



# 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/>

# 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

# 3

## LISTS

- ▶ Numbered lists
- ▶ Bullet lists
- ▶ Definition lists

There are lots of occasions when we need to use lists. HTML provides us with three different types:

- **Ordered lists** are lists where each item in the list is numbered. For example, the list might be a set of steps for a recipe that must be performed in order, or a legal contract where each point needs to be identified by a section number.
- **Unordered lists** are lists that begin with a bullet point (rather than characters that indicate order).
- **Definition lists** are made up of a set of terms along with the definitions for each of those terms.

The image shows a silver MacBook Pro laptop open on a light-colored wooden desk. The screen displays a web page titled "Scrambled Eggs" from the URL <http://www.htmlandcssbook.com/code/Chapter-03/example.html>. The page content includes:

## Scrambled Eggs

Eggs are one of my favorite foods. Here is a recipe for deliciously rich scrambled eggs.

### Ingredients

- 2 eggs
- 1lb butter
- 2oz cream

### Method

1. Melt butter in a frying pan over a medium heat
2. Gently mix the eggs and cream in a bowl
3. Pour the eggs into the melted butter
4. Using a spatula fold the eggs from the edge of the pan to the center every 20 seconds (as if you are making an omelette)
5. When the eggs are still moist remove from the heat (it will continue to cook on the plate until served)

# ORDERED LISTS

## <ol>

The ordered list is created with the <ol> element.

## <li>

Each item in the list is placed between an opening <li> tag and a closing </li> tag. (The li stands for list item.)

Browsers indent lists by default.

Sometimes you may see a type attribute used with the <ol> element to specify the type of numbering (numbers, letters, roman numerals and so on). It is better to use the CSS list-style-type property covered on pages 333-335.

chapter-03/ordered-lists.html

HTML

```
<ol>
  <li>Chop potatoes into quarters</li>
  <li>Simmer in salted water for 15-20
      minutes until tender</li>
  <li>Heat milk, butter and nutmeg</li>
  <li>Drain potatoes and mash</li>
  <li>Mix in the milk mixture</li>
</ol>
```

RESULT

1. Chop potatoes into quarters
2. Simmer in salted water for 15-20 minutes until tender
3. Heat milk, butter and nutmeg
4. Drain potatoes and mash
5. Mix in the milk mixture

# UNORDERED LISTS

## HTML

chapter-03/unordered-lists.html

```
<ul>
  <li>1kg King Edward potatoes</li>
  <li>100ml milk</li>
  <li>50g salted butter</li>
  <li>Freshly grated nutmeg</li>
  <li>Salt and pepper to taste</li>
</ul>
```

## RESULT

- 1kg King Edward potatoes
- 100ml milk
- 50g salted butter
- Freshly grated nutmeg
- Salt and pepper to taste

<ul>

The unordered list is created with the <ul> element.

<li>

Each item in the list is placed between an opening <li> tag and a closing </li> tag. (The li stands for list item.)

Browsers indent lists by default.

Sometimes you may see a type attribute used with the <ul> element to specify the type of bullet point (circles, squares, diamonds and so on). It is better to use the CSS list-style-type property covered on pages 333-335.

# DEFINITION LISTS

## <dl>

The definition list is created with the `<dl>` element and usually consists of a series of terms and their definitions.

Inside the `<dl>` element you will usually see pairs of `<dt>` and `<dd>` elements.

## <dt>

This is used to contain the term being defined (the definition term).

## <dd>

This is used to contain the definition.

Sometimes you might see a list where there are two terms used for the same definition or two different definitions for the same term.

chapter-03/definition-lists.html

HTML

```
<dl>
  <dt>Sashimi</dt>
  <dd>Sliced raw fish that is served with condiments such as shredded daikon radish or ginger root, wasabi and soy sauce</dd>
  <dt>Scale</dt>
  <dd>A device used to accurately measure the weight of ingredients</dd>
  <dd>A technique by which the scales are removed from the skin of a fish</dd>
  <dt>Scamorze</dt>
  <dt>Scamorzo</dt>
  <dd>An Italian cheese usually made from whole cow's milk (although it was traditionally made from buffalo milk)</dd>
</dl>
```

RESULT

### Sashimi

Sliced raw fish that is served with condiments such as shredded daikon radish or ginger root, wasabi and soy sauce

### Scale

A device used to accurately measure the weight of ingredients

A technique by which the scales are removed from the skin of a fish

### Scamorze

### Scamorzo

An Italian cheese usually made from whole cow's milk (although it was traditionally made from buffalo milk)

# NESTED LISTS

## HTML

chapter-03/nested-lists.html

```
<ul>
  <li>Mousses</li>
  <li>Pastries
    <ul>
      <li>Croissant</li>
      <li>Mille-feuille</li>
      <li>Palmier</li>
      <li>Profiterole</li>
    </ul>
  </li>
  <li>Tarts</li>
</ul>
```

You can put a second list inside an `<li>` element to create a sublist or nested list.

Browsers display nested lists indented further than the parent list. In nested unordered lists, the browser will usually change the style of the bullet point too.

## RESULT

- Mousses
- Pastries
  - Croissant
  - Mille-feuille
  - Palmier
  - Profiterole
- Tarts

A close-up photograph of an open MacBook Pro laptop. The screen displays a web page from <http://www.htmlandcssbook.com/code/chapter-03/example.html> titled "Scrambled Eggs". The page contains a short introduction, a section titled "Ingredients" with a bulleted list, and a section titled "Method" with a numbered list of steps. The laptop is resting on a light-colored surface, and its floral patterned keyboard cover is visible.

Safari File Edit View History Bookmarks Develop Window Help

Lists

http://www.htmlandcssbook.com/code/chapter-03/example.html Google

## Scrambled Eggs

Eggs are one of my favorite foods. Here is a recipe for deliciously rich scrambled eggs.

### Ingredients

- 2 eggs
- 1tbs butter
- 2tbs cream

### Method

1. Melt butter in a frying pan over a medium heat
2. Gently mix the eggs and cream in a bowl
3. Once butter has melted add cream and eggs
4. Using a spatula fold the eggs from the edge of the pan to the center every 20 seconds (as if you are making an omelette)
5. When the eggs are still moist remove from the heat (it will continue to cook on the plate until served)

MacBook Pro

# EXAMPLE

## LISTS



Here you can see a main heading followed by an introductory paragraph. An unordered list is used to outline the ingredients and an ordered list is used to describe the steps.

```
<html>
  <head>
    <title>Lists</title>
  </head>
  <body>
    <h1>Scrambled Eggs</h1>
    <p>Eggs are one of my favourite foods. Here is a recipe for deliciously rich scrambled eggs.</p>
    <h2>Ingredients</h2>
    <ul>
      <li>2 eggs</li>
      <li>1tbs butter</li>
      <li>2tbs cream</li>
    </ul>
    <h2>Method</h2>
    <ol>
      <li>Melt butter in a frying pan over a medium heat</li>
      <li>Gently mix the eggs and cream in a bowl</li>
      <li>Once butter has melted add cream and eggs</li>
      <li>Using a spatula fold the eggs from the edge of the pan to the center every 20 seconds (as if you are making an omelette)</li>
      <li>When the eggs are still moist remove from the heat (it will continue to cook on the plate until served)</li>
    </ol>
  </body>
</html>
```



# SUMMARY

## LISTS

- ▶ There are three types of HTML lists: ordered, unordered, and definition.
- ▶ Ordered lists use numbers.
- ▶ Unordered lists use bullets.
- ▶ Definition lists are used to define terminology.
- ▶ Lists can be nested inside one another.



# 4

## LINKS

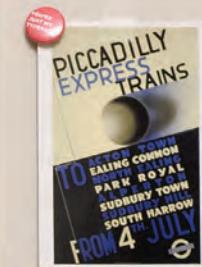
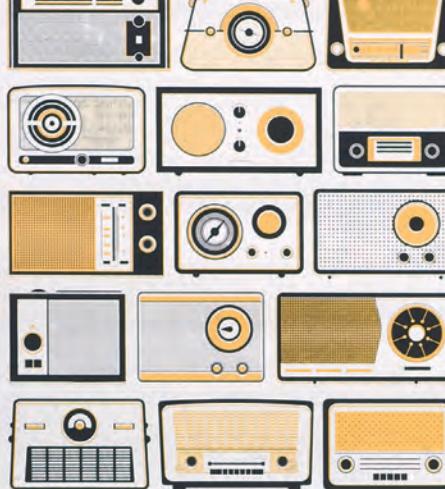
- ▶ Creating links between pages
- ▶ Linking to other sites
- ▶ Email links



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



Le Radioblog  
Archives  
RSS

New disk sheet  
- Reprise  
- in day

AN pach  
for day

**Film Folk**

**Festival Diary**

Here are some of the film festivals we will be attending this year.  
Please [contact us](#) if you would like more information.

**January**

[Sundance Film Festival](#)  
Park City, Utah, USA  
20 - 30 January 2011

**February**

[Tribeca](#)  
Sydney, Australia  
20 February 2011

**March**

[South by Southwest](#)  
Austin, Texas, USA  
11 - 20 March 2011

**April**

[London Independent Film Festival](#)  
London, UK  
15 - 24 April 2011

**May**

[Cannes International Film Festival](#)  
Cannes, France  
11 - 22 May 2011

[Done](#)



# WRITING LINKS

Links are created using the `<a>` element. Users can click on anything between the opening `<a>` tag and the closing `</a>` 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 `<a href="http://www.imdb.com">IMDB</a>` with annotations explaining its components:

- A bracket under the attribute `href="http://www.imdb.com"` is labeled "THIS IS THE PAGE THE LINK TAKES YOU TO".
- A bracket under the text "IMDB" is labeled "THIS IS THE TEXT THE USER CLICKS ON".
- A bracket under the entire tag `<a>` is labeled "OPENING LINK TAG".
- A bracket under the closing tag `</a>` is labeled "CLOSING LINK TAG".

The text between the opening <a> tag and closing </a> 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 `</a>` 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).

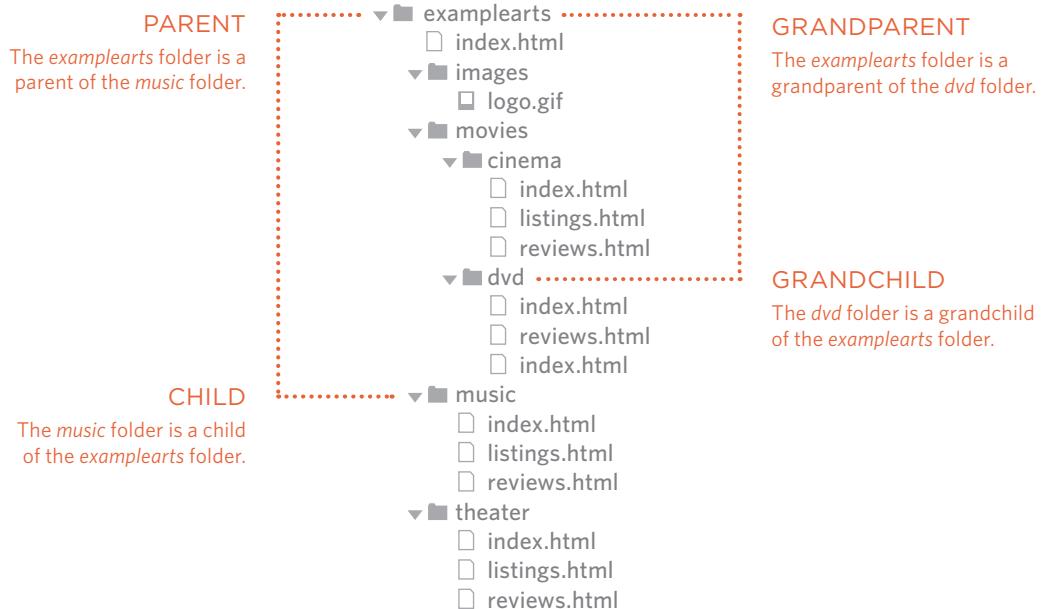
## HOME PAGES

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:  
`<a href="reviews.html">Reviews</a>`

### 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:  
`<a href="music/listings.html">Listings</a>`

### 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:  
`<a href="movies/dvd/reviews.html">Reviews</a>`

### 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:  
`<a href="..../index.html">Home</a>`

### 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:  
`<a href="...../index.html">Home</a>`

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.

# EMAIL LINKS

## mailto:

To create a link that starts up the user's email program and addresses an email to a specified email address, you use the `<a>` element. However, this time the value of the `href` attribute starts with `mailto:` and is followed by the email address you want the email to be sent to.

On the right you can see that an email link looks just like any other link but, when it is clicked on, the user's email program will open a new email message and address it to the person specified in the link.

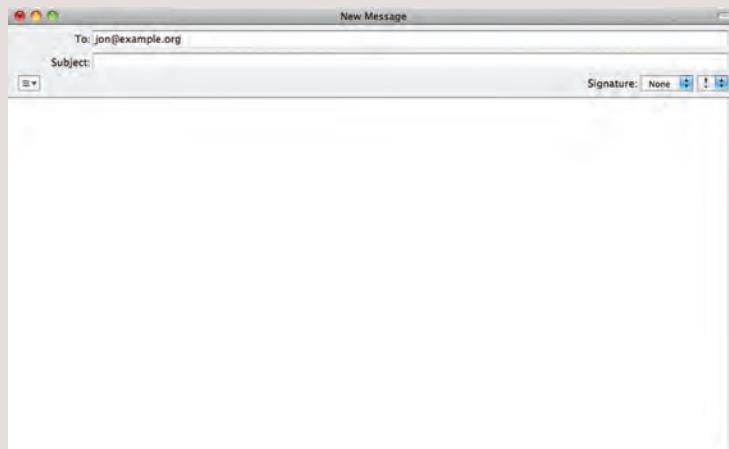
chapter-04/email-links.html

HTML

```
<a href="mailto:jon@example.org">Email Jon</a>
```

RESULT

[Email Jon](mailto:jon@example.org)



# OPENING LINKS IN A NEW WINDOW

## HTML

chapter-04/opening-links-in-a-new-window.html

```
<a href="http://www.imdb.com" target="_blank">  
Internet Movie Database</a> (opens in new window)
```

## RESULT

[Internet Movie Database](http://www.imdb.com) (opens in new window)

## target

If you want a link to open in a new window, you can use the target attribute on the opening `<a>` tag. The value of this attribute should be `_blank`.

One of the most common reasons a web page author might want a link to be opened in a new window is if it points to another website. In such cases, they hope the user will return to the window containing their site after finishing looking at the other one.

Generally you should avoid opening links in a new window, but if you do, it is considered good practice to inform users that the link will open a new window before they click on it.

# 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, `<a href="#top">` 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.

