Producing for Web 2.0

A student guide

Third edition

Jason Whittaker

Producing for Web 2.0

‘

Producing for Web 2.0 is a clear and practical guide to the planning, set up and management of a website. It gives readers an overview of the current technologies available for online communications and shows how to use them for maximum effect.

The third edition sets out the practical toolkit needed for web design and content management. It is supported by a regularly updated and comprehensive website at www.producingforweb2.com where readers can take part in blogs and forums, see examples of programming and demonstrations of concepts discussed in the book, as well as try things out themselves on the testing site.

Producing for Web 2.0 includes:

- illustrated examples of good page design and site content,
- comprehensive online support and testing areas,
- advice on content, maintenance and how to use sites effectively,
- ideas on how to maximise available programs and applications,
- tips on using multimedia, including video, audio, Flash and images,
- a glossary and a list of terminology,
- a chapter on ethics and internet regulations for journalists and writers,
- tutorials for the main applications used in website design,
- step-by-step guides to difficult areas with screenshots,
- guides to good practice for all those involved in journalism, broadcasting and media studies, and
- a list of resources including websites and guides to further reading.

It is the perfect guide for anyone coming to web design for the first time, or producing multimedia materials.

Jason Whittaker teaches on the Journalism programme at University College Falmouth, and is also Professor of English and Media Arts. He has more than 15 years’ experience as a technology journalist and is the author of several books on new technologies, including *The Cyberspace Handbook* (2003) and *Web Production for Writers and Journalists* (2002), as well as several books on William Blake. His most recent publication is *Magazine Production* (2008).

Producing for Web 2.0

A student guide

Third edition

Jason Whittaker

Routledge
Taylor & Francis Group
LONDON AND NEW YORK

First published 2000 as *Producing for the Web*
by Routledge
2 Park Square, Milton Park, Abingdon, Oxon OX14 4RN

Simultaneously published in the USA and Canada
by Routledge
270 Madison Ave, New York, NY 10016

Second edition published 2002 as *Web Production for Writers and Journalists*

Routledge is an imprint of the Taylor & Francis Group, an informa business

This edition published in the Taylor & Francis e-Library, 2009.

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Cataloging in Publication Data
Whittaker, Jason, 1969–
Producing for Web 2.0: a student guide / Jason Whittaker. — 3rd ed.
p. cm.
Rev. ed. of: Web production for writers and journalists. 2002
Includes bibliographical references and index.
1. Web sites—Design. 2. English language—Composition and exercises. 3. Web site development. 4. Authorship. 5. Web publishing. I. Whittaker, Jason, 1969– Web production for writers and journalists. II. Title.
TK5105.888.W48 2009
006.7—dc22

2008053200

ISBN 0-203-88203-2 Master e-book ISBN

ISBN 10: 0–415–48621–1 (hbk)
ISBN 10: 0–415–48622-x (pbk)
ISBN 10: 0–203–88203–2 (ebk)

ISBN 13: 978–0–415–48621–7 (hbk)
ISBN 13: 978–0–415–48622–4 (pbk)
ISBN 13: 978–0–203–88203–0 (ebk)

For Sam, for her patience

Contents

Walkthroughs

Preface to the third edition

When *Producing for the Web* was first published in 2000, the world of web design was very different from that of today. The first dotcom boom had taken place (shortly to be followed by a dotcom bust), and while a number of the key technologies had already been established, the transformation in web publishing that has come to be known as **Web 2.0** was still to occur.

That first book concentrated principally on HTML and general principles of design. By the time it was revised, as *Web Production for Writers and Journalists* in 2002, Dreamweaver was being accepted by many as the industry-standard application for web design. That edition concentrated much more on creating websites using Dreamweaver, as well as introducing more substantial sections on providing content for the web.

In the intervening six years, online publication has changed dramatically. Blogs, while not entirely new in 2002, were rarely used, and many of the most popular sites on the web today, such as Wikipedia, Facebook and YouTube did not exist or had only just been founded. Even Google, far and away the most successful of the 'new' new media companies at the time of writing, was little more than a humble search engine, without a host of add-on utilities such as maps, email and even an office suite.

