Contents

1 The web design challenge 9

The goal of this book 10
The diversity of devices 11
How devices affect design 12
Introducing key technologies 14
Degrading gracefully 16
What about mobile? 17
Understanding accessibility 18
Top accessibility principles 19
DIY or outsource? 21
How to set up your website 22

2 Planning your website 23

The purpose of your website 24
How will you compete? 26
Understanding your visitors 27
Creating a sitemap 28
Planning interaction 30
Hosting your website 31
Why domain names matter 32
8 domain name buying tips 33
Domain name pitfalls 34
Working with web designers 35

3 Creating effective website content 37

Ideas for content 38
How web writing differs 40
8 top web writing tips 41
Images that work 42
Finding photos on Flickr 44
Creating a cartoon 45
Compressing images 46
Adding a map to your site 48

4 **Layout and design** 49

The role of your design 50
Fixed or flexible? 51
Using the grid 53
Tips for good alignment 54
Thinking above the fold 55
Organizing information 56
Creating a color scheme 57
Using gradients 59
Choosing fonts 60
What is your look and feel? 61
The right look and feel 62

5 **Designing effective navigation** 63

What is navigation? 64
Laying out your navigation 65
Grouping the options 67
Dropdown menus 68
Using icons 70
Tabbed navigation 71
The role of the homepage 72
You are here... 74
Encouraging exploration 76
Adding a search engine 78
Effective error messages 80
14 tips for effective links 81

6 **HTML: The language of the web** 83

What is HTML? 84
Structuring HTML pages 85
Adding pictures 86
Adding links 87
Creating tables 88
More advanced tables 90
What is a web form? 91

Choosing form elements 92
Using the input tag 94
Using other form elements 96
Creating lists 97
The art of good HTML 98
Dividing the page up 100
Your next steps with HTML 102
Reference: structure 103
Reference: header 104
Reference: text and forms 105
Tables, lists, links, images 106

7 CSS: Giving your pages some style 107

Why use CSS? 108
Transforming your HTML 109
How CSS describes design 110
Adding CSS to your site 111
How to change colors 112
Styling your text 114
Padding, border and margin 116
Styling with class 118
Advanced CSS selectors 119
Creating a simple layout 120
Styling the content 123
Printer-friendly pages 124
List bullets and numbers 125
Creating a navbar with CSS 126
Advanced positioning 128
CSS3 techniques 130
Next steps with CSS 132
Reference: text formatting 133
Reference: backgrounds 134
Reference: layout 135
Reference: pseudo-classes 137
Reference: CSS3 138

8 JavaScript for interactive pages 139

What is JavaScript? 140
How to add JavaScript 142
Creating functions 143
Responding to user actions 144
Showing and hiding content 146
How it works 147
Adding a toggle routine 148
Simple form validation 149
Checking text boxes 150
More advanced form tests 151
Customizing by date 152
Opening new windows 154
Adding random content 155
Adding a photo slideshow 156
Saving time with jQuery 157
2 quick jQuery examples 158

9 Audio, video and Flash 159

What is Flash? 160
Using Flash Professional 161
Adding video using Flash 162
Adding audio using Flash 164
Creating a Flash animation 165
Top tips for Flash animation 168

10 Tools for website design 169

Using Microsoft Word 170
HTML and visual editors 171
Introducing Dreamweaver 173
A Dreamweaver tour 174

11 Adding a shopping cart 177

Essentials for ecommerce 178
What is PayPal? 179
Choosing the right cart 180
Creating your cart buttons 181
Adding your cart buttons 184
Adding a View Cart button 185
Testing your shopping cart 186

12 Adding a social dimension 187

Why social matters 188
Integrating with Facebook 189
Integrating with Twitter 190
Adding a "Share this" box 191
Adding photos from Flickr 192
Adding comments 193
Moderating comments 194

13 Content management systems 195

What is a CMS? 196
What is WordPress? 197
Setting up WordPress 198
Adding pages 200
Adding posts 201
Managing comments 202

14 Testing and launching 203

When is it ready to launch? 204
Testing your website 205
Technical testing 206
Usability testing 207
Publishing your site by FTP 208
Updating your website 210