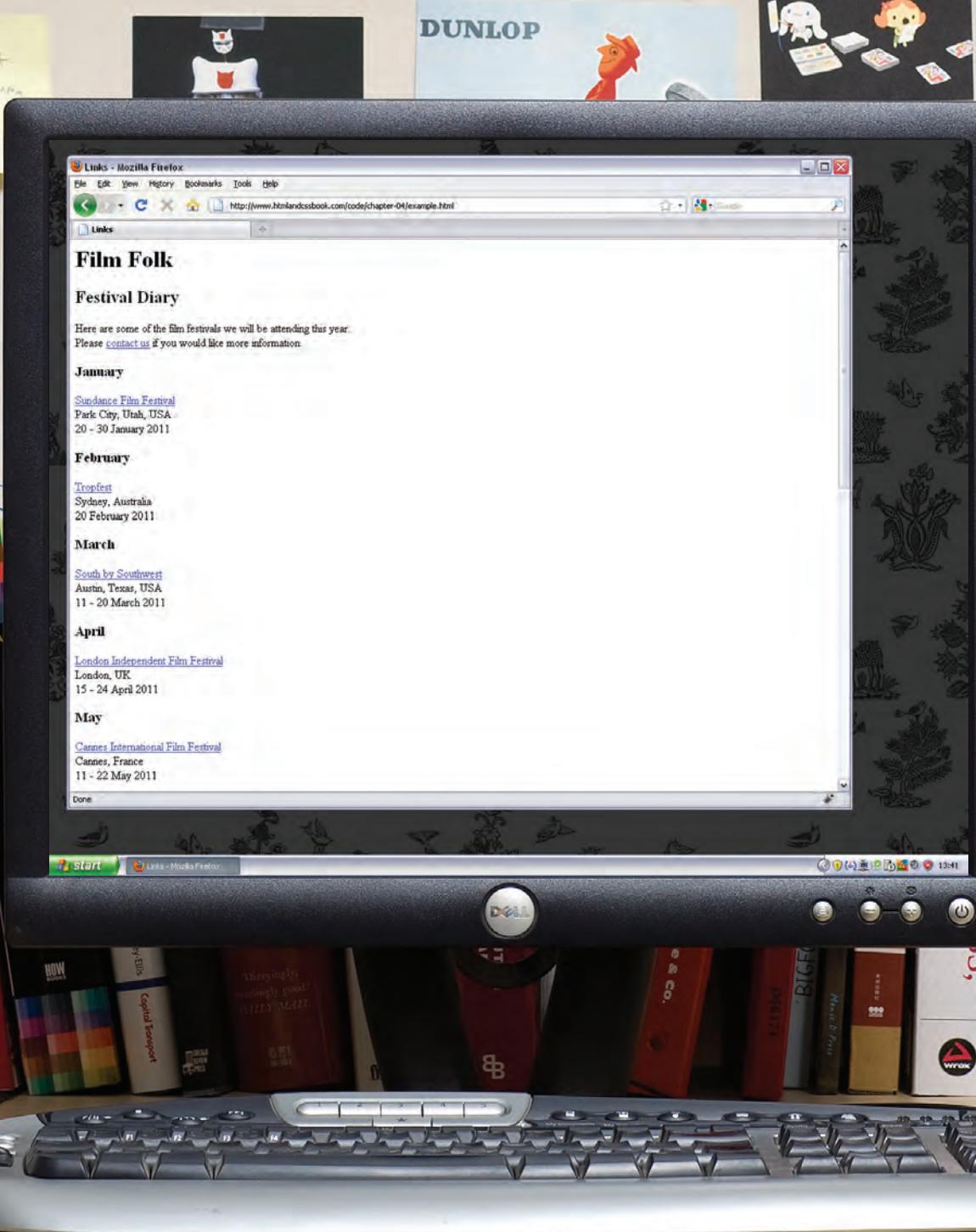
Therefore, the href attribute will contain the address for the page (either an absolute URL or a relative URL), followed by the # symbol, followed by the value of the id attribute that is used on the element you are linking to.

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">
```

Erin Charlotte trouve l'affiche à son goût. Elle la temporene avec sa pâte et la pose dans la nuit à réaliser des milliers d'affiches, au clair de lune.

style sheet  
- typefaces  
- ie browser



# EXAMPLE LINKS

This example is of a web page about film.

The `<h1>` element is used with an `id` attribute at the top of the page so that a link can be added to take readers from the bottom of the page to the top. There is an email link to allow readers to contact the author of the web page. There are also a number of links to qualified URLs. These link to various film festivals. Below this list is a link to a relative URL which is an "about" page that lives in the same directory.

```
<html>
  <head>
    <title>Links</title>
  </head>
  <body>
    <h1 id="top">Film Folk</h1>
    <h2>Festival Diary</h2>
    <p>Here are some of the film festivals we
      will be attending this year.<br />Please
      <a href="mailto:filmfolk@example.org">
        contact us</a> if you would like more
        information.</p>
    <h3>January</h3>
    <p><a href="http://www.sundance.org">
      Sundance Film Festival</a><br />
      Park City, Utah, USA<br />
      20 - 30 January 2011</p>
    <h3>February</h3>
    <p><a href="http://www.tropfest.com">
      Tropfest</a><br />
      Sydney, Australia<br />
      20 February 2011</p>
    <!-- additional content -->
    <p><a href="about.html">About Film Folk</a></p>
    <p><a href="#top">Top of page</a></p>
  </body>
</html>
```





# SUMMARY

## LINKS

- ▶ Links are created using the `<a>` element.
- ▶ The `<a>` element uses the `href` attribute to indicate the page you are linking to.
- ▶ If you are linking to a page within your own site, it is best to use relative links rather than qualified URLs.
- ▶ You can create links to open email programs with an email address in the "to" field.
- ▶ You can use the `id` attribute to target elements within a page that can be linked to.



# 5

## IMAGES

- ▶ How to add images to pages
- ▶ Choosing the right format
- ▶ Optimizing images for the web

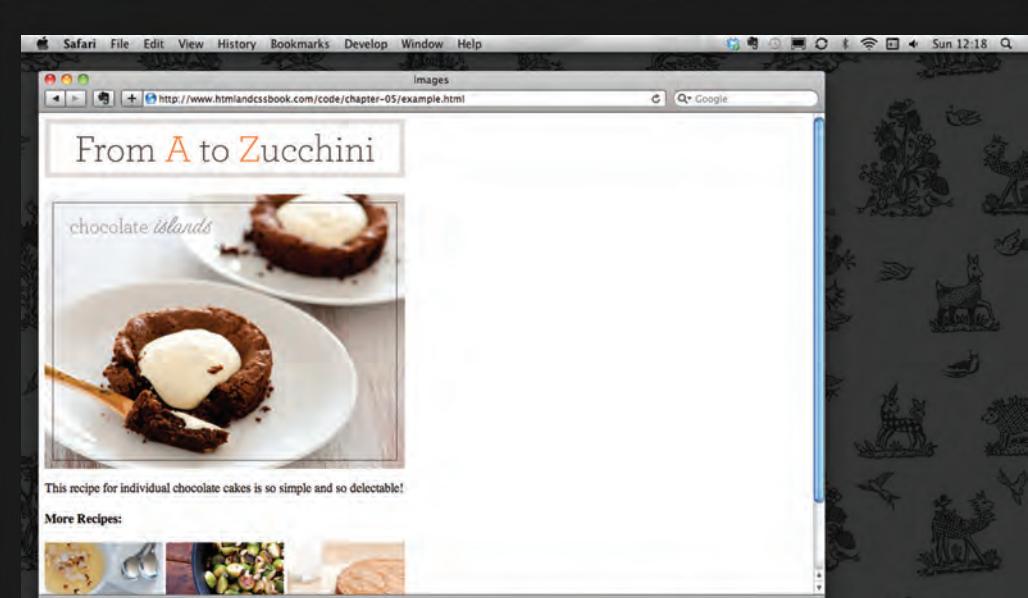
There are many reasons why you might want to add an image to a web page: you might want to include a logo, photograph, illustration, diagram, or chart.

There are several things to consider when selecting and preparing images for your site, but taking time to get them right will make it look more attractive and professional.

In this chapter you will learn how to:

- Include an image in your web pages using HTML
- Pick which image format to use
- Show an image at the right size
- Optimize an image for use on the web to make pages load faster

You can also use CSS to include images in your pages using the background-image property, which you will meet on pages 413-420.



# CHOOSING IMAGES FOR YOUR SITE

A picture can say a thousand words, and great images help make the difference between an average-looking site and a really engaging one.

Images can be used to set the tone for a site in less time than it takes to read a description. If you do not have photographs to use on your website, there are companies who sell **stock images**; these are images you

pay to use (there is a list of stock photography websites below). Remember that all images are subject to copyright, and you can get in trouble for simply taking photographs from another website.

If you have a page that shows several images (such as product photographs or members of a team) then putting them on a simple, consistent background helps them look better as a group.

## IMAGES SHOULD...

- ✓ Be relevant
- ✓ Convey information
- ✓ Convey the right mood
- ✓ Be instantly recognisable
- ✓ Fit the color palette

## STOCK PHOTOS

- [www.istockphoto.com](http://www.istockphoto.com)
- [www.gettyimages.com](http://www.gettyimages.com)
- [www.veer.com](http://www.veer.com)
- [www.sxc.hu](http://www.sxc.hu)
- [www.fotolia.com](http://www.fotolia.com)

## ONLINE EXTRA

We have provided an online gallery that helps you choose the right image for your website. You can find it in the tools section of the site accompanying this book.

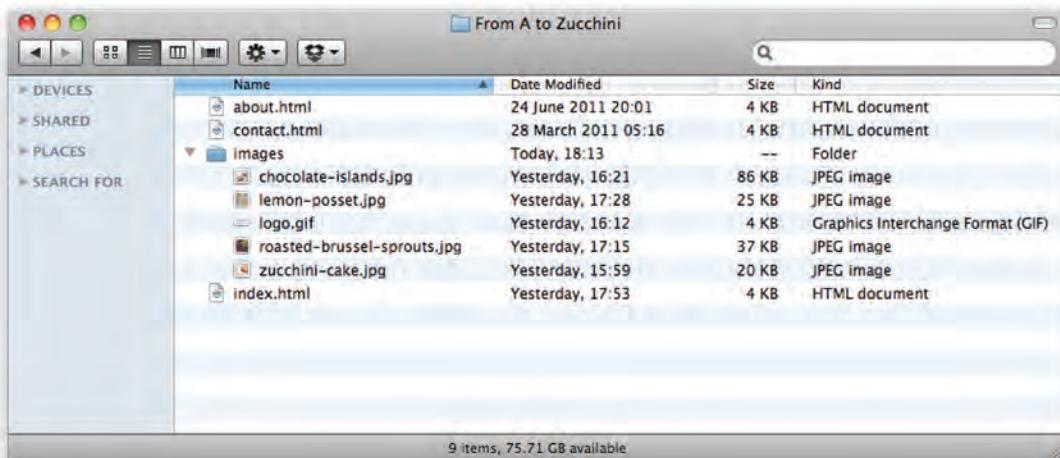
# STORING IMAGES ON YOUR SITE

If you are building a site from scratch, it is good practice to create a folder for all of the images the site uses.

As a website grows, keeping images in a separate folder helps you understand how the site is organized. Here you can see an example of the files for a website; all of the images are stored in a folder called **images**.

On a big site you might like to add subfolders inside the **images** folder. For example, images such as logos and buttons might sit in a folder called **interface**, product photographs might sit in a page called **products**, and images related to news might live in a folder called **news**.

If you are using a content management system or blogging platform, there are usually tools built into the admin site that allow you to upload images, and the program will probably already have a separate folder for image files and any other uploads.



# ADDING IMAGES

## <img>

To add an image into the page you need to use an <img> element. This is an empty element (which means there is no closing tag). It must carry the following two attributes:

### src

This tells the browser where it can find the image file. This will usually be a relative URL pointing to an image on your own site. (Here you can see that the images are in a child folder called **images** — relative URLs were covered on pages 83-84).

### alt

This provides a text description of the image which describes the image if you cannot see it.

### title

You can also use the title attribute with the <img> element to provide additional information about the image. Most browsers will display the content of this attribute in a tooltip when the user hovers over the image.

chapter-05/adding-images.html

HTML

```

```

RESULT



The text used in the alt attribute is often referred to as **alt text**. It should give an accurate description of the image content so it can be understood by screen reader software (used by people with visual impairments) and search engines.

If the image is just to make a page look more attractive (and it has no meaning, such as a graphic dividing line), then the alt attribute should still be used but the quotes should be left empty.

# HEIGHT & WIDTH OF IMAGES

## HTML

chapter-05/height-and-width-of-images.html

```

```

## RESULT



You will also often see an `<img>` element use two other attributes that specify its size:

### height

This specifies the height of the image in pixels.

### width

This specifies the width of the image in pixels.

Images often take longer to load than the HTML code that makes up the rest of the page. It is, therefore, a good idea to specify the size of the image so that the browser can render the rest of the text on the page while leaving the right amount of space for the image that is still loading.

The size of images is increasingly being specified using CSS rather than HTML — see pages 409–410 for more information about this.

# WHERE TO PLACE IMAGES IN YOUR CODE

Where an image is placed in the code will affect how it is displayed. Here are three examples of image placement that produce different results:

## 1: BEFORE A PARAGRAPH

The paragraph starts on a new line after the image.

## 2: INSIDE THE START OF A PARAGRAPH

The first row of text aligns with the bottom of the image.

## 3: IN THE MIDDLE OF A PARAGRAPH

The image is placed between the words of the paragraph that it appears in.

chapter-05/where-to-place-images.html

HTML

```

<p>There are around 10,000 living species of birds
    that inhabit different ecosystems from the
    Arctic to the Antarctic. Many species undertake
    long distance annual migrations, and many more
    perform shorter irregular journeys.</p>
<hr />
<p>There are around 10,000 living
    species of birds that inhabit different
    ecosystems from the Arctic to the Antarctic. Many
    species undertake long distance annual
    migrations, and many more perform shorter
    irregular journeys.</p>
<hr />
<p>There are around 10,000 living species of birds
    that inhabit different ecosystems from the
    Arctic to the Antarctic.Many species undertake long
    distance annual migrations, and many more perform
    shorter irregular journeys.</p>
```

## RESULT



There are around 10,000 living species of birds that inhabit different ecosystems from the Arctic to the Antarctic. Many species undertake long distance annual migrations, and many more perform shorter irregular journeys.



There are around 10,000 living species of birds that inhabit different ecosystems from the Arctic to the Antarctic. Many species undertake long distance annual migrations, and many more perform shorter irregular journeys.

There are around 10,000 living species of birds that inhabit different



ecosystems from the Arctic to the Antarctic. Many species undertake long distance annual migrations, and many more perform shorter irregular journeys.

Where you place the image in the code is important because browsers show HTML elements in one of two ways:

**Block elements always appear on a new line.** Examples of block elements include the `<h1>` and `<p>` elements.

If the `<img>` is followed by a block level element (such as a paragraph) then the block level element will sit on a new line after the image as shown in the first example on this page.

**Inline elements sit within a block level element and do not start on a new line.** Examples of inline elements include the `<b>`, `<em>`, and `<img>` elements.

If the `<img>` element is inside a block level element, any text or other inline elements will flow around the image as shown in the second and third examples on this page.

Block and inline elements are discussed in greater depth on pages 185-186.

# OLD CODE: ALIGNING IMAGES HORIZONTALLY

## align

The align attribute was commonly used to indicate how the other parts of a page should flow around an image. It has been removed from HTML5 and new websites should use CSS to control the alignment of images (as you will see on pages 411-412).

I have discussed it here because you are likely to come across it if you look at older code, and because some visual editors still insert this attribute when you indicate how an image should be aligned.

The align attribute can take these horizontal values:

### left

This aligns the image to the left (allowing text to flow around its right-hand side).

### right

This aligns the image to the right (allowing text to flow around its left-hand side).

chapter-05/aligning-images-horizontally.html

HTML

