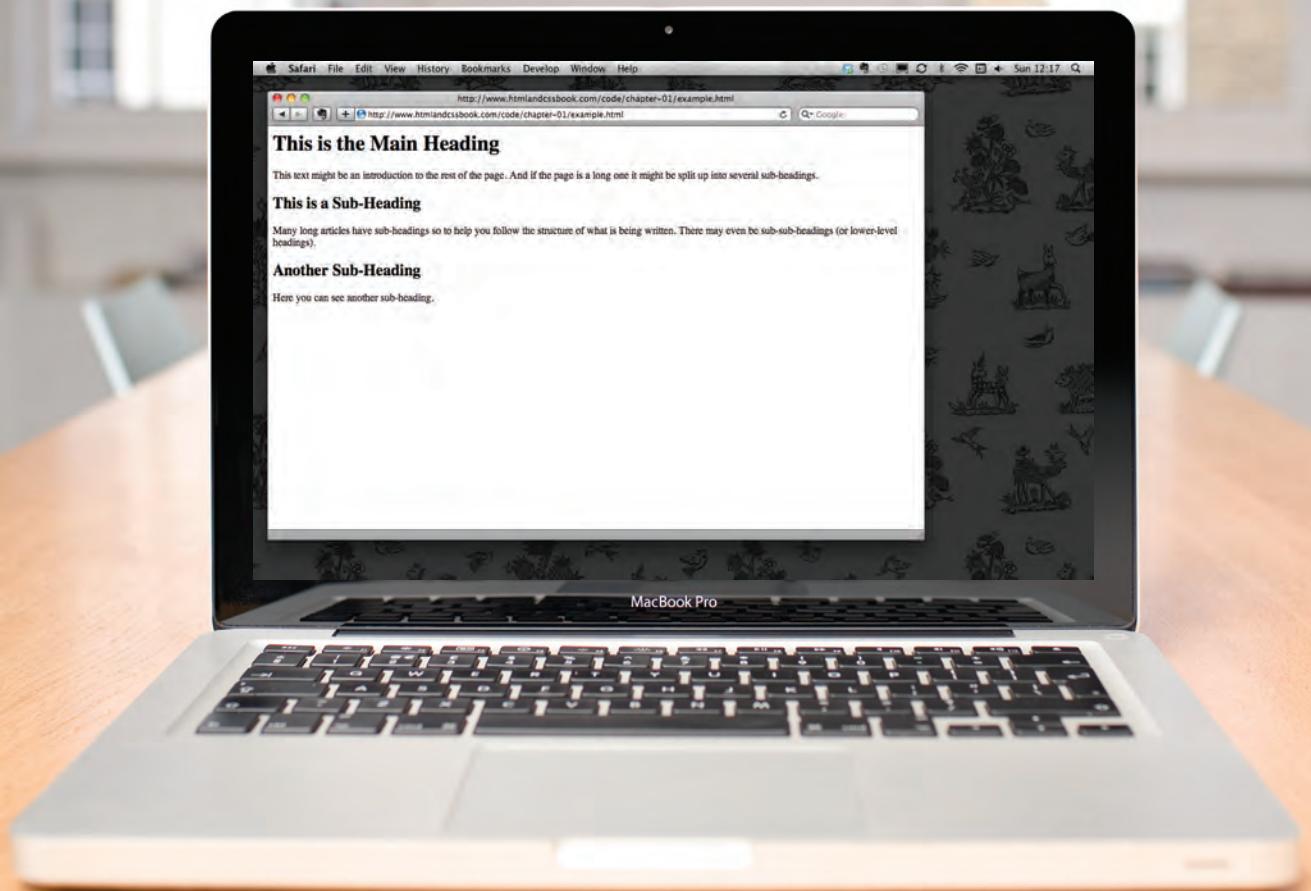
- ▶ Understanding structure
- ▶ Learning about markup
- ▶ Tags and elements

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 negative attitudes, 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 its peak 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 just treat 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