15 Promoting your website 211

Introducing search results 212
How people search 213
Researching key phrases 214
Where to put search terms 216
7 top tips for SEO 218
Submit your website 219
Using keyword advertising 220
Display advertising 222
More promotion tips 223
Bringing visitors back again 224

16 Measuring success 225

Measuring visitor interaction 226
What the numbers mean 228
Other important metrics 229
Creating a survey 230
Enabling evolution 231

Credits 232

Index 233

1 The web design challenge

Web design is the art of creating websites that are attractive and easy to use. But it's complicated by all the different browsers and devices visitors use. Learn about the principles for building a site that keeps everyone happy.

10 The goal of this book
11 The diversity of devices
12 How devices affect design
14 Introducing key technologies
16 Degrading gracefully
17 What about mobile?
18 Understanding accessibility
19 Top accessibility principles
21 DIY or outsource?
22 How to set up your website

The goal of this book

The most exciting thing about the Internet is not the way it's transformed virtually every industry, nor the way it's enabled us all to be better informed about the world, and more in touch with our family and friends, than ever before.

You can find supporting resources for this book, including all the links from it (so you don't have to type them in) at www.ineasysteps.com or www.sean.co.uk

It's the fact that anyone can join in. If you've got something to share, the world is waiting.

You don't need anyone's permission to set up a website. You don't even need that much money: you can sometimes host a site for free and can rent a domain name for a year for about $10 or £10. Of course, you might choose to invest a bit more. You might want to pay for professional hosting with more advanced features.

If you're promoting your business, you might prefer to commission professional designers, and maybe even pay someone to help promote your site.

If you're doing it all yourself, you'll most likely pay through the time you spend on the design, although you might splash out on some software that makes it easier.

In this book, I'll teach you the key principles of web design, which is the art and science of building effective and attractive websites. This book will introduce you to the key technologies of the web, and the design principles that underpin successful sites.

Whether you're a budding designer who plans to build your own site, or you just want to be able to talk to professional designers in their own language, this book will give you an understanding of how websites are designed, so that you can make the right decisions in launching your own site.

I won't lie to you and tell you that website design can be mastered in 240 pages. The best websites are built using a combination of technical and artistic skill, and a good understanding of how people use the Internet. Above all, it takes practice to learn how to make the most of the technologies the web offers.

But, as the philosopher said, every journey begins with a single step. And this book will ensure you set off in the right direction, with robust knowledge of not just the technologies and techniques you could use, but of those that you should use, and those that will deliver the best experience for your visitors.

The diversity of devices

One of the skills a web designer needs is the ability to put themselves into the shoes of the website visitor. Web designers have to think about what their visitors know, what they will expect the website to do, and how they will expect it to work.

The most basic part of this is understanding the range of different devices that people might use to view your website. These include:

- A desktop computer, laptop or netbook
- An Android phone, iPhone or other handheld device
- A tablet device, such as an iPad
- An cell phone with a small screen and keypad
- A screenreader, which reads web pages aloud to blind people
- A refreshable Braille display, which a blind person can run their fingers along to read the website content
- A games console, such as the PlayStation, Nintendo Wii or Nintendo 3DS

When we use the word "design", we tend to think of something visual. But, in its purest sense, a website isn't necessarily a visual medium. Somebody using a screenreader might experience it as a stream of spoken text. Some handheld devices with small screens include a mode to switch off the layout and just view the content, so that it fits better in the space available.

While you do need your website to look appealing, it's a mistake to think that you can (or should) focus purely on the look of a website. Some of the most important work in web design goes on behind the scenes, where the visitors can't see it. It's about adding meaning to the web page, so that people can use it easily even if their device doesn't support all the features available on a desktop computer.

The challenge of web design is to create a site that is engaging and easy to use, whatever is used to view it. Visitors want the flexibility to use whichever device they prefer. Sometimes they will combine devices, using a desktop during their lunch break at work and a mobile on the way home, for example. All they care about is whether the site works or not. It's your job to make sure it does.