```
<p>There are around  
10,000 living species of birds that inhabit  
different ecosystems from the Arctic to the  
Antarctic. Many species undertake long distance  
annual migrations, and many more perform shorter  
irregular journeys.</p>  
<hr />  
<p>There are around  
10,000 living species of birds that inhabit  
different ecosystems from the Arctic to the  
Antarctic. Many species undertake long distance  
annual migrations, and many more perform shorter  
irregular journeys.</p>
```

## RESULT



There are around 10,000 living species of birds that inhabit different ecosystems from the Arctic to the Antarctic. Many species undertake long distance annual migrations, and many more perform shorter irregular journeys.

There are around 10,000 living species of birds that inhabit different ecosystems from the Arctic to the Antarctic. Many species undertake long distance annual migrations, and many more perform shorter irregular journeys.



This looks a lot neater than having one line of text next to the image (as shown on the previous example).

When you give the `align` attribute a value of `left`, the image is placed on the left and text flows around it.

When you give the `align` attribute a value of `right`, the image is placed on the right and the text flows around it.

When text flows right up to the edge of an image it can make it harder to read. You will learn how to add a gap between text and images on pages 313-314 using the CSS padding and margin properties.

# OLD CODE: ALIGNING IMAGES VERTICALLY

As you saw on the last page, the align attribute is no longer used in HTML5, but it is covered here because you may see it used in older websites and it is still used in the code created by some visual editors.

You can see how to use CSS to achieve the same effects on pages 285-286.

There are three values that the align attribute can take that control how the image should align vertically with the text that surrounds it:

## top

This aligns the first line of the surrounding text with the top of the image.

## middle

This aligns the first line of the surrounding text with the middle of the image.

## bottom

This aligns the first line of the surrounding text with the bottom of the image.

chapter-05/aligning-images-vertically.html

HTML

```
<p>There are around  
10,000 living species of birds that inhabit  
different ecosystems from the Arctic to the  
Antarctic. Many species undertake long distance  
annual migrations, and many more perform shorter  
irregular journeys.</p>  
<hr />  
<p>There are around  
10,000 living species of birds that inhabit  
different ecosystems from the Arctic to the  
Antarctic. Many species undertake long distance  
annual migrations, and many more perform shorter  
irregular journeys.</p>  
<hr />  
<p>There are around  
10,000 living species of birds that inhabit  
different ecosystems from the Arctic to the  
Antarctic. Many species undertake long distance  
annual migrations, and many more perform shorter  
irregular journeys.</p>
```

## RESULT



There are around 10,000 living species of birds that

inhabit different ecosystems from the Arctic to the Antarctic. Many species undertake long distance annual migrations, and many more perform shorter irregular journeys.



There are around 10,000 living species of birds that

inhabit different ecosystems from the Arctic to the Antarctic. Many species undertake long distance annual migrations, and many more perform shorter irregular journeys.



There are around 10,000 living species of birds that

inhabit different ecosystems from the Arctic to the Antarctic. Many species undertake long distance annual migrations, and many more perform shorter irregular journeys.

The value of `top` places the first line of text near the top of the image and subsequent lines of text appear under the image.

The value of `middle` places the first line of text near the vertical middle of the image and subsequent lines of text appear under the image.

The value of `bottom` places the first line of text near the bottom of the image and subsequent lines of text under the image.

When text flows right up to the edge of an image it can make it harder to read. You will learn how to add a gap between text and images on pages 313-314 using the CSS padding and margin properties.

If you would like all of the text to wrap around the image (rather than just one line of text), you should use the CSS `float` property discussed on pages 370-372.

In older code, you may see the `align` attribute used with the values `left` or `right` to achieve the same effect (as described on the previous page), although its use is no longer recommended.

# THREE RULES FOR CREATING IMAGES

There are three rules to remember when you are creating images for your website which are summarized below. We go into greater detail on each topic over the next nine pages.

1

## SAVE IMAGES IN THE RIGHT FORMAT

Websites mainly use images in jpeg, gif, or png format. If you choose the wrong image format then your image might not look as sharp as it should and can make the web page slower to load.

2

## SAVE IMAGES AT THE RIGHT SIZE

You should save the image at the same width and height it will appear on the website. If the image is smaller than the width or height that you have specified, the image can be distorted and stretched. If the image is larger than the width and height if you have specified, the image will take longer to display on the page.

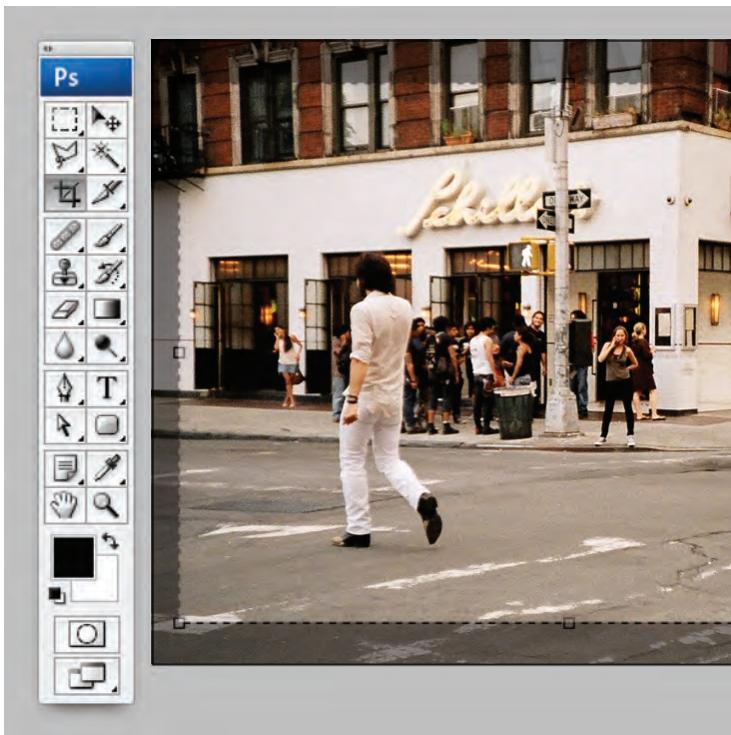
3

## USE THE CORRECT RESOLUTION

Computer screens are made up of dots known as pixels. Images used on the web are also made up of tiny dots. Resolution refers to the number of dots per inch, and most computer screens only show web pages at 72 pixels per inch. So saving images at a higher resolution results in images that are larger than necessary and take longer to download.

# TOOLS TO EDIT & SAVE IMAGES

There are several tools you can use to edit and save images to ensure that they are the right size, format, and resolution.



The most popular tool amongst web professionals is **Adobe Photoshop**. (In fact, professional web designers often use this software to design entire sites.) The full version of Photoshop is expensive, but there is a cheaper version called Photoshop Elements which would suit the needs of most beginners.

## OTHER SOFTWARE

Adobe Fireworks  
Pixelmator  
PaintShop Pro  
Paint.net

## ONLINE EDITORS

[www.photoshop.com](http://www.photoshop.com)  
[www.pixlr.com](http://www.pixlr.com)  
[www.splashup.com](http://www.splashup.com)  
[www.ipiccy.com](http://www.ipiccy.com)

## ONLINE EXTRA

Watch videos that demonstrate how to resize images and save them in the correct format using both of these applications.

# IMAGE FORMATS: JPEG

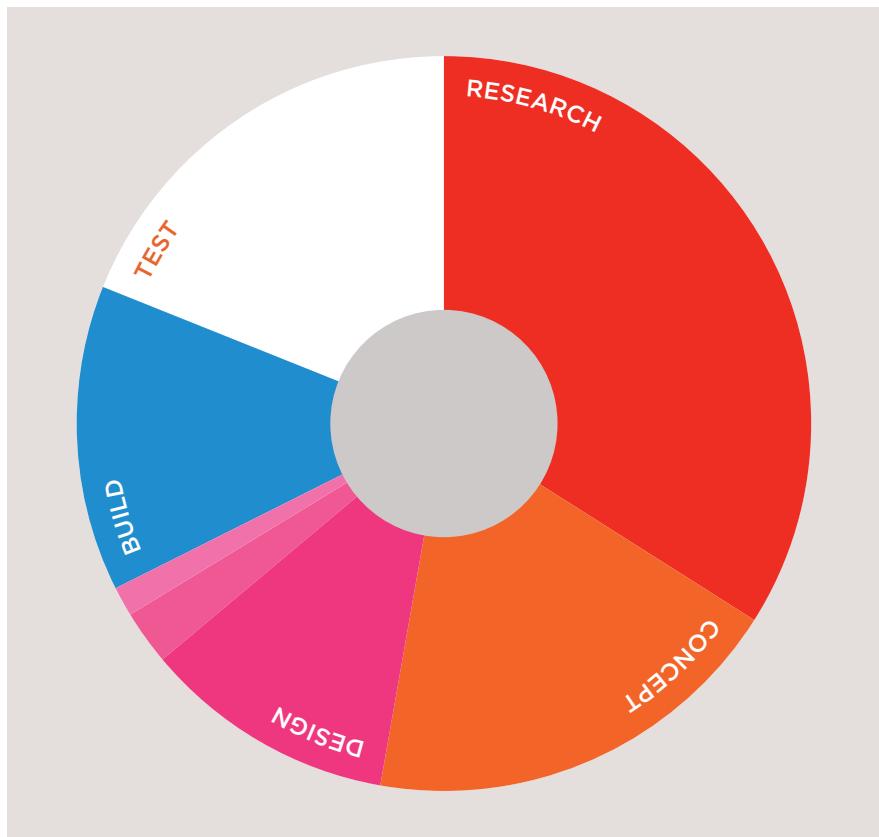
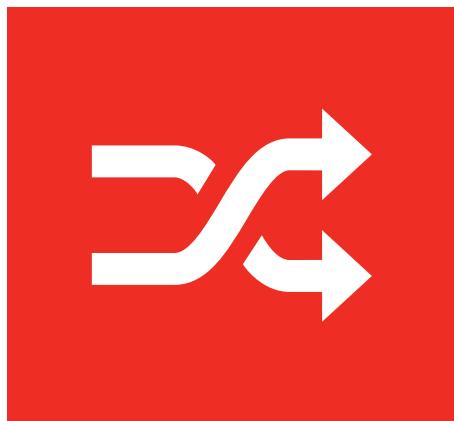


Whenever you have many different colors in a picture you should use a JPEG. A photograph that features snow or an overcast sky might look like it has large areas that are just white or gray, but the picture is usually made up of many different colors that are subtly different.

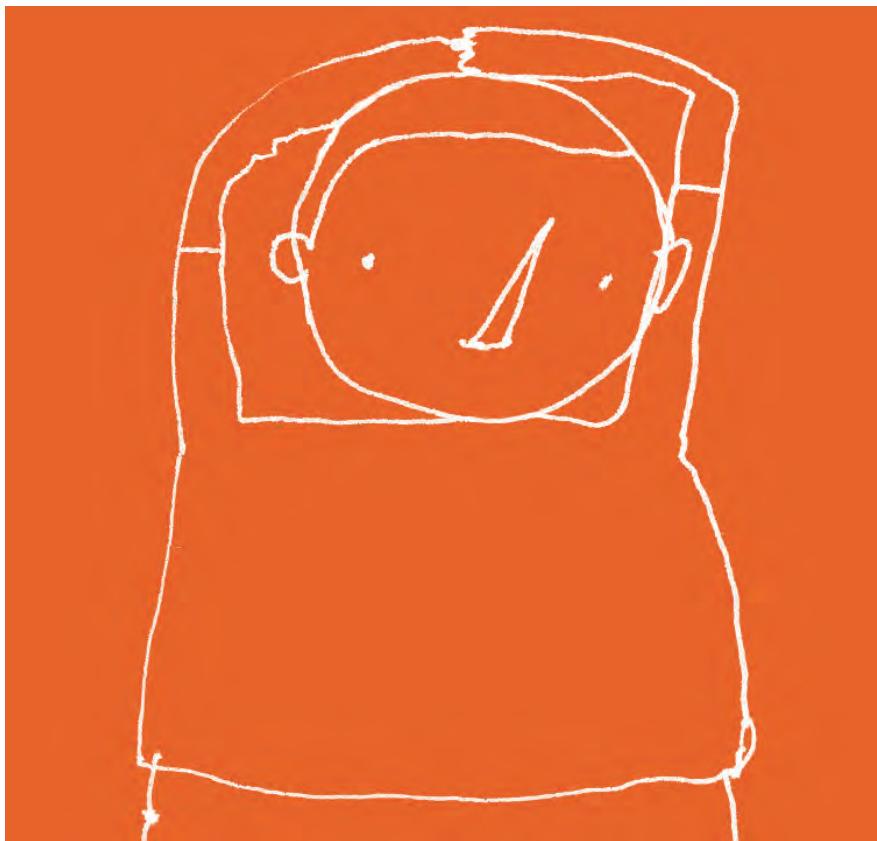




# IMAGE FORMATS: GIF



Use GIF or PNG format  
when saving images  
with few colors or large  
areas of the same color.



When a picture has an area that is filled with exactly the same color, it is known as flat color. Logos, illustrations, and diagrams often use flat colors. (Note that photographs of snow, sky, or grass are not flat colors, they are made up of many subtly different shades of the same color and are not as suited to GIF or PNG format.)

# IMAGE DIMENSIONS

The images you use on your website should be saved at the same width and height that you want them to appear on the page.

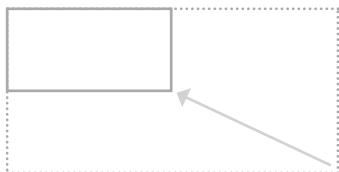
For example, if you have designed a page to include an image that is 300 pixels wide by 150 pixels tall, the image you use should be 300 x 150 pixels. You may need to use image editing tools to resize and crop the

image. When sourcing images, it is important to understand how you can alter the dimensions of an image; imagine that you had designed a web page to include an image that is 300 pixels wide by 150 pixels tall:

## ONLINE EXTRA

Visit the tools section of the website accompanying this book to watch a video guide to resizing images in Photoshop and GIMP.

**REDUCING IMAGE SIZE**  
You can reduce the size of images to create a smaller version of the image.



**Example:** If your image is 600 pixels wide and 300 pixels tall, you can reduce the size of the image by 50%.

**Result:** This will create an image that is quicker to download.

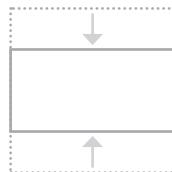
**INCREASING IMAGE SIZE**  
You can't increase the size of photos significantly without affecting the image quality.



**Example:** If your image is only 100 pixels wide by 50 pixels tall, increasing the size by 300% would result in poor quality.

**Result:** The image will look blurry or blocky.

**CHANGING SHAPE**  
Only some images can be cropped without losing valuable information (see next page).



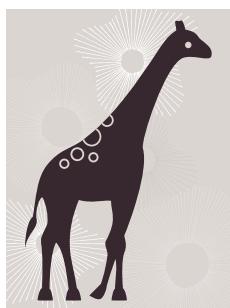
**Example:** If your image is 300 pixels square, you can remove parts of it, but in doing so you might lose valuable information.

**Result:** Only some images can be cropped and still make sense.

# CROPPING IMAGES

When cropping images it is important not to lose valuable information. It is best to source images that are the correct shape if possible.

PORTRAIT

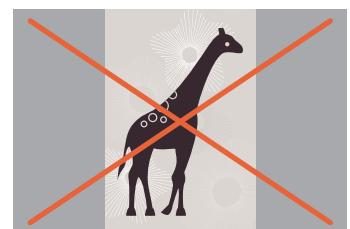


Here you can see an illustration of a giraffe that is best suited to appearing in **portrait**.

LANDSCAPE

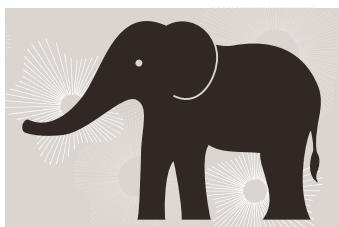


If we **crop** this illustration to make it landscape we lose the head and feet.



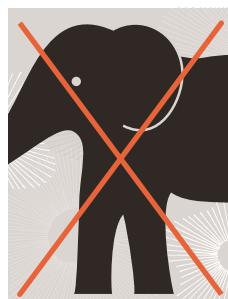
If we **add extra space** to the left and right of the illustration the background is not continued.

LANDSCAPE

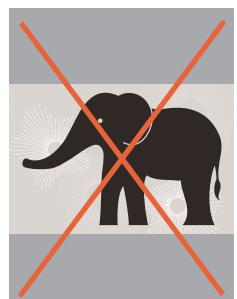


Here you can see an illustration of an elephant that is best suited to appearing in **landscape**.

PORTRAIT



If we **crop** this illustration to make it portrait we lose the trunk and the hindquarters.



If we **add extra space** to the top and bottom of the illustration the background is not continued.

# IMAGE RESOLUTION

Images created for the web should be saved at a resolution of 72 ppi. The higher the resolution of the image, the larger the size of the file.

JPGs, GIFs, and PNGs belong to a type of image format known as **bitmap**. They are made up of lots of miniature squares. The **resolution** of an image is the number of squares that fit within a 1 inch x 1 inch square area.

Images appearing on **computer** screens are made of tiny squares called **pixels**. A small segment of this photograph has been magnified to show how it is made up of pixels. The web browsers on most desktop

computers display images at a resolution of **72** pixels per inch (ppi). Images in **print** materials (such as books and magazines) are made up of tiny circles called **dots**. These images are usually printed at a resolution of **300** dots per inch (dpi).



For this image:  
JPEG at 300 dpi = 1,526kb  
JPEG at 72 ppi = 368kb

Due to the fact that computer displays are capped at a resolution of 72 ppi, using images on the web with a higher resolution will not result in better image quality — only in larger file sizes, which will increase the time needed to load them and therefore slow down viewing of your web pages.

# VECTOR IMAGES

Vector images differ from bitmap images and are resolution-independent. Vector images are commonly created in programs such as Adobe Illustrator.

When an image is a line drawing (such as a logo, illustration, or diagram), designers will often create it in vector format. Vector formatted images are very different to bitmap images.

Vector images are created by placing points on a grid, and drawing lines between those points. A color can then be added to "fill in" the lines that have been created.

The advantage of creating line drawings in vector format is that you can increase the dimensions of the image without affecting the quality of it.

The current method of using vector images for display on websites involves saving a bitmap version of the original vector image and using that.

Scalable Vector Graphics (SVG) are a relatively new format used to display vector images directly on the web (eliminating the need to create bitmap versions of them), however its use is not yet widespread.



# ANIMATED GIFS

Animated GIFs show several frames of an image in sequence and therefore can be used to create simple animations.

Below you can see the individual frames that make up an animated GIF that shows an orange dot revolving around a circle — like the kind of animation you might see when a web page is loading.

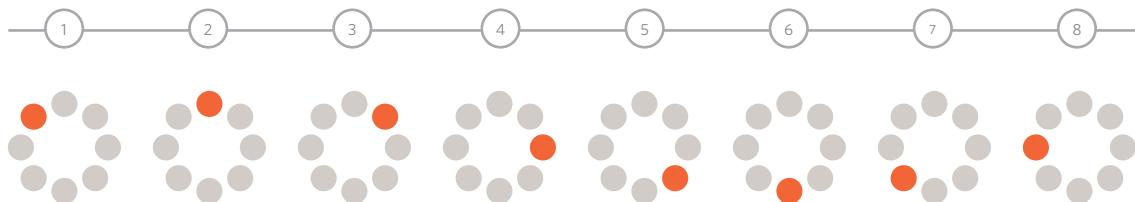
Some image editing applications such as Adobe Photoshop allow you to create animated GIFs. There are several tutorials about how to do this on the web. There are also several websites that allow you to upload the graphics for the individual frames and create the animated GIF for you.

## IT IS IMPORTANT TO REMEMBER:

Each extra frame of the image increases the size of the file, and can therefore add to the time it takes for an image to download (and web users do not like waiting a long time for images to download).

Because GIFs are not an ideal format for displaying photographs, animated GIFs are really only suitable for simple illustrations.

Some designers frown on animated GIFs because they remember a lot of amateur web designers overusing them in the 1990's.



# TRANSPARENCY

Creating an image that is partially transparent (or "see-through") for the web involves selecting one of two formats:

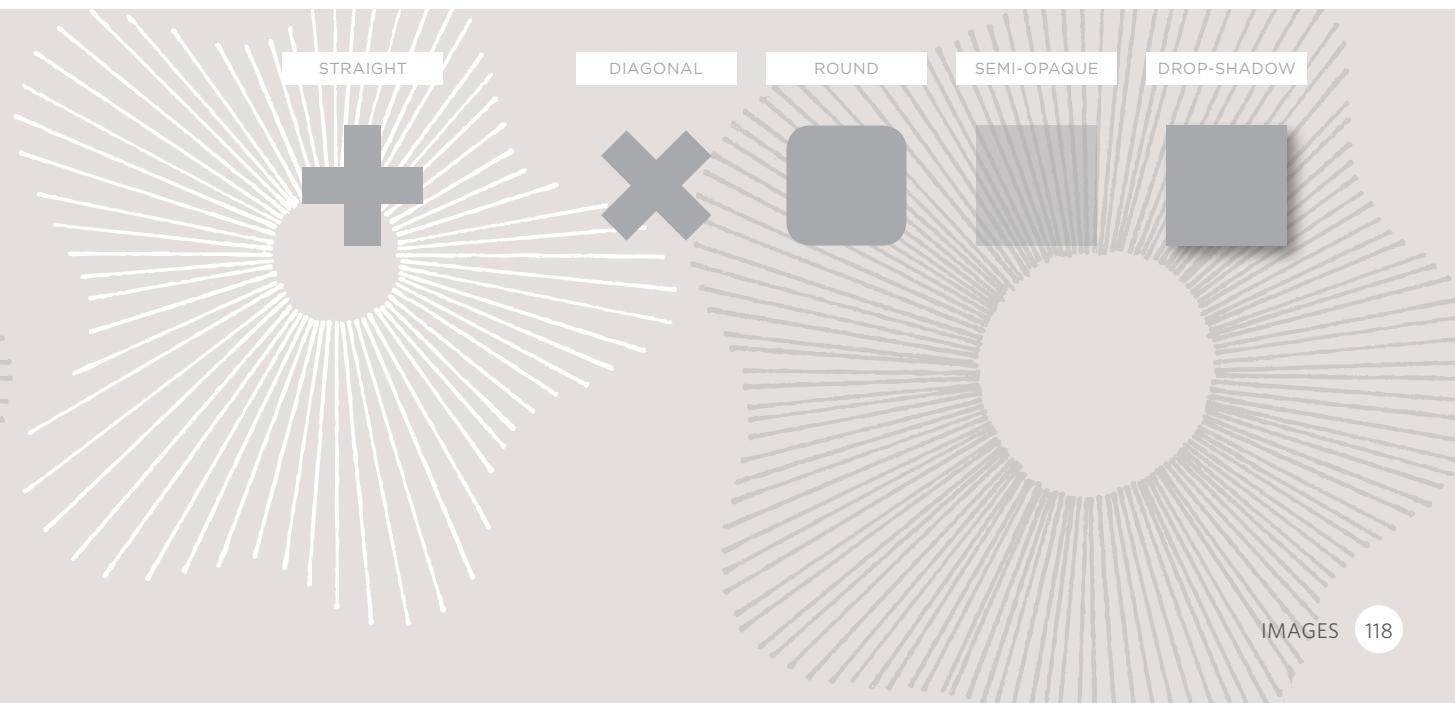
## TRANSPARENT GIF

If the transparent part of the image has straight edges and it is 100% transparent (that is, not semi-opaque), you can save the image as a GIF (with the transparency option selected).

## PNG

If the transparent part of the image has diagonal or rounded edges or if you want a semi-opaque transparency or a drop-shadow, then you will need to save it as a PNG.

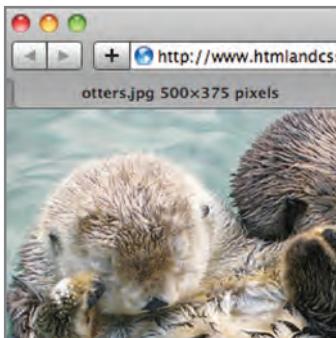
Transparent PNGs are not fully supported in older browsers, most notably Internet Explorer 6 (IE6). There is some JavaScript you can use to get around this issue. The details of this script can be found in the tools section of the website accompanying this book.



# EXAMINING IMAGES ON THE WEB

## CHECKING THE SIZE OF IMAGES

If you are updating a website, you might need to check the size of an existing image before creating a new one to replace it. This can be achieved by right-clicking on the image and making a selection from the pop-up menu that appears. (Mac users will need to hold down the control key and click rather than right-click.)



On the left you can see how to check the size of images and how to download them using Safari. Below is a brief overview of what to select in the pop-up menu to perform these functions in various browsers.

### CHROME

Size: **Open Image in New Tab**

Size appears in new tab

Download: **Save Image As**

### FIREFOX

Size: **View Image Info**

Size appears in pop-up window

Download: **Save Image As**

### INTERNET EXPLORER

Size: **Properties**

Size appears in pop-up window

Download: **Save Image**

### SAFARI

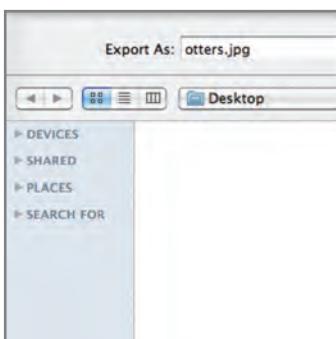
Size: **Open Image in New Tab**

Size appears in title bar

Download: **Save Image As**

## DOWNLOADING IMAGES

If you want to download images from a website, you can do so by accessing the same pop-up menu. (Please remember however that all images online are subject to copyright and require explicit permission to reuse.)



# HTML5: FIGURE AND FIGURE CAPTION

## HTML

chapter-05/figure-and-figure-caption.html