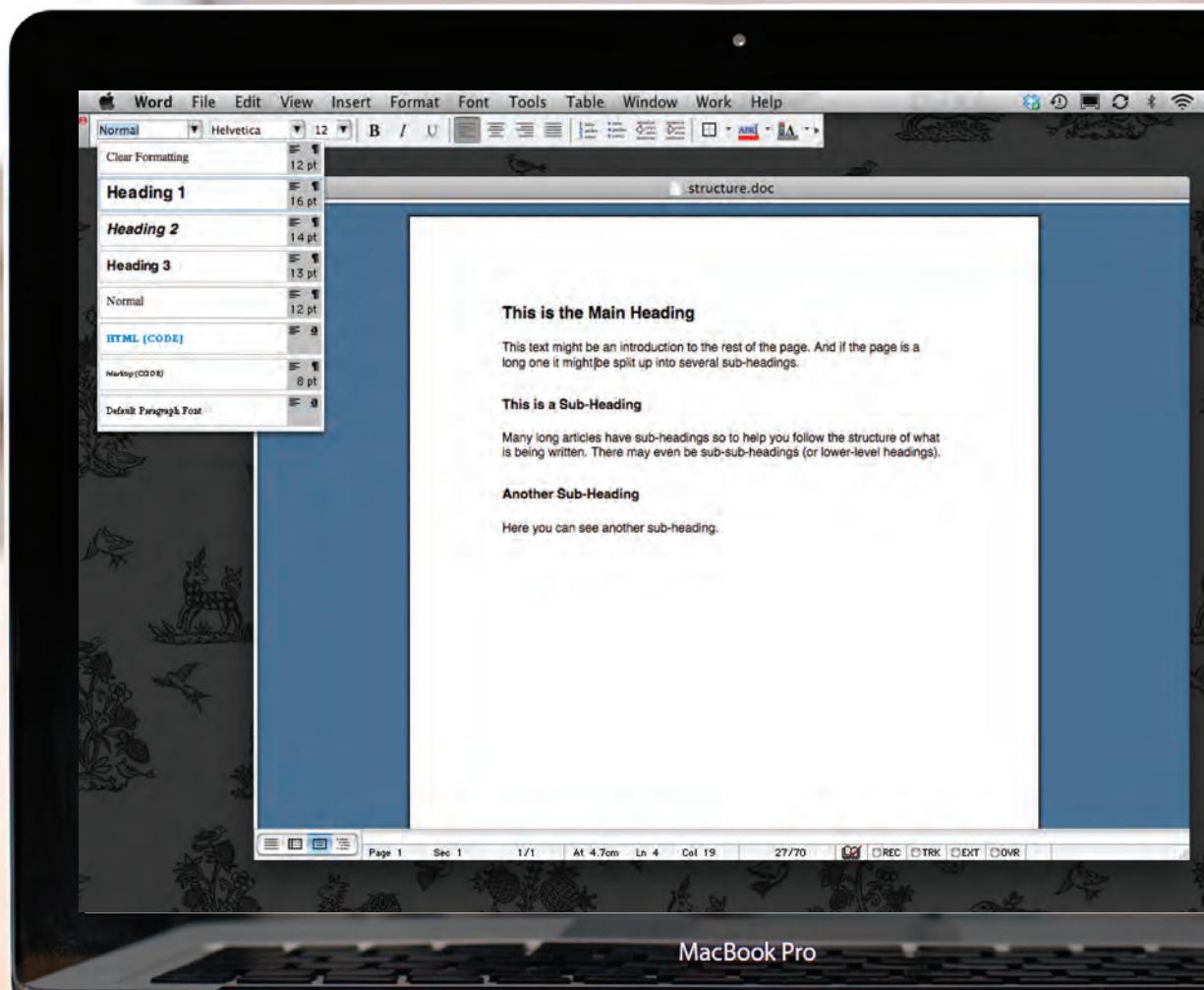


# STRUCTURING WORD DOCUMENTS

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 < 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