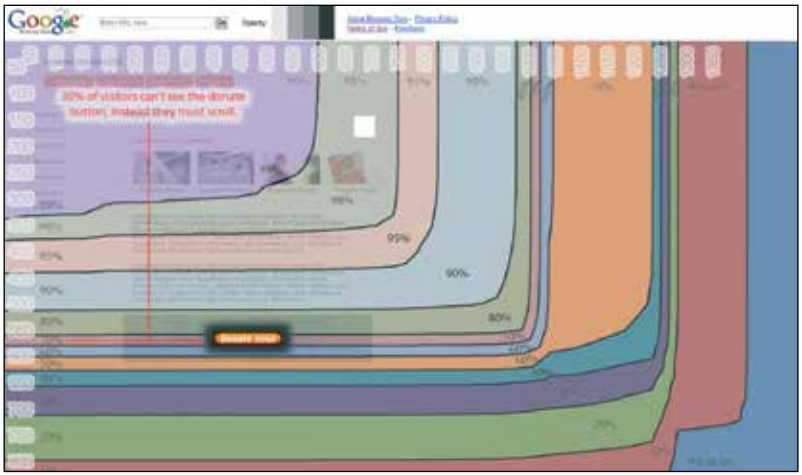
Websites can be viewed using a number of different devices. Shown above, from the top: Apple iPad tablet; Nintendo DS games console; HTC mobile phone; Alva refreshable Braille display.

How devices affect design

Different devices have different capabilities, and this should inform the decisions you make while designing your website. For example:

- There might be no support for pictures (or "graphics") on the device. A screenreader and Braille display can't show images, and mobile devices sometimes enable users to switch them off to speed up their browsing.
- There might be no conventional keyboard. It puts everyone off if they have to type in lots of information to use a website, but those who have to use a virtual keyboard on their touchscreen might be especially deterred.
- The website animation technology Flash might not be supported. The iPhone and iPad don't use Flash, and non-visual devices (such as screenreaders) can't render it either.
- There might be no mouse. Some users struggle to use a mouse because of physical impairment, and some devices (including games consoles and netbooks) don't have a mouse. If you demand precise use of a mouse, you might lose visitors.
- There is a wide range of screen sizes. Even on a desktop computer, people will have different sized monitors and will open the browser window to different widths, depending on what else they're doing at the same time as web browsing. Designers tend to have large monitors, but they shouldn't forget that most of their audience have to settle for smaller screens.
- Devices differ in how much of the web page can be seen at once. Mobile devices might provide a small picture of the whole web page with little detail, so users can identify the content to zoom in on. Clear headings are important to help users navigate to the section they should read. Users with poor vision sometimes use screen magnifiers to massively enlarge a small part of the screen. If people are zoomed in on one part of the page, they can't see any updates you make to a different part of it.
- A screenreader user can't get a quick overview of what's on the page by skim-reading it: while a PC screen is two-dimensional, a screenreader has a one dimensional interface: a stream of audio reading the web page aloud.

Web browsers are mostly free of charge, so why not download a few of the popular ones so that you can try them out? You'll get a taste of how they differ from your browser of choice, and can use them for testing your website design as it evolves.



Left: Google mocked up a charity website to show that 30% of people can't see the donation button without scrolling if it's too far down. In Google Analytics (see Chapter 16), you can see a similar graph for your website. Go to the Behaviour section on the left, then click In-Page Analytics, and click the Browser Size button on the right.

The web browser challenge

As well as the differences in hardware used to visit websites, people often have a choice over which software they use. On a desktop computer, they might be using Microsoft's Internet Explorer or Apple's Safari browser. They might have downloaded Google's Chrome browser or the Firefox browser.

They might even use more than one browser, depending on whether they're at work or at home.

That's another reason website design is a challenge: Web designers have to build sites that work with a wide range of different browsers, and each browser has its own quirks.

It doesn't end there, though. These browsers are updated regularly, but not everybody upgrades at the same time. Internet Explorer version 8 (IE8) was released in 2009, but it is the last version to work on Windows XP, so it's likely to have a significant market share for some time yet.