```
<figure>
  
  <br />
  <figcaption>Sea otters hold hands when they
      sleep so they don't drift away from each
      other.</figcaption>
</figure>
```

## RESULT



Sea otters hold hands when they sleep so they don't drift away from each other.

## <figure>

Images often come with captions. HTML5 has introduced a new `<figure>` element to contain images and their caption so that the two are associated.

You can have more than one image inside the `<figure>` element as long as they all share the same caption.

## <figcaption>

The `<figcaption>` element has been added to HTML5 in order to allow web page authors to add a caption to an image.

Before these elements were created there was no way to associate an `<img>` element with its caption.

Older browsers that do not understand HTML5 elements simply ignore the new elements and display the content of them.

In this example, the logo is a GIF because it uses flat colors, while the photographs are JPEGs. The main photo is placed inside the HTML5 `<figure>` element and has its own caption.

The `alt` attribute on each image provides a description for those using screen readers and the `title` attribute provides additional information. (This is shown in the tooltip.)

This example does not use the `height`, `width`, or `align` attributes as these are being phased out and you are encouraged to use CSS properties instead.



# EXAMPLE IMAGES



```
<html>
  <head>
    <title>Images</title>
  </head>
  <body>
    <h1>
      
    </h1>
    <figure>
      
      <p>
        <figcaption>
          This recipe for individual chocolate
          cakes is so simple and so delectable!
        </figcaption>
      </p>
    </figure>
    <h4>More Recipes:</h4>
    <p>
      
      
      