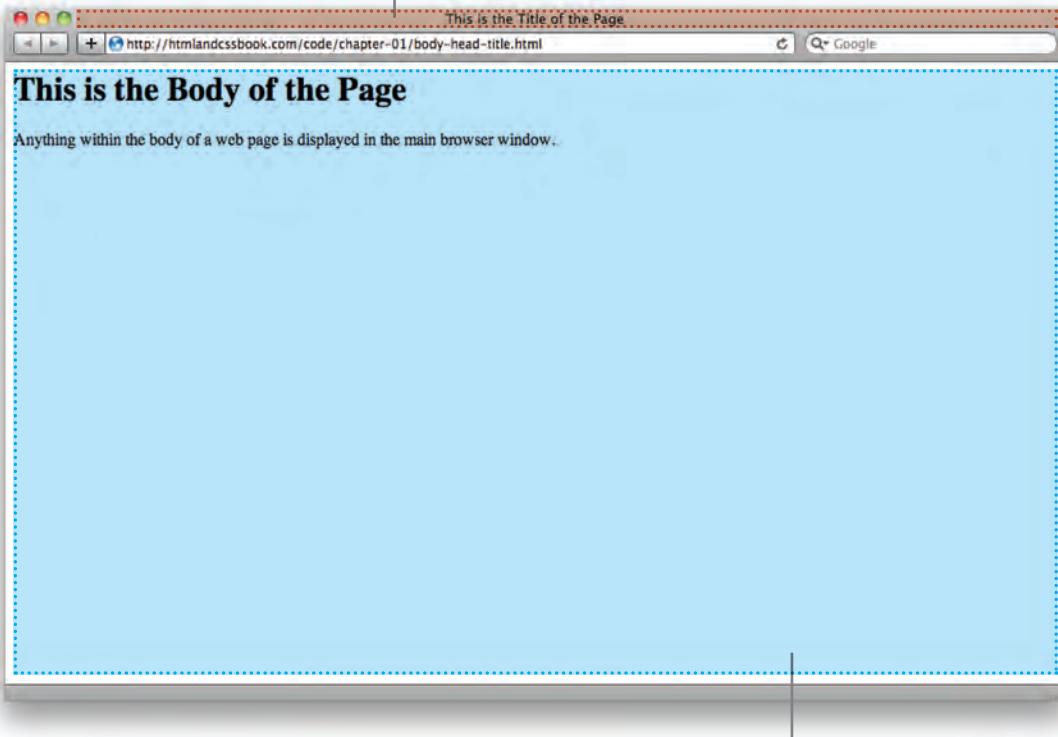
```
<html>
  <head>
    <title>This is the Title of the Page</title>
  </head>
  <body>
    <h1>This is the Body of the Page</h1>
    <p>Anything within the body of a web page is
       displayed in the main browser window.</p>
  </body>
</html>
```

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

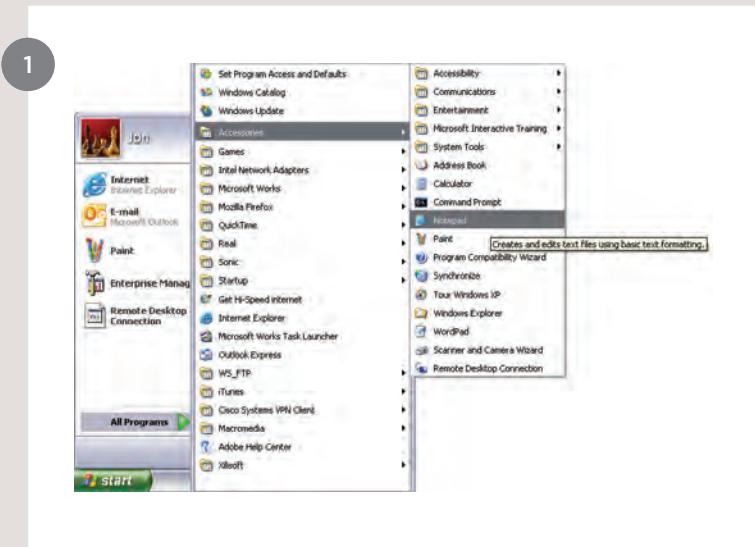
page, we add code around the original text we want to display and the browser then uses the code to display the page correctly. So the tags we add are the markup.