Despite high profile campaigns in the web community to persuade people to switch to something more secure and more fully featured, many people are still using IE6, which was released way back in 2001. If you're a web designer, you can choose to ignore it, but are you willing to turn away those customers? It's better to create a site that works for everyone, and let your visitors decide how they want to experience it.

Use your web analytics software to identify the most popular browsers among people who visit your website. Bear in mind, though, that anyone who can't use your website easily might just leave. See Chapter 16 for more on analytics.

Introducing key technologies

A website is made up of lots of different technologies and types of file. Part of the challenge of web design is learning how to use these different tools and technologies together, and how to pick the right ones for each aspect of your design. We'll cover the most important of these later in the book, but for now, here's a summary of the role they can play in your website:

HTML

HTML is the main language used for writing web pages. When somebody downloads one of your web pages, the first thing they are sent is an HTML file. HTML enables you to describe the structure of your text content, and to tell the web browser where it can find the other files for the web page, including its images and CSS files. HTML5 is the latest version, but many people are still using browsers that don't support its new features, so you need to make sure that your website also works without these.

CSS

CSS is used for the design and layout of your web page. It contains instructions for colors, fonts, and the position of different elements on the screen. CSS3 is the latest version and introduces features like rounded corners and multiple background images.

JavaScript

JavaScript enables you to update your web page after it has downloaded, and to respond to user actions on it. Some people have written sophisticated games using JavaScript, but it's typically used for simple effects like refreshing the screen with the latest news, or revealing some new content on the page when somebody clicks on a tab. JavaScript is also often used to check that a form has been completed correctly (for example, by checking there is an @ sign in an email address). This check can be performed before the user submits the form, so that the user gets instant feedback. Not all devices support JavaScript, so you need to ensure that your site (especially its navigation) also works without it.

Java

Confusingly, Java and JavaScript are not the same thing at all. Java enables you to run little programs inside your web page, as long as the browser has a Java plug-in. It's extremely slow, so almost nobody uses it for new websites today. If you already have web content in Java, consider recreating it using JavaScript or Flash.

Your website will also include a file for each image, and might also include files for audio recordings, videos and downloadable files like brochures in the printer-friendly PDF format.

How quickly a web page downloads depends on how many files it has, and how large they are in total. Each additional file results in another request to the server, which takes time, in addition to the time taken to download. To speed up a page, minimize the number of different image, style sheet, JavaScript and Flash files you use. You can minify your HTML, CSS and JavaScript, which means taking all the unnecessary spacing out to make files smaller. Major corporate sites set a limit of 300kb per page.

Flash

Adobe's Flash is used for more sophisticated interactions. You can use it to create animations and games or to display rich media, such as audio or video. Flash requires a plug-in to enable users to see and interact with its content. iPhone and iPad owners can't use Flash on their devices.

Like JavaScript, you shouldn't depend on Flash for features of your website you can easily implement another way.

A plug-in is software that is added to a browser to enable it to handle new types of content. If users don't have the plug-in, they won't be able to view that content, so you'll need to provide alternative content, too. Very few websites will be able to convince people to install software just to view that site, so you should stick to using plug-ins that are already popular.

AJAX

AJAX is a special way of using JavaScript to update a web page, by getting new information from the server without updating the whole web page. When you use Google Maps, for example, it doesn't download the entire planet's worth of maps to your computer. It just downloads the bits you want to see. If you zoom in, it downloads a higher resolution map of the area you're looking at. If you pan to another town, it fetches its map from the server. AJAX is the term used for the combination of technologies that enables the website to fetch the new information you need, whenever you need it. If you need to use AJAX techniques, you'll need to master HTML, CSS and JavaScript first.

Server scripting languages (PHP/MySQL)

There are a number of different server scripting languages, and they tend to go through fashionable phases. The most popular one today is PHP, but others include ASP and Perl. These enable you to write programs that run on the server, so that web pages can be customized for each visitor. This is particularly useful if the site needs to offer personalized content, such as a social networking site that will want to serve a different home page to each person depending on who their friends are.