    </p>
  </body>
</html>
```



# SUMMARY

## IMAGES

- ▶ The <img> element is used to add images to a web page.
- ▶ You must always specify a src attribute to indicate the source of an image and an alt attribute to describe the content of an image.
- ▶ You should save images at the size you will be using them on the web page and in the appropriate format.
- ▶ Photographs are best saved as JPEGs; illustrations or logos that use flat colors are better saved as GIFs.



# 6

## TABLES

- ▶ How to create tables
- ▶ What information suits tables
- ▶ How to represent complex data in tables

There are several types of information that need to be displayed in a grid or table. For example: sports results, stock reports, train timetables.

When representing information in a table, you need to think in terms of a grid made up of rows and columns (a bit like a spreadsheet). In this chapter you will learn how to:

- Use the four key elements for creating tables
- Represent complex data using tables
- Add captions to tables



*Shares & Markets* Business

Adult fares	Pay as you go	
	Peak*	Off-Peak*
Zone 1 only	£2.00	£1.50
Zones 1-2	£2.20	£1.70
Zones 1-3	£2.80	£2.00
Zones 1-4	£3.30	£2.30
Zones 1-5	£4.20	£2.60
Zones 1-6	£5.20	£3.20
Zones 2, 3, 4, 5 or 6	£1.50	£1.30
Zones 2-3, 3-4, 4-5, or 5-6	£1.90	£1.50
Zones 2-4, 3-5 or 4-6	£2.30	£1.70
Zones 2-5 or 3-6	£2.90	£2.00
Zones 2-6	£3.40	£2.20

**\*Peak** Oyster single fares apply from 0630 to 0930 and from 1600 to 1900 Monday to Friday (excluding public holidays).

**\*Off-Peak** Oyster single fare applies at all other times.

Keep our claims helpline number safe in  
case you need it, 0800 980 8750.

Ask about the 5% discount offered on premiums for additional pets

## Benefits

	Premier	Up to	Standard
Veterinary fees for illness and injury	£7,000 a year	£4,000 a year	
- Behavioural	£250	Not Covered	
Prescription food	£200	Not Covered	
complementary treatments	£1,000	Not Covered	
each illness/injury	No Limit	£1,000	
ing for missing pet	£1,250	£800	
ring pet	£750	£500	
ding/pet minder	£1,500	£300	
to age 9)	£700pw	£600pw	
	£750 total	£600pw total	
	£1,500	£900	
Yes	Yes	Yes	
£2,000,000		£1,000,000	
£3,000		Not Covered	
£2,000		Not Covered	



...are in sterling except where stated and reflect the closing mid price. The yield is taken months' declared net dividend as at month end, less tax. The price, except for Irish stocks, which are calculated on a 20% gross basis.

The earnings per share ratio is calculated as a percentage of earnings per share divided by the current share price. The earnings per share ratio excludes extraordinary items, net assets (NAAV) replace the P/E ratio.

**TRADE** ■ Ex-dividend. Shareholders of the issue will not receive dividends for the period from the ex-dividend date up to and including the record date.

For a listing of a stock on the FSE see the listing or a stock on this page should be listed in the FSE 100 Shares at the Independent.

— the Kalmar's  
can be found at

— 3 —

10

三

100

卷之三

卷之三

卷之三

# WHAT'S A TABLE?

A table represents information in a grid format. Examples of tables include financial reports, TV schedules, and sports results.

Grids allow us to understand complex data by referencing information on two axes.

Each block in the grid is referred to as a **table cell**. In HTML a table is written out row by row.

The screenshot shows the Reuters homepage with a focus on the 'Commodities' section. At the top, there's a navigation bar with links for 'News & Markets', 'Sectors & Industries', and 'Analysis & Opinion'. Below the main content, there's a large grid table for the 'THOMSON REUTERS/JEFFERIES CRB INDEX(TR/J CRB)'. The table includes columns for 'Change' (-3.36), 'Open' (360.92), 'High' (361.19), 'Low' (357.99), and 'Times' (04/18 14:58). To the right of this is a sidebar titled 'MARKETS' with sections for 'U.S.', 'EUROPE', 'ASIA', and 'SECTORS', and a 'Trade forex with Citi.' button. Further down, there's a 'Market Indices' section with a search bar and a table of indices like DOW, S&P 500, NASDAQ, and TR US INDEX. At the bottom, there's a 'Currencies' section and a 'Commodities' section with a 'GOLD' entry.

THOMSON REUTERS/JEFFERIES CRB INDEX(TR/J CRB)					
Change	Open	High	Low	Times	
-3.36	360.92	361.19	357.99	04/18 14:58	

MARKETS	
U.S.	EUROPE
ASIA	
SECTORS	
Sponsored by <a href="#">Trade forex with Citi.</a>	

Market Indices	
Enter company name or Symbol	<input type="button" value="SEARCH"/>
DOW	12,189.14
S&P 500	1,305.75
NASDAQ	2,731.64
TR US INDEX	119.44

Currencies	
EUR/USD	1.4234
GBP/USD	1.6262
USD/JPY	82.600

Commodities	
GOLD	1,496.20

# BASIC TABLE STRUCTURE

## <table>

The <table> element is used to create a table. The contents of the table are written out row by row.

## <tr>

You indicate the start of each row using the opening <tr> tag. (The tr stands for table row.)

It is followed by one or more <td> elements (one for each cell in that row).

At the end of the row you use a closing </tr> tag.

## <td>

Each cell of a table is represented using a <td> element. (The td stands for table data.)

At the end of each cell you use a closing </td> tag.

Some browsers automatically draw lines around the table and/or the individual cells. You will learn how to control the borders of tables using CSS on pages 309-312 and 337-340.

chapter-06/basic-table-structure.html

HTML

```
<table>
  <tr>
    <td>15</td>
    <td>15</td>
    <td>30</td>
  </tr>
  <tr>
    <td>45</td>
    <td>60</td>
    <td>45</td>
  </tr>
  <tr>
    <td>60</td>
    <td>90</td>
    <td>90</td>
  </tr>
</table>
```

RESULT

15	15	30
45	60	45
60	90	90

# TABLE HEADINGS

## HTML

chapter-06/table-headings.html

```
<table>
  <tr>
    <th></th>
    <th scope="col">Saturday</th>
    <th scope="col">Sunday</th>
  </tr>
  <tr>
    <th scope="row">Tickets sold:</th>
    <td>120</td>
    <td>135</td>
  </tr>
  <tr>
    <th scope="row">Total sales:</th>
    <td>$600</td>
    <td>$675</td>
  </tr>
</table>
```

## RESULT

Saturday	Sunday
<b>Tickets sold:</b> 120	135
<b>Total sales:</b> \$600	\$675

## <th>

The `<th>` element is used just like the `<td>` element but its purpose is to represent the heading for either a column or a row. (The `th` stands for table heading.)

Even if a cell has no content, you should still use a `<td>` or `<th>` element to represent the presence of an empty cell otherwise the table will not render correctly. (The first cell in the first row of this example shows an empty cell.)

Using `<th>` elements for headings helps people who use screen readers, improves the ability for search engines to index your pages, and also enables you to control the appearance of tables better when you start to use CSS.

You can use the `scope` attribute on the `<th>` element to indicate whether it is a heading for a column or a row. It can take the values: `row` to indicate a heading for a row or `col` to indicate a heading for a column.

Browsers usually display the content of a `<th>` element in bold and in the middle of the cell.

# SPANNING COLUMNS

Sometimes you may need the entries in a table to stretch across more than one column.

The `colspan` attribute can be used on a `<th>` or `<td>` element and indicates how many columns that cell should run across.

In the example on the right you can see a timetable with five columns; the first column contains the heading for that row (the day), the remaining four represent one hour time slots.

If you look at the table cell that contains the words 'Geography' you will see that the value of the `colspan` attribute is 2, which indicates that the cell should run across two columns. In the third row, 'Gym' runs across three columns.

You can see that the second and third rows have fewer `<td>` elements than there are columns. This is because, when a cell extends across more than one column, the `<td>` or `<th>` cells that would have been in the place of the wider cells are not included in the code.

I added some CSS styles to this example so that you can see how the cells span more than one column. You will learn how to do this on pages 250, 337-340.

chapter-06/spanning-columns.html

HTML

```
<table>
  <tr>
    <th></th>
    <th>9am</th>
    <th>10am</th>
    <th>11am</th>
    <th>12am</th>
  </tr>
  <tr>
    <th>Monday</th>
    <td colspan="2">Geography</td>
    <td>Math</td>
    <td>Art</td>
  </tr>
  <tr>
    <th>Tuesday</th>
    <td colspan="3">Gym</td>
    <td>Home Ec</td>
  </tr>
</table>
```

RESULT

	9am	10am	11am	12am
Monday	Geography		Math	Art
Tuesday	Gym		Home Ec	

# SPANNING ROWS

## HTML

chapter-06/spanning-rows.html

```
<table>
<tr>
  <th></th>
  <th>ABC</th>
  <th>BBC</th>
  <th>CNN</th>
</tr>
<tr>
  <th>6pm - 7pm</th>
  <td rowspan="2">Movie</td>
  <td>Comedy</td>
  <td>News</td>
</tr>
<tr>
  <th>7pm - 8pm</th>
  <td>Sport</td>
  <td>Current Affairs</td>
</tr>
</table>
```

## RESULT

	ABC	BBC	CNN
6pm - 7pm	Movie	Comedy	News
7pm - 8pm		Sport	Current Affairs

You may also need entries in a table to stretch down across more than one row.

The `rowspan` attribute can be used on a `<th>` or `<td>` element to indicate how many rows a cell should span down the table.

In the example on the left you can see that ABC is showing a movie from 6pm - 8pm, whereas the BBC and CNN channels are both showing two programs during this time period (each of which lasts one hour).

If you look at the last `<tr>` element, it only contains three elements even though there are four columns in the result below. This is because the movie in the `<tr>` element above it uses the `rowspan` attribute to stretch down and take over the cell below.

I have added some CSS styles to this example so that you can see how the cells span more than one row. You will learn how to apply these CSS styles to tables on pages 250, 337-340.

# LONG TABLES

There are three elements that help distinguish between the main content of the table and the first and last rows (which can contain different content).

These elements help people who use screen readers and also allow you to style these sections in a different manner than the rest of the table (as you will see when you learn about CSS).

## <thead>

The headings of the table should sit inside the <thead> element.

## <tbody>

The body should sit inside the <tbody> element.

## <tfoot>

The footer belongs inside the <tfoot> element.

By default, browsers rarely treat the content of these elements any differently than other elements however designers often use CSS styles to change their appearance.

chapter-06/long-tables.html

HTML

```
<table>
  <thead>
    <tr>
      <th>Date</th>
      <th>Income</th>
      <th>Expenditure</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>1st January</th>
      <td>250</td>
      <td>36</td>
    </tr>
    <tr>
      <th>2nd January</th>
      <td>285</td>
      <td>48</td>
    </tr>
    <!-- additional rows as above -->
    <tr>
      <th>31st January</th>
      <td>129</td>
      <td>64</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td></td>
      <td>7824</td>
      <td>1241</td>
    </tr>
  </tfoot>
</table>
```

## RESULT

Date	Income	Expenditure
1st January	250	36
2nd January	285	48
3rd January	260	42
4th January	290	38
5th January	310	115
6th January	168	14
7th January	226	20
8th January	253	37
9th January	294	33
10th January	216	46
11th January	244	29
12th January	297	32
13th January	328	86
14th January	215	38
15th January	254	30
16th January	256	27
17th January	311	68
18th January	212	39
19th January	234	36
20th January	221	43
21st January	259	38
22nd January	246	31
23rd January	248	17
24th January	229	45
25th January	263	34
26th January	258	41
27th January	283	22
28th January	256	30
29th January	278	47
30th January	251	15
31st January	129	64
	7824	1241

Some of the HTML editors that come in content management systems offer tools to help draw tables. If the first row of your table only contains `<th>` elements then you may find that the editor inserts a `<thead>` element automatically.

Part of the reason for having separate `<thead>` and `<tfoot>` elements is so that, if you have a table that is taller than the screen (or, if printed, longer than one page) then the browser can keep the header and footer visible whilst the contents of the table scroll. This is intended to make it easier for users to see which column the data is in (however this functionality is not implemented by default in any current browser).

I have added some CSS styles to this example so that you can see the contents of the `<thead>` and `<tfoot>` being treated differently than the rest of the rows. You will learn how to apply these CSS styles to tables on pages 309-312 and 337-340.

# OLD CODE: WIDTH & SPACING

There are some outdated attributes which you should not use on new websites. You may, however, come across some of them when looking at older code, so I will mention them here. All of these attributes have been replaced by the use of CSS.

The `width` attribute was used on the opening `<table>` tag to indicate how wide that table should be and on some opening `<th>` and `<td>` tags to specify the width of individual cells. The value of this attribute is the width of the table or cell in pixels.

The columns in a table need to form a straight line, so you often only see the `width` attribute on the first row (and all subsequent rows would use that setting).

The opening `<table>` tag could also use the `cellpadding` attribute to add space inside each cell of the table, and the `cellspacing` attribute to create space between each cell of the table. The values for these attributes were given in pixels.

I added CSS styles to this example so that you can see the width of the table cells more clearly. If you want to control the width or spacing of tables and cells you should use CSS as shown on pages 303, 337-340.

chapter-06/width-and-spacing.html

HTML

```
<table width="400" cellpadding="10" cellspacing="5">
<tr>
  <th width="150"></th>
  <th>Withdrawn</th>
  <th>Credit</th>
  <th width="150">Balance</th>
</tr>
<tr>
  <th>January</th>
  <td>250.00</td>
  <td>660.50</td>
  <td>410.50</td>
</tr>
<tr>
  <th>February</th>
  <td>135.55</td>
  <td>895.20</td>
  <td>1170.15</td>
</tr>
</table>
```

RESULT

	Withdrawn	Credit	Balance
January	250.00	660.50	410.50
February	135.55	895.20	1170.15

# OLD CODE: BORDER & BACKGROUND

## HTML

chapter-06/border-and-background.html

```
<table border="2" bgcolor="#efefef">
<tr>
  <th width="150"></th>
  <th>Withdrawn</th>
  <th>Credit</th>
  <th width="150" bgcolor="#cccccc">Balance</th>
</tr>
<tr>
  <th>January</th>
  <td>250.00</td>
  <td>660.50</td>
  <td bgcolor="#cccccc">410.50</td>
</tr>
<tr>
  <th>February</th>
  <td>135.55</td>
  <td>895.20</td>
  <td bgcolor="#cccccc">1170.15</td>
</tr>
</table>
```

## RESULT

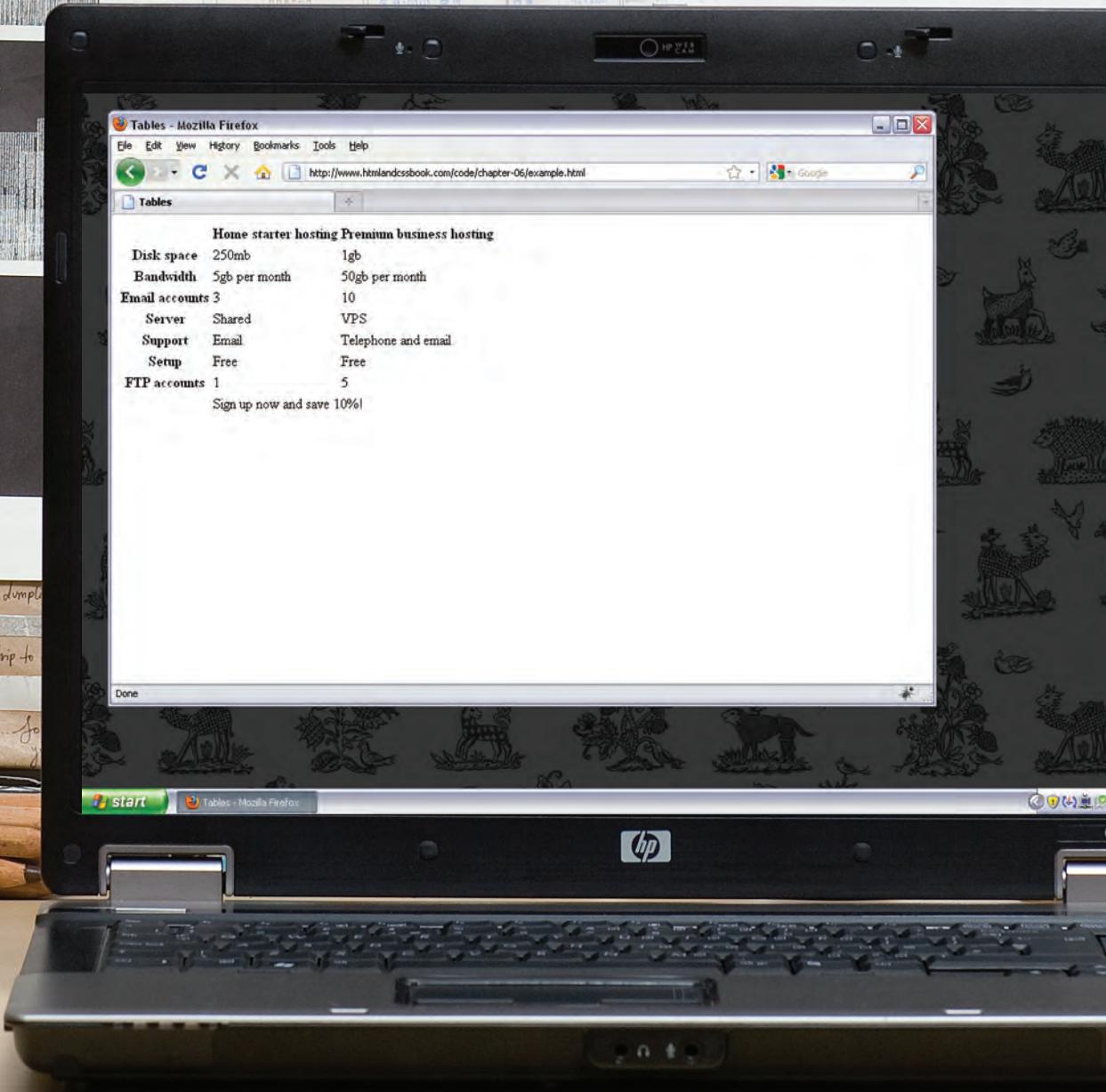
	Withdrawn	Credit	Balance
January	250.00	660.50	410.50
February	135.55	895.20	1170.15

The border attribute was used on both the `<table>` and `<td>` elements to indicate the width of the border in pixels.

The `bgcolor` attribute was used to indicate background colors of either the entire table or individual table cells. The value is usually a hex code (which we discuss on pages 249-252).

This example uses the HTML border and `bgcolor` attributes. No CSS attributes were utilized in this example.

When building a new website you should use CSS to control the appearance of the table rather than these attributes. They are only covered here because you may come across them if you look at the code of older websites.



This example shows a table for customers to compare website hosting packages. There are table headings in the first row and first column of the table.

The empty cell in the top left still has a `<th>` element to represent it. Each cell of the table must be accounted for by a `<th>` or `<td>` element. The `<th>` elements use

the `scope` attribute to indicate whether they are headings for a row or column. The final row uses the `colspan` attribute to spread across all three columns.

# EXAMPLE TABLES