# 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:

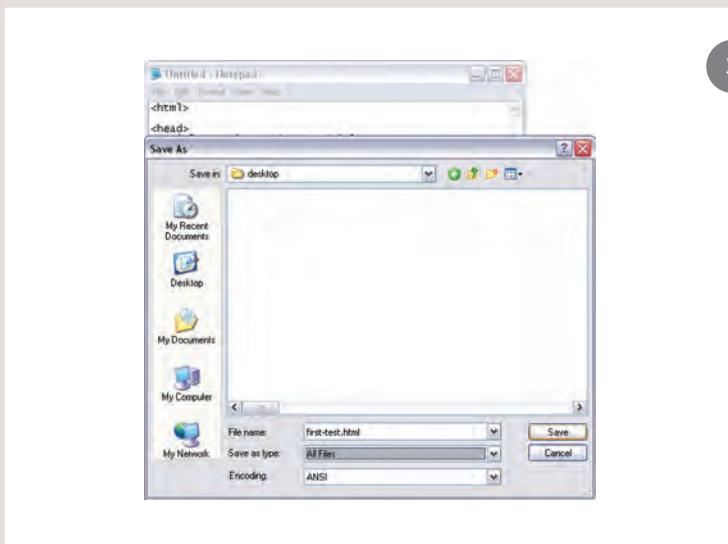
**Start**  
**All Programs (or Programs)**  
**Accessories**  
**Notepad**

You might also like to download a free editor called Notepad++ from [notepad-plus-plus.org](http://notepad-plus-plus.org).



Type the code shown on the right.

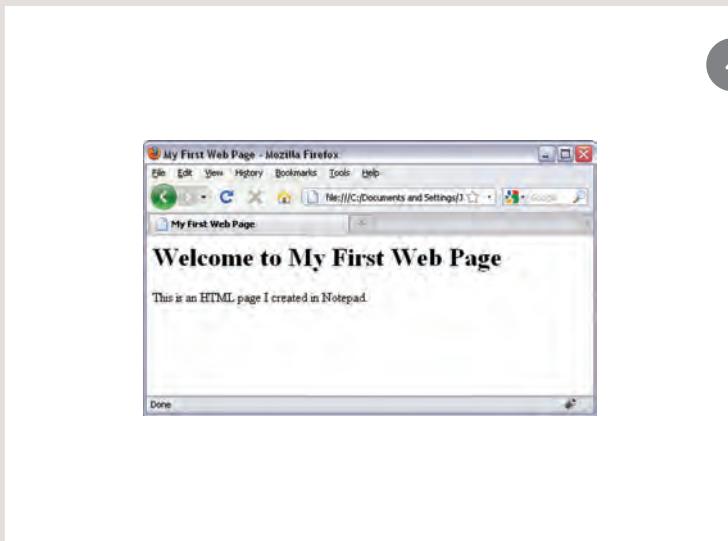




3

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.

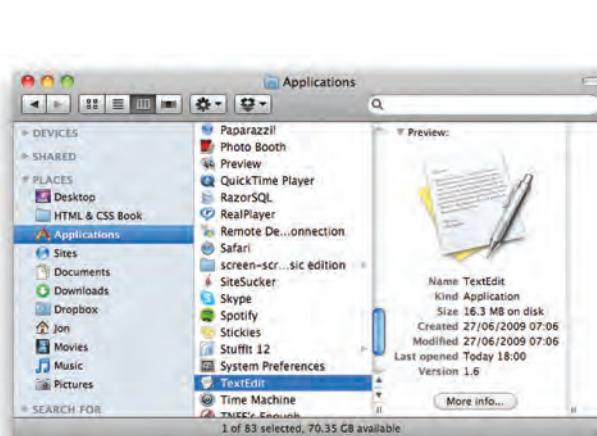
If it doesn't look like this, find the file you just created on your computer and make sure that it has the file extension **.html** (if it is **.txt** then you need to go back to Notepad and save the file again, but this time put quote marks around the name "**first-test.html**").

# 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](http://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.

# CODE IN A CONTENT MANAGEMENT SYSTEM



If you are working with a content management system, blogging platform, or e-commerce application, you will probably log into a special administration section of the website to control it. The tools provided in the administration sections of these sites usually allow you to edit parts of the page rather than the entire page, which means you will rarely see the `<html>`, `<head>`, or `<body>` elements.