Server side scripting languages are also used for sites with lots of pages. Their content can be stored in a database, and when somebody wants to look at it, the program on the server makes the web page HTML and sends it out. It saves you having to maintain hundreds of web pages, because you can just take care of one template, the scripting program, and the database of content. See Chapter 13 for more on content management systems, which enable you to do this.

MySQL is the name of the database software most often used with PHP. Hosting companies will charge a bit more for servers that support PHP and MySQL, but can usually help you to set up the database. You can find lots of free PHP scripts you can install on your server to offer features like forums.

Degrading gracefully

Take every opportunity to try out your website on unfamiliar devices. It's impractical for most people to own every device available, but if you come across one in a shop or at a friend's house, don't miss the chance to see how your site performs.

You might be thinking that the solution is to create a different website for all the different user groups you have. You're not the first to think of this.

Some years ago, it was fashionable for organizations to create a separate, text-only version of their website for the benefit of screenreader users. These users have requirements that are extremely different to the typical PC user, so many companies thought it would be easier to make a separate site for them than to reconcile their needs with those of mainstream website visitors.

Having a separate screenreader site created two major problems. The first was that this site inevitably ended up getting neglected. When resources were scarce, website managers would prioritize updating the main site. The text-only version would be put to one side and, ultimately, be forgotten. A screenreader user might visit the site for the latest news and find that it wasn't available to them, although it was available to others using the main website. The organization's attempt to look after blind visitors by specifically catering for their needs ultimately resulted in them being discriminated against.

The second problem was more subtle. Some screenreader users felt they were missing out because the organization had stripped out all the visual content. The fact that they couldn't see the content didn't mean that they weren't interested in what was in it, or what its message was. Pictures often provide context or supporting information for text or visitor comments.

A website could use the new CSS3 properties to create rounded corners on a box of content. If the browser doesn't support CSS3, the box will have square corners instead. This is a good example of degrading gracefully. The content can all still be accessed, but its presentation has adapted to work on a more limited browser.

One site for all

Today, it's not necessary to create a separate website for specific audiences. There are too many different devices and configurations for you to cater for them all anyway. Instead, you should aim to create a single website design that degrades gracefully. That means that it can still be used if particular features aren't available on the visitor's device or browser. For example, if somebody can't see a photo, you should provide text that explains its content. If the web browser doesn't support content created using Flash or Javascript, visitors should still be able to navigate the site and use the rest of its content.

A well designed website will work on any device and web browser.

What about mobile?

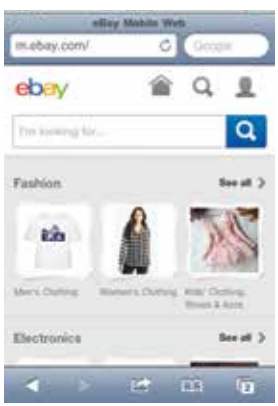
There is one exception to the rule of "one site for all", which you might have come across. Many companies today are creating separate mobile websites. The difference in size between the largest available PC screen and the smallest mobile screen is so vast, some companies argue that serving two separate websites is the best way to keep both audiences happy.

While websites should always be optimized for speed, users of mobile devices often have a slower web connection than those using a desktop PC, too, so they need a site that is optimized to download in small chunks, with excessive decorative images stripped out.

There are two times when it's a good idea to create a separate site for users of mobile devices, assuming a significant chunk of your audience might want to use mobile devices:

- **If you have a tiny website.** You can probably maintain two different versions of a website easily if it's only got five pages.
- **If you have a vast website.** If your site has hundreds of pages, you'll probably need to use a content management system for it (see Chapter 13). This should make it possible to have your mobile site automatically generated from the same database of articles as your main website. That will ensure your mobile site always reflects the same content as the main site, and will update at the same time.

It adds a lot of complexity to maintain more than one website, however you do it. You'll have to decide whether the extra work is justified for your website.



Mobile sites like **m.ebay.com** and **mobile.wikipedia.org** streamline the site design so they're easy to read on mobile devices. Both sites use a single column, instead of the multiple column layout used on the main websites.

Understanding accessibility

Accessibility is all about making sure your website can be used easily by everybody. When people talk about accessibility, they often think of blind people using screenreaders. But it's not just about those who use completely different technologies to visit your website.