```
<html>
  <head>
    <title>Tables</title>
  </head>
  <body>
    <table>
      <thead>
        <tr>
          <th></th>
          <th scope="col">Home starter hosting</th>
          <th scope="col">Premium business hosting</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <th scope="row">Disk space</th>
          <td>250mb</td>
          <td>1gb</td>
        </tr>
        <tr>
          <th scope="row">Bandwidth</th>
          <td>5gb per month</td>
          <td>50gb per month</td>
        </tr>
        <!-- more rows like the two above here -->
      </tbody>
      <tfoot>
        <tr>
          <td></td>
          <td colspan="2">Sign up now and save 10%!</td>
        </tr>
      </tfoot>
    </table>
  </body>
</html>
```



# SUMMARY TABLES

- ▶ The <table> element is used to add tables to a web page.
- ▶ A table is drawn out row by row. Each row is created with the <tr> element.
- ▶ Inside each row there are a number of cells represented by the <td> element (or <th> if it is a header).
- ▶ You can make cells of a table span more than one row or column using the rowspan and colspan attributes.
- ▶ For long tables you can split the table into a <thead>, <tbody>, and <tfoot>.



# 7

## FORMS

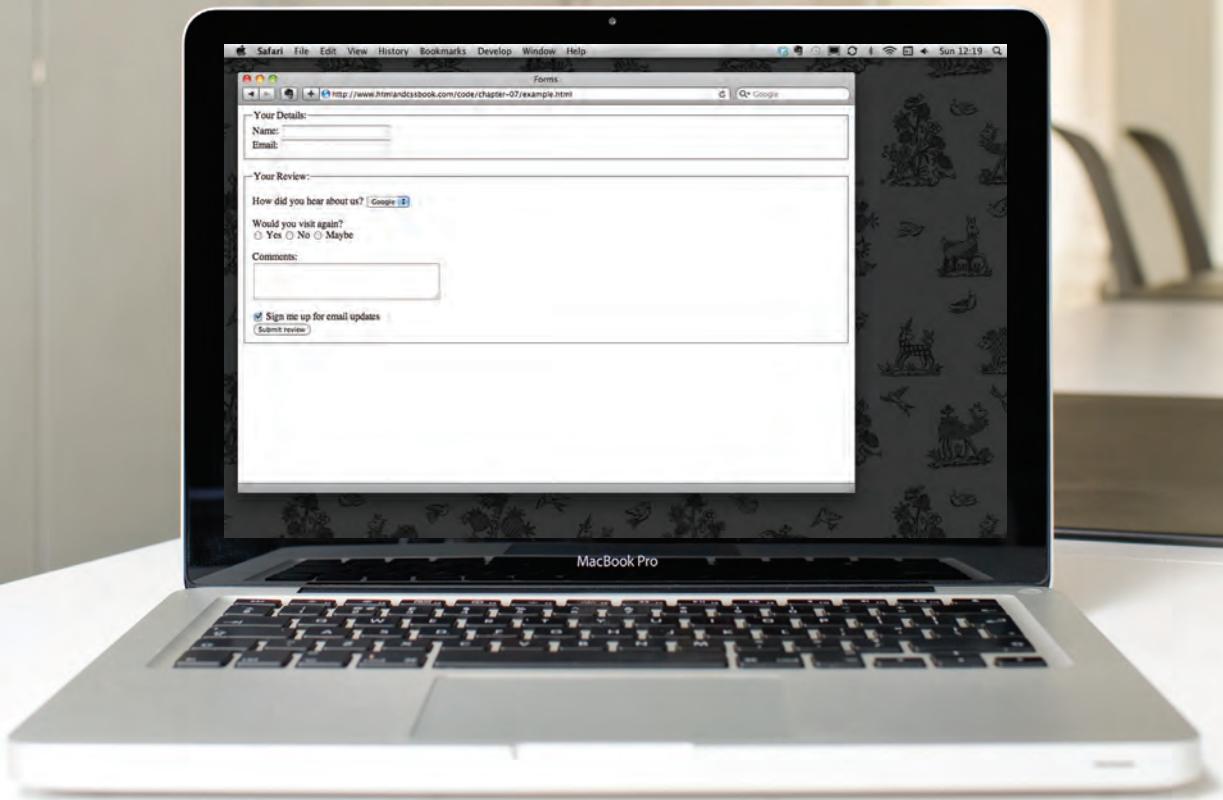
- ▶ How to collect information from visitors
- ▶ Different kinds of form controls
- ▶ New HTML5 form controls

Traditionally, the term 'form' has referred to a printed document that contains spaces for you to fill in information.

HTML borrows the concept of a form to refer to different elements that allow you to collect information from visitors to your site.

Whether you are adding a simple search box to your website or you need to create more complicated insurance applications, HTML forms give you a set of elements to collect data from your users. In this chapter you will learn:

- How to create a form on your website
- The different tools for collecting data
- New HTML5 form controls

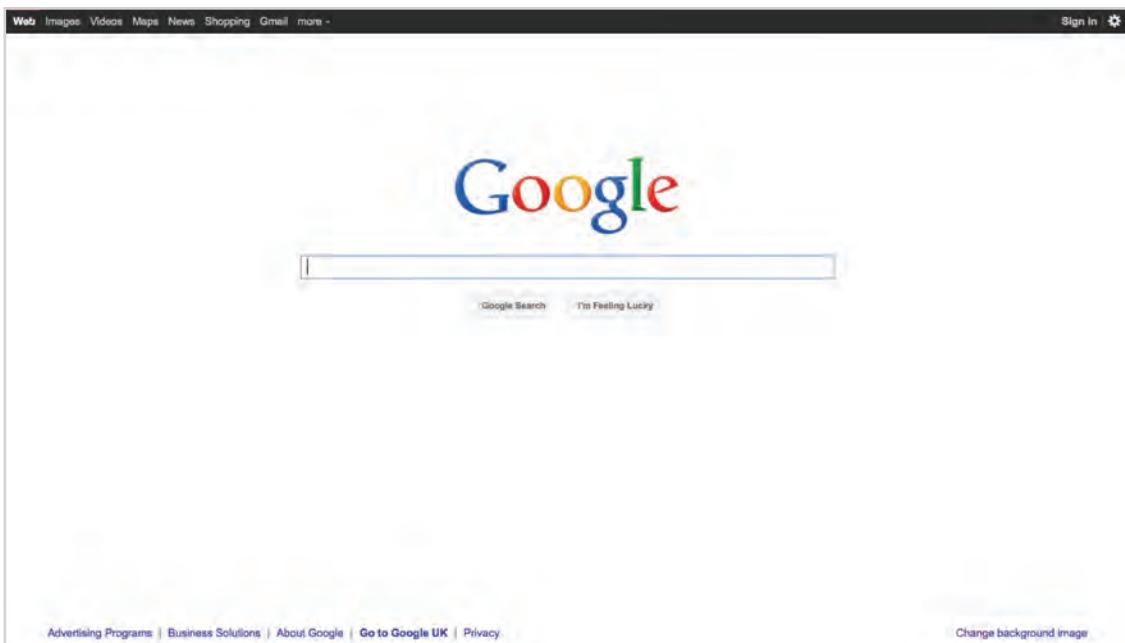


# WHY FORMS?

The best known form on the web is probably the search box that sits right in the middle of Google's homepage.

In addition to enabling users to search, forms also allow users to perform other functions online. You will see forms

when registering as a member of a website, when shopping online, and when signing up for newsletters or mailing lists.



# FORM CONTROLS

There are several types of form controls that you can use to collect information from visitors to your site.

## ADDING TEXT:

### Text input (single-line)

Used for a single line of text such as email addresses and names.

### Password input

Like a single line text box but it masks the characters entered.

### Text area (multi-line)

For longer areas of text, such as messages and comments.

## MAKING CHOICES:

### Radio buttons

For use when a user must select one of a number of options.

 Rock  Pop  Jazz

### Checkboxes

When a user can select and unselect one or more options.

 iTunes  Last.fm  Spotify

### Drop-down boxes

When a user must pick one of a number of options from a list.

## SUBMITTING FORMS:

### Submit buttons

To submit data from your form to another web page.

### Image buttons

Similar to submit buttons but they allow you to use an image.

## UPLOADING FILES:

### File upload

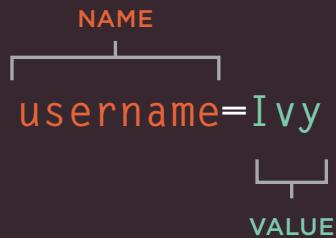
Allows users to upload files (e.g. images) to a website.

# HOW FORMS WORK

A user fills in a form and then presses a button to submit the information to the server.



A form may have several form controls, each gathering different information. The server needs to know which piece of inputted data corresponds with which form element.



To differentiate between various pieces of inputted data, information is sent from the browser to the server using name/value pairs. In this example, the form asks for the visitor's username and also for their favorite jazz musician. The name/value pairs sent to the server are:

`username=Ivy`

If the form control allows the user to enter text, then the value of the form control is whatever the user has typed in.

`vote=Herbie`

If the form control allows you to choose from a fixed set of answers (e.g. radio buttons, checkboxes or a drop down list), the web page author will add code that gives each option an automatic value.

You should never change the name of a form control in a page unless you know that the code on the server will understand this new value.

# FORM STRUCTURE

## <form>

Form controls live inside a <form> element. This element should always carry the action attribute and will usually have a method and id attribute too.

### action

Every <form> element requires an action attribute. Its value is the URL for the page on the server that will receive the information in the form when it is submitted.

### method

Forms can be sent using one of two methods: get or post.

With the get method, the values from the form are added to the end of the URL specified in the action attribute. The get method is ideal for:

- short forms (such as search boxes)
- when you are just retrieving data from the web server (not sending information that should be added to or deleted from a database)

chapter-07/form-structure.html

HTML

```
<form action="http://www.example.com/subscribe.php"
      method="get">
  <p>This is where the form controls will appear.
  </p>
</form>
```

RESULT

This is where the form controls will appear.

With the post method the values are sent in what are known as HTTP headers. As a rule of thumb you should use the post method if your form:

- allows users to upload a file
- is very long
- contains sensitive data (e.g. passwords)
- adds information to, or deletes information from, a database

If the method attribute is not used, the form data will be sent using the get method.

### id

We look at the id attribute on page 183, but the value is used to identify the form distinctly from other elements on the page (and is often used by scripts — such as those that check you have entered information into fields that require values).

# TEXT INPUT

## HTML

chapter-07/text-input.html

```
<form action="http://www.example.com/login.php">
<p>Username:
<input type="text" name="username" size="15"
       maxlength="30" />
</p>
</form>
```

## RESULT

Username:

### size

The `size` attribute should not be used on new forms. It was used in older forms to indicate the width of the text input (measured by the number of characters that would be seen).

For example, a value of 3 would create a box wide enough to display three characters

(although a user could enter more characters if they desired).

In any new forms you write, CSS should be used to control the width of form elements. The `size` attribute is only mentioned here because you may come across it when looking at older code.

## <input>

The `<input>` element is used to create several different form controls. The value of the `type` attribute determines what kind of input they will be creating.

### type="text"

When the `type` attribute has a value of `text`, it creates a single-line text input.

### name

When users enter information into a form, the server needs to know which form control each piece of data was entered into. (For example, in a login form, the server needs to know what has been entered as the username and what has been given as the password.) Therefore, each form control requires a `name` attribute. The value of this attribute identifies the form control and is sent along with the information they enter to the server.

### maxlength

You can use the `maxlength` attribute to limit the number of characters a user may enter into the text field. Its value is the number of characters they may enter. For example, if you were asking for a year, the `maxlength` attribute could have a value of 4.

# PASSWORD INPUT

## <input>

### **type="password"**

When the type attribute has a value of password it creates a text box that acts just like a single-line text input, except the characters are blocked out. They are hidden in this way so that if someone is looking over the user's shoulder, they cannot see sensitive data such as passwords.

### **name**

The name attribute indicates the name of the password input, which is sent to the server with the password the user enters.

### **size, maxlength**

It can also carry the size and maxlength attributes like the single-line text input.

chapter-07/password-input.html

HTML

```
<form action="http://www.example.com/login.php">
  <p>Username:
    <input type="text" name="username" size="15"
           maxlength="30" />
  </p>
  <p>Password:
    <input type="password" name="password" size="15"
           maxlength="30" />
  </p>
</form>
```

RESULT

Username:

Password:

Although the password is hidden on the screen, this does not mean that the data in a password control is sent securely to the server. You should never use these for sending sensitive data such as credit card numbers.

For full security, the server needs to be set up to communicate with users' browsers using Secure Sockets Layer (SSL). The topic of SSL is beyond the scope of this book, however there are links to learn more about it on the accompanying website.

# TEXT AREA

## HTML

chapter-07/textarea.html