Looking at the content management system on the opposite page, you have a box

that allows you to enter a title for the page, another box for the main article, a way to enter a publication date, and something to indicate which section of the site this page belongs in.

For an e-commerce store, you might have boxes that allow you to enter a title for the product, a description of the product, its price, and the quantity available.

That is because they use a single 'template' to control all of the pages for a section of the site. (For example, an e-commerce

system might use the same template to show all of their products.) The information you supply is placed into the templates.

The advantage of this approach is that people who do not know how to write web pages can add information to a website and it is also possible to change the presentation of something in the template, and it will automatically update every page that uses that template. If you imagine an e-commerce store with 1,000 items for sale, just



altering one template is a lot easier than changing the page for each individual product. In systems like this, when you have a large block of text that you can edit, such as a news article, blog entry or the description of a product in an e-commerce store, you will often see a text editor displayed.

Text editors usually have controls a little like those on your word processor, giving you different options to style text, add links or insert images. Behind the scenes these editors

are adding HTML code to your text, just like the code you have seen earlier in this chapter. Many of these editors will have an option that allows you to see (and edit) the code that they produce.

Once you know how to read and edit this code, you can take more control over these sections of your website.

In the example above, you can see that the text editor has a tab for Visual / HTML views of what the user enters. Other systems

might have a button (which often shows angle brackets) to indicate how to access the code.

Some content management systems offer tools that also allow you to edit the template files. If you do try to edit template files you need to check the documentation for your CMS as they all differ from each other. You need to be careful when editing template files because if you delete the wrong piece of code or add something in the wrong place the site may stop working entirely.

# 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/](http://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

## <b>

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

The `<b>` 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 `<b>` 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</i>.
<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 4<sup>th</sup> of September you will learn about E=MC<sup>2</sup>.

The amount of CO<sub>2</sub> in the atmosphere grew by 2ppm in 2009<sub>1</sub>.

## <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 2<sup>2</sup>.

## <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 H<sub>2</sub>O.

# 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.

<br />

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 <br />.

## 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

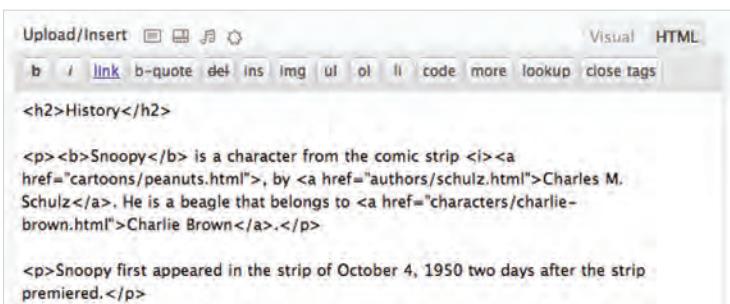
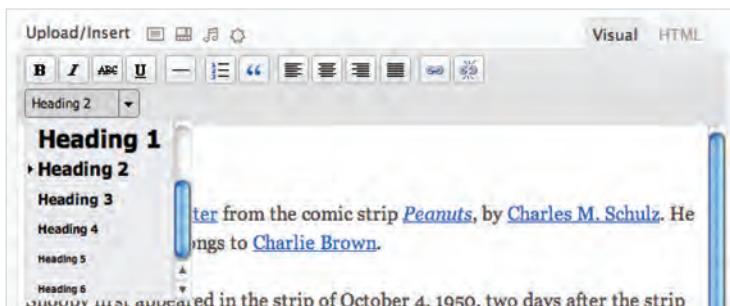
Content management systems and HTML editors such as Dreamweaver usually have two views of the page you are creating: a visual editor and a code view.

**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 `<em>` 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 `<em>` 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 `<em>` element, or a search engine might register that your page features a quote if you use the `<blockquote>` element.

# STRONG & EMPHASIS

## <strong>

The use of the `<strong>` 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 `<strong>` 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>
```

RESULT

**Beware: Pickpockets operate in this area.**

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

## <em>

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

By default browsers will show the contents of an `<em>` 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>
```

RESULT

*I think Ivy was the first.*

*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](mailto: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>  
<del>**

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

The content of a `<ins>` element is usually underlined, while the content of a `<del>` 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 `<em>`.

```
<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).