There are lots of people who have more subtle needs that you can help or hinder through the way you design your website. A lot of these people wouldn't consider themselves to have any special needs at all. They'd just find your website hard to use, and give up. For example:

- Can somebody who is color-blind understand the availability of tickets, or is it shown as purely different colored blobs?
- Can somebody who has impaired vision increase the font on your web page so it's easier to read?
- If somebody can't use the mouse, can they navigate your site using the keyboard?
- Can a visitor with impaired hearing get the same information that you've included in an audio file?
- Can somebody with a mental impairment understand your content, or is it too jargony and long-winded?

Most professional website designers today understand the importance of accessibility, but it's the least visible aspect of a website design project, so it can be overlooked. Much of the important work is to do with how a website is coded, and when you visit it using a conventional desktop PC, you can't easily tell whether it's been well designed for other devices or not.

Accessible websites tend to be easier for everybody to use. They adapt well to mobile devices, and are easier for search engines to index, too. If you need to convince colleagues of the importance of accessibility, you could argue that you want to reach as many customers as possible, or could point out that your business might contravene disability discrimination laws if you don't.

Most websites strive for accessibility because they want to be as inclusive as possible, though. It's not hard to make most websites accessible, and it doesn't cost any more, as long as you plan to make your site accessible from the start.

If you outsource your design, take particular care to make sure the designers are creating an accessible design. If you're never asked to provide any alternative text, ask yourself whether they're making it up for you, or not bothering with it at all.

Try navigating the web by keyboard for yourself, especially your own site when it's ready for testing. In Google Chrome, click a web page and then use the Tab key to move through the links. Press Enter to visit a link, and the Backspace key to go back a page. Use the Up and Down arrow keys to scroll the page.

Top accessibility principles

I'll share some tips for making your site accessible when we look at HTML and navigation later, but to provide some context, there are some basic principles that can go a long way towards making your site easy to use.

1. Provide alternative text for anything that isn't text, such as an image or video. This does not necessarily appear on screen. When you add an image to a web page, for example, you also provide alternative text that is only presented to the user if the image isn't available.

2. Make sure that your site still works when JavaScript, Flash or other interactive elements are switched off. When this is impossible, provide the same information in text form on an accessible web page.

3. If users have to respond within a certain time limit, make sure that they are warned and have enough time to request more time, if they need it. Give users controls to start and stop content that moves or updates itself.

4. Use clear and consistent navigation. Make sure that users can easily understand where each link will take them, and give each page a title that describes its purpose.

5. Make sure the website can be used with the keyboard, and that users can see what link or element they are focused on.

6. Make it easy to skip straight to the content. If your web page has a long list of links or header information, provide a link that enables people to jump straight to where the article starts. This is particularly important for screenreader users.

7. Don't open pop-up windows, or change the window the user is in, without warning. This is confusing if you can't see the screen, and it stops the Back button working, the most important navigational tool on the Internet.

Make sure your alternative text conveys the same information that it is replacing, but keep it brief. You can say "close up of clock tower", or "the gold detailing of Big Ben shines on a sunny day". Ask yourself what the purpose of the image is, and try to articulate that. You don't have to mention the blue skies, or where the big hand is pointing, unless it's important.

Don't provide alternative text for purely decorative images. Make these background images, so that assistive technology can ignore them.

...cont'd

For detailed guidance, see the Web Content Accessibility Guidelines (WCAG) published by web standards body W3C (**www.w3.org/TR/WCAG20/**). Section 508 (**www.section508.gov**) outlines accessibility requirements for US government agencies, and includes good advice for all websites.

There are several automated accessibility checkers online, including Wave (**http://wave.webaim.org/**) and Cynthia Says (**www.cynthiasays.com**). They test your website against the WCAG guidelines. They help to identify problems, but accessibility is about whether people can use your site, not whether it appears to tick a particular box. Use your own judgement and feedback from users, too.

8. Use the simplest language appropriate for your site's subject matter. Complicated language is harder to understand, especially if you can't easily flick your eyes back to the previous line for a refresher on what it said.

9. Give users control over how they view your content. Make sure your site works, for example, if people enlarge the default text size in their browser.