```
<form action="http://www.example.com/comments.php">
  <p>What did you think of this gig?</p>
  <textarea name="comments" cols="20" rows="4">Enter
    your comments...</textarea>
</form>
```

## RESULT

What did you think of this gig?

If you are creating a new form, you should use CSS to control the width and height of a `<textarea>`. However, if you are looking at older code, you may see the `cols` and `rows` attributes used with this element.

The `cols` attribute indicates how wide the text area should be (measured in numbers of characters). The `rows` attribute indicates how many rows the text area should take up vertically.

## <textarea>

The `<textarea>` element is used to create a multi-line text input. Unlike other input elements this is not an empty element. It should therefore have an opening and a closing tag.

Any text that appears between the opening `<textarea>` and closing `</textarea>` tags will appear in the text box when the page loads.

If the user does not delete any text between these tags, this message will get sent to the server along with whatever the user has typed. (Some sites use JavaScript to clear this information when the user clicks in the text area.)

# RADIO BUTTON

## <input>

### type="radio"

Radio buttons allow users to pick just one of a number of options.

#### name

The name attribute is sent to the server with the value of the option the user selects. When a question provides users with options for answers in the form of radio buttons, the value of the name attribute should be the same for all of the radio buttons used to answer that question.

#### value

The value attribute indicates the value that is sent to the server for the selected option. The value of each of the buttons in a group should be different (so that the server knows which option the user has selected).

#### checked

The checked attribute can be used to indicate which value (if any) should be selected when the page loads. The value of this attribute is checked. Only one radio button in a group should use this attribute.

chapter-07/radio-button.html

HTML

```
<form action="http://www.example.com/profile.php">
<p>Please select your favorite genre:<br />
<input type="radio" name="genre" value="rock" checked="checked" /> Rock
<input type="radio" name="genre" value="pop" /> Pop
<input type="radio" name="genre" value="jazz" /> Jazz
</p>
</form>
```

RESULT

Please select your favorite genre:

Rock  Pop  Jazz

**Please note:** Once a radio button has been selected it cannot be deselected. The user can only select a different option. If you are only allowing the user one

option and want them to be able to deselect it (for example if they are indicating they agree to terms and conditions), you should use a checkbox instead.

# CHECKBOX

## HTML

chapter-07/checkbox.html

```
<form action="http://www.example.com/profile.php">
<p>Please select your favorite music service(s):
<br />
<input type="checkbox" name="service"
      value="itunes" checked="checked" /> iTunes
<input type="checkbox" name="service"
      value="lastfm" /> Last.fm
<input type="checkbox" name="service"
      value="spotify" /> Spotify
</p>
</form>
```

## RESULT

Please select your favorite music service(s):  
 iTunes  Last.fm  Spotify

## <input>

### type="checkbox"

Checkboxes allow users to select (and unselect) one or more options in answer to a question.

### name

The name attribute is sent to the server with the value of the option(s) the user selects. When a question provides users with options for answers in the form of checkboxes, the value of the name attribute should be the same for all of the buttons that answer that question.

### value

The value attribute indicates the value sent to the server if this checkbox is checked.

### checked

The checked attribute indicates that this box should be checked when the page loads. If used, its value should be checked.

# DROP DOWN LIST BOX

## <select>

A drop down list box (also known as a select box) allows users to select one option from a drop down list.

The <select> element is used to create a drop down list box. It contains two or more <option> elements.

### name

The name attribute indicates the name of the form control being sent to the server, along with the value the user selected.

## <option>

The <option> element is used to specify the options that the user can select from. The words between the opening <option> and closing </option> tags will be shown to the user in the drop down box.

### value

The <option> element uses the value attribute to indicate the value that is sent to the server along with the name of the control if this option is selected.

chapter-07/drop-down-list-box.html

HTML

```
<form action="http://www.example.com/profile.php">
<p>What device do you listen to music on?</p>
<select name="devices">
<option value="ipod">iPod</option>
<option value="radio">Radio</option>
<option value="computer">Computer</option>
</select>
</form>
```

RESULT

What device do you listen to music on?



### selected

The selected attribute can be used to indicate the option that should be selected when the page loads. The value of this attribute should be selected.

If this attribute is not used, the first option will be shown when the page loads. If the user does not select an option, then the first item will be sent to the server as the value for this control.

The function of the drop down list box is similar to that of the radio buttons (in that only one option can be selected). There are two key factors in choosing which to use:

1. If users need to see all options at a glance, radio buttons are better suited.
2. If there is a very long list of options (such as a list of countries), drop down list boxes work better.

# MULTIPLE SELECT BOX

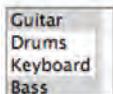
## HTML

chapter-07/multiple-select-box.html

```
<form action="http://www.example.com/profile.php">
<p>Do you play any of the following instruments?
(You can select more than one option by holding
down control on a PC or command key on a Mac
while selecting different options.)</p>
<select name="instruments" size="3"
multiple="multiple">
<option value="guitar" selected="selected">
    Guitar</option>
<option value="drums">Drums</option>
<option value="keyboard"
selected="selected">Keyboard</option>
<option value="bass">Bass</option>
</select>
</form>
```

## RESULT

Do you play any of the following instruments? (You can select more than one option by holding down control on a PC or command key on a Mac while selecting different options.)



## <select>

### size

You can turn a drop down select box into a box that shows more than one option by adding the `size` attribute. Its value should be the number of options you want to show at once. In the example you can see that three of the four options are shown.

Unfortunately, the way that browsers have implemented this attribute is not perfect, and it should be tested thoroughly if used (in particular in Firefox and Safari on a Mac).

### multiple

You can allow users to select multiple options from this list by adding the `multiple` attribute with a value of `multiple`.

It is a good idea to tell users if they can select more than one option at a time. It is also helpful to indicate that on a PC they should hold down the **control** key while selecting multiple options and on a Mac they should use the **command** key while selecting options.

# FILE INPUT BOX

## <input>

If you want to allow users to upload a file (for example an image, video, mp3, or a PDF), you will need to use a file input box.

### **type="file"**

This type of input creates a box that looks like a text input followed by a **browse** button. When the user clicks on the **browse** button, a window opens up that allows them to select a file from their computer to be uploaded to the website.

When you are allowing users to upload files, the **method** attribute on the <form> element must have a value of **post**. (You cannot send files using the HTTP **get** method.)

When a user clicks on the **browse** button, the presentation of the window that allows them to browse for the file they want to upload will match the windows of the user's operating system. You cannot control the appearance of these windows.

chapter-07/file-input-box.html

HTML

```
<form action="http://www.example.com/upload.php"
      method="post">
  <p>Upload your song in MP3 format:</p>
  <input type="file" name="user-song" /><br />
  <input type="submit" value="Upload" />
</form>
```

RESULT



# SUBMIT BUTTON

## HTML

chapter-07/submit-button.html

```
<form action="http://www.example.com/subscribe.php">
  <p>Subscribe to our email list:</p>
  <input type="text" name="email" />
  <input type="submit" name="subscribe"
        value="Subscribe" />
</form>
```

## RESULT

Subscribe to our email list:

## <input>

### type="submit"

The submit button is used to send a form to the server.

### name

It can use a name attribute but it does not need to have one.

### value

The value attribute is used to control the text that appears on a button. It is a good idea to specify the words you want to appear on a button because the default value of buttons on some browsers is 'Submit query' and this might not be appropriate for all kinds of form.

Different browsers will show submit buttons in different ways and tend to fit the visual presentation of the browser. If you want to control the appearance of a submit button, you can either use CSS (as you will learn on page 343), or you can use an image for the button.

# IMAGE BUTTON

## <input>

### type="image"

If you want to use an image for the submit button, you can give the type attribute a value of image. The src, width, height, and alt attributes work just like they do when used with the <img> element (which we saw on pages 99-100).

chapter-07/image-button.html

HTML

```
<form action="http://www.example.org/subscribe.php">
  <p>Subscribe to our email list:</p>
  <input type="text" name="email" />
  <input type="image" src="images/subscribe.jpg"
        width="100" height="20" />
</form>
```

RESULT

Subscribe to our email list:

# BUTTON & HIDDEN CONTROLS

## HTML

chapter-07/button-and-hidden-controls.html

```
<form action="http://www.example.com/add.php">
  <button> Add</button>
  <input type="hidden" name="bookmark"
    value="lyrics" />
</form>
```

## RESULT



## <button>

The `<button>` element was introduced to allow users more control over how their buttons appear, and to allow other elements to appear inside the button.

This means that you can combine text and images between the opening `<button>` tag and closing `</button>` tag.

## <input>

### `type="hidden"`

This example also shows a hidden form control. These form controls are not shown on the page (although you can see them if you use the **View Source** option in the browser). They allow web page authors to add values to forms that users cannot see. For example, a web page author might use a hidden field to indicate which page the user was on when they submitted a form.

# LABELLING FORM CONTROLS

## <label>

When introducing form controls, the code was kept simple by indicating the purpose of each one in text next to it. However, each form control should have its own <label> element as this makes the form accessible to vision-impaired users.

The <label> element can be used in two ways. It can:

1. Wrap around both the text description and the form input (as shown on the first line of the example to your right).
2. Be kept separate from the form control and use the `for` attribute to indicate which form control it is a label for (as shown with the radio buttons).

### for

The `for` attribute states which form control the label belongs to. Note how the radio buttons use the `id` attribute. The value of the `id` attribute uniquely identifies an element from all other elements on a page. (The `id` attribute is covered on page 183.)

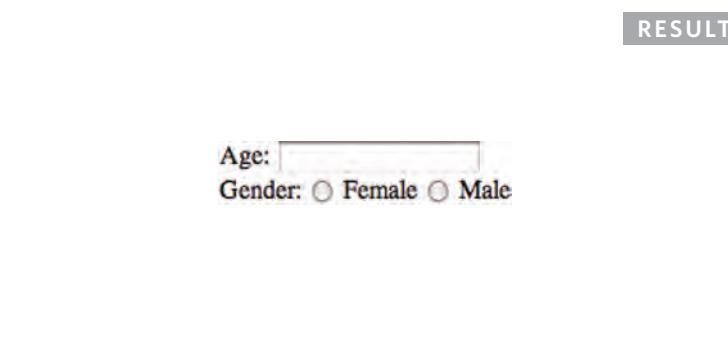
The value of the `for` attribute matches that of the `id` attribute on the form control it is labelling. This technique using the `for` and

chapter-07/labelling-form-controls.html

HTML

```
<label>Age: <input type="text" name="age" /></label>
<br/ >
Gender:
<input id="female" type="radio" name="gender"
       value="f">
<label for="female">Female</label>
<input id="male" type="radio" name="gender"
       value="m">
<label for="male">Male</label>
```

RESULT



Age:

Gender:  Female  Male

`id` attributes can be used on any form control. When a <label> element is used with a checkbox or radio button, users can click on either the form control or the label to select. The expanded clickable area makes the form easier to use. The position of the label is very important. If users do not know where to enter information or what information to enter, they are less likely to use the form correctly.

As a rule of thumb, here are the best places to place labels on form controls.

#### ABOVE OR TO THE LEFT:

- Text inputs
- Text areas
- Select boxes
- File uploads

#### TO THE RIGHT:

- Individual checkboxes
- Individual radio buttons

# GROUPING FORM ELEMENTS

## HTML

chapter-07/grouping-form-elements.html

```
<fieldset>
  <legend>Contact details</legend>
  <label>Email:<br />
  <input type="text" name="email" /></label><br />
  <label>Mobile:<br />
  <input type="text" name="mobile" /></label><br />
  <label>Telephone:<br />
  <input type="text" name="telephone" /></label>
</fieldset>
```

## RESULT

Contact details

Email:

Mobile:

Telephone:

## <fieldset>

You can group related form controls together inside the `<fieldset>` element. This is particularly helpful for longer forms.

Most browsers will show the `fieldset` with a line around the edge to show how they are related. The appearance of these lines can be adjusted using CSS.

## <legend>

The `<legend>` element can come directly after the opening `<fieldset>` tag and contains a caption which helps identify the purpose of that group of form controls.

# HTML5: FORM VALIDATION

You have probably seen forms on the web that give users messages if the form control has not been filled in correctly; this is known as **form validation**.

Traditionally, form validation has been performed using JavaScript (which is beyond the scope of this book). But HTML5 is introducing validation and leaving the work to the browser.

Validation helps ensure the user enters information in a form that the server will be able to understand when the form is submitted. Validating the contents of the form before it is sent to the server helps:

- Reduce the amount of work the server has to do
- Enables users to see if there are problems with the form faster than if validation were performed on the server.

chapter-07/html5-form-validation.html

HTML

```
<form action="http://www.example.com/login/"  
      method="post">  
  <label for="username">Username:</label>  
  <input type="text" name="username"  
        required="required" /><br />  
  <label for="password">Password:</label>  
  <input type="password" name="password"  
        required="required" />  
  <input type="submit" value="Submit" />  
</form>
```

RESULT

At the time of writing, only Chrome and Opera supported HTML5 validation, although other browsers are expected to follow. In order to support older browsers (that do not understand HTML5), web page authors are likely to continue using JavaScript to validate forms.

An example of HTML5 form validation is the `required` attribute, which can be used on any form element that the user is expected to fill in. This HTML5 attribute does not need a value, but in HTML 4 all attributes must have a value. So, some people give this attribute a value of `required`.

# HTML5: DATE INPUT

## HTML

chapter-07/html5-date-input.html

```
<form action="http://www.example.com/bookings/"  
      method="post">  
  <label for="username">Departure date:</label>  
  <input type="date" name="depart" />  
  <input type="submit" value="Submit" />  
</form>
```

## RESULT

Departure date:

## <input>

Many forms need to gather information such as dates, email addresses, and URLs. This has traditionally been done using text inputs.

HTML5 introduces new form controls to standardize the way that some information is gathered. Older browsers that do not recognize these inputs will just treat them as a single line text box.

## type="date"

If you are asking the user for a date, you can use an <input> element and give the type attribute a value of date. This will create a date input in browsers that support the new HTML5 input types.

This example shows what the date input looks like in the Opera browser. The appearance of the date input changes across different browsers.

# HTML5: EMAIL & URL INPUT

## <input>

HTML5 has also introduced inputs that allow visitors to enter email addresses and URLs. Browsers that do not support these input types will just treat them as text boxes.

### type="email"

If you ask a user for an email address, you can use the email input. Browsers that support HTML5 validation will check that the user has provided information in the correct format of an email address. Some smart phones also optimize their keyboard to display the keys you are most likely to need when entering an email address (such as the @ symbol).

### type="url"

A URL input can be used when you are asking a user for a web page address. Browsers that support HTML5 validation will check that the user has provided information in the format of a URL. Some smart phones also optimize their keyboard to display the keys you are most likely to need when entering a URL.

chapter-07/html5-email-input.html

HTML

```
<form action="http://www.example.org/subscribe.php">
  <p>Please enter your email address:</p>
  <input type="email" name="email" />
  <input type="submit" value="Submit" />
</form>
```

Please enter your email address:

RESULT

ivy  Please enter an email address.

chapter-07/html5-url-input.html

HTML

```
<form action="http://www.example.org/profile.php">
  <p>Please enter your website address:</p>
  <input type="url" name="website" />
  <input type="submit" value="Submit" />
</form>
```

Please enter your website address:

RESULT

ivy  Please enter a URL.

# HTML5: SEARCH INPUT

## HTML

chapter-07/html5-search-input.html

```
<form action="http://www.example.org/search.php">
  <p>Search:</p>
  <input type="search" name="search" />
  <input type="submit" value="Search" />
</form>
```

## RESULT

Search:

## HTML

chapter-07/html5-placeholder.html