10. Make sure that any information conveyed by color can also be understood in other ways. For example, it's okay to use different colors to draw attention to different icons, but you might also make them different shapes or put them in different columns to indicate their meaning.

11. Make sure there is enough contrast in the foreground and background colors you choose for everything to be legible on a black and white screen.

12. Avoid overcomplicated page layouts. If there are too many different sections or boxes it can be overwhelming for visitors to try to navigate.

13. Use HTML and CSS correctly. Test whether the code has any errors in it. Desktop browsers can be quite forgiving, but assistive technology might struggle to understand your web page if it includes technical errors.

14. Use the accessibility features available in HTML to help people navigate tables and forms. They enable you to provide a context that might otherwise be communicated by the position of things on the screen.

15. Don't use a picture of text unless it's essential. It's a good way to show a company logo, for example. But don't create an image for a paragraph of text just because you like a particular font.

DIY or outsource?

Should you design your site yourself, or pay somebody else to do it for you? It depends on what you want to achieve with your website design project.

The strengths of DIY

Creating your own website can be an extremely satisfying creative project. People build websites for the same reason they write novels and sing songs: it's fun, and it's rewarding to increase your expertise over time. Building websites has an added dimension, because what you make can be easily found by others, so you can get feedback. It's also a cost-effective approach, compared to hiring a team of professionals.

If you want to build your own website, you can take one of two paths. Firstly, you can learn how HTML and CSS work and code your site by hand. These technologies aren't too hard to learn, and this book includes a quick tour around their most important features to get you started. Hand coding your site gives you maximum control and flexibility.

Alternatively, you could use a visual design package. These enable you to build a website in the same way you might make a newsletter in a word processor. They make it much easier to get started with website design, although it can take a lot of effort to build something impressive.

Once you know how to create websites, you could buy a template and adapt it to accelerate your web design process.

Outsourcing your design

Working with a web design firm is likely to lead to more professional-looking results first time. You can benefit from the firm's experience of designing perhaps hundreds of websites. You can focus on your content and the site's purpose and don't have to worry too much about technical details.

That said, it's still *your* website, and you need to take responsibility for managing the designers. You need to understand enough about web design to know whether their proposals are a good idea or not, and what the technical limitations might be on any designs they propose.

You might also want to learn enough HTML and CSS to be able to edit the design after the project has ended.

If you commission somebody else to build your website, make sure you have the ability to update it easily later.

Make sure you plan enough time for designers to do a good job on your website. Design is a process, with designers suggesting ideas and you providing feedback, until you reach the design you want.

How to set up your website

Now that you understand the design challenge, you're ready to take it on! Here is an outline of the process, although your own project might include extra steps, such as getting the CEO to approve the content:

1. **Planning.** Time invested in planning at the start will avoid rework later. In particular, a clear understanding of the site's purpose should inform the whole process.

2. **Content creation.** You need to write your text and gather your pictures so that the website can be designed to present them in the best way.

3. **Web design.** This involves many small steps. You'll need to structure your content, and design how it is presented. You also need to think about how people will move between different parts of your website. You might want to integrate a shopping cart or social networking features. You can commission a web design firm, or you can learn the technologies to design it yourself.

4. **Testing.** To make sure your design works, you need to test it at regular intervals. Design is often an experimental process, with the designer implementing an idea, testing it with prospective site visitors, and using the feedback to refine the design.

5. **Launching the website.** To make your website available to the public, you need to copy it onto a web server, which will send the web pages to your site visitors over the Internet. You will need to rent web server space from a hosting company.

6. **Promoting the website.** Attract visitors by making it easy for people to find your site through search engines, adverts, and links.

7. **Measuring your success.** Find out how well your site is fulfilling its purpose, and use that information to refine it.

If you want to practise your web design skills but don't have any content available, there's an exercise on my website, www.sean.co.uk, you could try. It provides a pack of text and photos for the fictional Castle Fictoria hotel.

There are often multiple ways to do the same thing using software. I can't show you them all in this book, so feel free to experiment if you spot a shortcut or alternative approach that you prefer.