```
<form action="http://www.example.org/search.php">
  <p>Search:</p>
  <input type="search" name="search"
    placeholder="Enter keyword" />
  <input type="submit" value="Search" />
</form>
```

## RESULT

Search:

## <input>

If you want to create a single line text box for search queries, HTML5 provides a special type of input for that purpose.

### **type="search"**

If you want to create a single line text box for search queries, HTML5 provides a special search input.

To create the HTML5 search box the `<input>` element should have a `type` attribute whose value is `search`. Older browsers will simply treat it like a single line text box.

Recent browsers add some features that improve usability. For example, Safari on a Mac adds a cross to clear the search box when you have started to enter information. Safari also automatically rounds the corners on the search input field.

### **placeholder**

On any text input, you can also use an attribute called `placeholder` whose value is text that will be shown in the text box until the user clicks in that area. Older browsers simply ignore this attribute.

Safari File Edit View History Bookmarks Develop Window Help

Forms

http://www.htmlandcssbook.com/code/chapter-07/example.html

Your Details:

Name:

Email:

Your Review:

How did you hear about us?  Google

Would you visit again?

Yes  No  Maybe

Comments:

Sign me up for email updates

MacBook Pro

# EXAMPLE FORMS



This example shows a feedback and newsletter sign-up form. It uses a variety of form controls.

The `<form>` element uses the `action` attribute to indicate the page that the data is being sent to. Each of the form controls sits inside the `<form>` element. Different types of form control are suited to collecting different types of data. The `<fieldset>` element is used to group related questions together. The `<label>` element indicates the purpose of each form control.

# EXAMPLE FORMS

```
<html>
  <head>
    <title>Forms</title>
  </head>
  <body>
    <form action="http://www.example.com/review.php" method="get">
      <fieldset>
        <legend>
          Your Details:
        </legend>
        <label>
          Name:
          <input type="text" name="name" size="30" maxlength="100">
        </label>
        <br />
        <label>
          Email:
          <input type="email" name="email" size="30" maxlength="100">
        </label>
        <br />
      </fieldset>
      <br />
      <fieldset>
        <legend>
          Your Review:
        </legend>
        <p>
          <label for="hear-about">
            How did you hear about us?
          </label>
          <select name="referrer" id="hear-about">
            <option value="google">Google</option>
            <option value="friend">Friend</option>
            <option value="advert">Advert</option>
            <option value="other">Other</option>
          </select>
        </p>
        <p>
```

# EXAMPLE FORMS

```
Would you visit again?  
<br />  
<label>  
  <input type="radio" name="rating" value="yes" />  
  Yes  
</label>  
<label>  
  <input type="radio" name="rating" value="no" />  
  No  
</label>  
<label>  
  <input type="radio" name="rating" value="maybe" />  
  Maybe  
</label>  
</p>  
<p>  
  <label for="comments">  
    Comments:  
</label>  
  <br />  
  <textarea rows="4" cols="40" id="comments">  
  </textarea>  
</p>  
<label>  
  <input type="checkbox" name="subscribe" checked="checked" />  
  Sign me up for email updates  
</label>  
<br />  
  <input type="submit" value="Submit review" />  
</fieldset>  
</form>  
</body>  
</html>
```



# SUMMARY FORMS

- ▶ Whenever you want to collect information from visitors you will need a form, which lives inside a `<form>` element.
- ▶ Information from a form is sent in name/value pairs.
- ▶ Each form control is given a name, and the text the user types in or the values of the options they select are sent to the server.
- ▶ HTML5 introduces new form elements which make it easier for visitors to fill in forms.



# 8

## EXTRA MARKUP

- ▶ Specifying different versions of HTML
- ▶ Identifying and grouping elements
- ▶ Comments, meta information and iframes

At this point, we have covered the main tags that fit nicely into groups and sections.

In this chapter, we will focus on some helpful topics that are not easily grouped together. You will learn about:

- The different versions of HTML and how to indicate which version you are using
- How to add comments to your code
- Global attributes, which are attributes that can be used on any element, including the `class` and `id` attributes
- Elements that are used to group together parts of the page where no other element is suitable
- How to embed a page within a page using iframes
- How to add information about the web page using the `<meta>` element
- Adding characters such as angled brackets and copyright symbols



# 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, `<font>` 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 `<img />`).
- 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 `<font>`).

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>, <ul>, and <li>.

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 `<em>Origins of Art</em>` cycle is organized around four themes: `<b>science, architecture, history</b>` and `<b>religion</b>`.

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>`, `<b>`, `<em>`, and `<img>`.

## 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.

## <span>

The `<span>` 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 `<span>` 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 `<span>` elements:

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

# IFRAMES

## <iframe>

An iframe is like a little window that has been cut into your page — and in that window you can see another page. The term iframe is an abbreviation of inline frame.

One common use of iframes (that you may have seen on various websites) is to embed a Google Map into a page. The content of the iframe can be any html page (either located on the same server or anywhere else on the web).

An iframe is created using the <iframe> element. There are a few attributes that you will need to know to use it:

### src

The src attribute specifies the URL of the page to show in the frame.

### height

The height attribute specifies the height of the iframe in pixels.

### width

The width attribute specifies the width of the iframe in pixels.

chapter-08/iframe.html

HTML

```
<iframe  
width="450"  
height="350"  
src="http://maps.google.co.uk/maps?q=moma+new+york  
&output=embed">  
</iframe>
```

RESULT



## HTML

chapter-08/iframes-continued.html

```
<iframe  
src="http://maps.google.co.uk/maps?q=moma+new+york  
&output=embed"  
width="450"  
height="350"  
frameborder="0"  
scrolling="no">  
</iframe>
```

## RESULT



## scrolling

The scrolling attribute will not be supported in HTML5. In HTML 4 and XHTML, it indicates whether the iframe should have scrollbars or not. This is important if the page inside the iframe is larger than the space you have allowed for it (using the height and width attributes). Scrollbars allow the user to move around the frame to see more content. It can take one of three values: yes (to show scrollbars), no (to hide scrollbars) and auto (to show them only if needed).

## frameborder

The frameborder attribute will not be supported in HTML5. In HTML 4 and XHTML, it indicates whether the frame should have a border or not. A value of 0 indicates that no border should be shown. A value of 1 indicates that a border should be shown.

## seamless

In HTML5, a new attribute called seamless can be applied to an iframe where scrollbars are not desired. The seamless attribute (like some other new HTML5 attributes) does not need a value, but you will often see authors give it a value of seamless. Older browsers do not support the seamless attribute.

# INFORMATION ABOUT YOUR PAGES

## <meta>

The <meta> element lives inside the <head> element and contains information about that web page.

It is not visible to users but fulfills a number of purposes such as telling search engines about your page, who created it, and whether or not it is time sensitive. (If the page is time sensitive, it can be set to expire.)

The <meta> element is an empty element so it does not have a closing tag. It uses attributes to carry the information.

The most common attributes are the name and content attributes, which tend to be used together. These attributes specify properties of the entire page. The value of the name attribute is the property you are setting, and the value of the content attribute is the value that you want to give to this property.

In the first line of the example on the opposite page, you can see a <meta> element where the name attribute indicates an intention to specify a description for the page. The content attribute is where this description is actually specified.

The value of the name attribute can be anything you want it to be. Some defined values for this attribute that are commonly used are:

### description

This contains a description of the page. This description is commonly used by search engines to understand what the page is about and should be a maximum of 155 characters. Sometimes it is also displayed in search engine results.

### keywords

This contains a list of comma-separated words that a user might search on to find the page. In practice, this no longer has any noticeable effect on how search engines index your site.

### robots

This indicates whether search engines should add this page to their search results or not. A value of `noindex` can be used if this page should not be added. A value of `nofollow` can be used if search engines should add this page in their results but not any pages that it links to.

## HTML

chapter-08/meta.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Information About Your Pages</title>
    <meta name="description"
          content="An Essay on Installation Art" />
    <meta name="keywords"
          content="installation, art, opinion" />
    <meta name="robots"
          content="nofollow" />
    <meta http-equiv="author"
          content="Jon Duckett" />
    <meta http-equiv="pragma"
          content="no-cache" />
    <meta http-equiv="expires"
          content="Fri, 04 Apr 2014 23:59:59 GMT" />
  </head>
  <body>
  </body>
</html>
```

The `<meta>` element also uses the `http-equiv` and `content` attributes in pairs. In our example, you can see three instances of the `http-equiv` attribute. Each one has a different purpose:

### author

This defines the author of the web page.

### pragma

This prevents the browser from caching the page. (That is, storing it locally to save time downloading it on subsequent visits.)

### expires

Because browsers often cache the content of a page, the `expires` option can be used to indicate when the page should expire (and no longer be cached). Note that the date must be specified in the format shown.

# 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 &lt; or &#60;. For an ampersand, you can use either &amp; or &#38;.

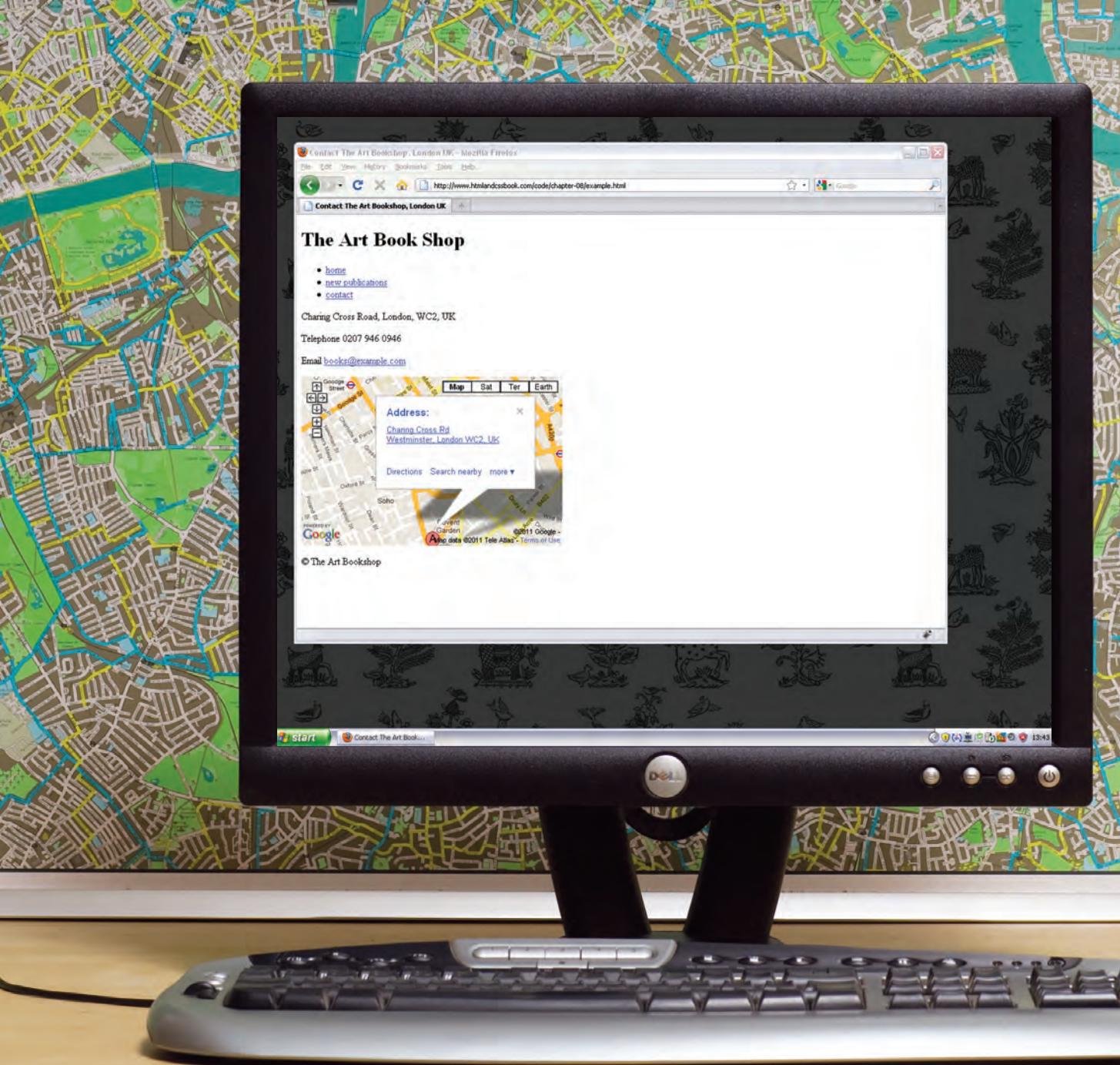
There are also special codes that can be used to show symbols such as copyright and trademark, currency symbols, mathematical characters, and some punctuation marks. For example, if you want to include a copyright symbol on a web page you can use either &copy; or &#169;.

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.

<	<b>Less-than sign</b> &lt; amp;#60;	¢	<b>Cent sign</b> &cent; amp;#162;	'	<b>Left single quote</b> &lsquo; amp;#8216;
>	<b>Greater-than sign</b> &gt; amp;#62;	£	<b>Pound sign</b> &pound; amp;#163;	'	<b>Right single quote</b> &rsquo; amp;#8217;
&	<b>Ampersand</b> &amp; amp;#38;	¥	<b>Yen sign</b> &yen; amp;#165;	"	<b>Left double quotes</b> &ldquo; amp;#8220;
"	<b>Quotation mark</b> &quot; amp;#34;	€	<b>Euro sign</b> &euro; amp;#8364;	"	<b>Right double quotes</b> &rdquo; amp;#8221;
(C)	<b>Copyright symbol</b> &copy; amp;#169;	X	<b>Multiplication sign</b> &times; amp;#215;		
(R)	<b>Registered trademark</b> &reg; amp;#174;	÷	<b>Division sign</b> &divide; amp;#247;		
TM	<b>Trademark</b> &trade; amp;#8482;				



This example starts by using a DOCTYPE to indicate that this is an HTML 4 web page. In the head, you can also see a <meta> tag describing the page's

content. Several elements use the id and class attributes to identify their purpose. The copyright symbol has been added using an escape code.

Parts of the page have been grouped using <div> elements, and comments have been added to indicate what the </div> elements are closing.

# EXAMPLE

## EXTRA MARKUP



```

<!DOCTYPE html PUBLIC
  "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta name="description" content="Telephone, email
      and directions for The Art Bookshop, London, UK" />
    <title>Contact The Art Bookshop, London UK</title>
  </head>
  <body>
    <div id="header">
      <h1>The Art Book Shop</h1>
      <ul>
        <li><a href="index.html">home</a></li>
        <li><a href="index.html">new publications</a>
          </li>
        <li class="current-page">
          <a href="index.html">contact</a></li>
        </ul>
    </div><!-- end header -->
    <div id="content">
      <p>Charing Cross Road, London, WC2, UK</p>
      <p><span class="contact">Telephone</span>
        0207 946 0946</p>
      <p><span class="contact">Email</span>
        <a href="mailto:books@example.com">books@example.com</a></p>
      <iframe width="425" height="275" frameborder="0"
        scrolling="no" marginheight="0" marginwidth="0"
        src="http://maps.google.co.uk/maps?f=q&
        source=s_q&hl=en&geocode=&
        q=charing+cross+road+london&output=embed">
      </iframe>
    </div><!-- end content -->
    <p>&copy; The Art Bookshop</p>
  </body>
</html>

```



# 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 `<span>` 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 `©`.