A big difference between *Web Production for Writers and Journalists* and *Producing for Web 2.0* is the status of Dreamweaver – and, indeed, any other web design application. Steve Hill, citing both Khoi Vinh, the design director for NYTimes.com, and Dave Lee, at UKPG, summed up the situation on his blog (srh.typepad.com/blog/2008/04/index.html) with a very simple question: 'Who's using Dreamweaver?'

These and other commentators have drawn up a wish list of the various skills that the ideal recruit would have (note: terms below and others elsewhere that appear in bold are explained in the glossary at the end of the book):

- an extensive knowledge of **XHTML** and **CSS** (cascading style sheets), the fundamental technology for creating web pages and determining their appearance,
- a sound knowledge of a scripting language such as JavaScript, to provide greater interactivity,
- an understanding of Flash, the main way for developing 'rich' applications online,
- a 'comfort' level with database and application programming languages such as MySQL and PHP being used to drive many dynamic sites online,
- multimedia skills with image, audio and video,
- last but not least, a solid foundation in news values – that is, knowing how to write in a way that will attract readers and convey information quickly.

The one skill that is rarely – if ever – required is a knowledge of Dreamweaver. And yet, as Hill points out, for a majority of students engaged in media studies, and some teachers too, activities such as online journalism are the same as creating websites, which amounts to learning Dreamweaver. The advantage of Dreamweaver as an 'industry app' (or, at least, that is how Adobe would like to see it) is that teaching the package provides a clear rationale for having something to teach – something that also costs money and is thus of value to the 'industry' (as with other sectors, such as page layout with InDesign, image editing with Photoshop, video editing with Avid, or computer aided design with AutoCAD).

But the vast majority of web production simply does not work that way these days. Not only was the original web built on open standards (such as **HTML**), but many of the most dynamic and useful systems for getting content online – **wikis**, **blogs**, **content management systems** (**CMSs**) – are also available as free, open-source products. Such applications can teach much more valuable transferable skills than learning one particular interface: despite the fact that very often one particular site or system such as Joomla! or Blogger may have superficial differences to the next, underlying principles are frequently the same. Despite this, I am not completely sceptical about the value of using Dreamweaver for certain tasks, and throughout this book the reader will encounter various examples of how to achieve particular aims using that program.

The ideal skills for web production include a knowledge of XHTML and CSS for creating and formatting web content, as well as dynamic scripting, multimedia skills, database design and knowing how to write in a way that will attract readers and convey information quickly.

The ideal web producer, then, would be a coder with a solid understanding of XHTML and CSS in particular, but also familiar with other programming and scripting languages as well as database design. Furthermore, he or she would be very competent with a wide range of multimedia formats, and an expert writer to match. The list sounds formidable – and in many ways it is. In practice, there are plenty of very successful web producers who only touch the most essential elements of design and concentrate instead on producing content, whether in the form of text or other media. What is more, the various platforms covered in this book make it easier than ever to get material online, so it is more than possible to publish your work without ever understanding a single line of code.

Despite this, for those who wish to move beyond the basics, a core knowledge of the technologies involved in online publishing is immensely valuable. What is more, while the media landscape for web production has changed immensely since the publication of *Web Production for Writers and Journalists*, an emphasis on key writing skills remains very important, and this title is in some ways a return to the grounding in design and underlying programming techniques similar to that covered in the first edition of *Producing for the Web*.

After introducing the reader to the principles of Web 2.0 in chapter 1, what makes it different to web production during the 1990s, chapter 2 will look at what is required to plan and prepare before you begin to build a site. Chapter 3 covers the core skills for web design, both in terms of general design and navigation as well as using XHTML and CSS. These skills are expanded into client- and server-side scripting in chapter 4, before moving onto how to use multimedia in chapter 5. *Producing for Web 2.0* assumes a certain degree of continuity between core web design skills and publishing platforms which can be considered as closer to Web 2.0 models, these being covered in chapter 6. Chapter 7 covers content management systems, specifically Joomla! Writing, which still remains central to most of the content that appears online, is the subject of chapter 8, while the final chapter on post-production considers the testing and promotion of your site, with particular emphasis on new requirements for modern web production, such as optimising a site for search engines.

THE *PRODUCING FOR WEB 2.0* SITE

To accompany this book, the companion site includes extended examples of coding included in the text, as well as technology updates and a blog with extended articles on new developments in the web world.

You can find the site at http://www.producingforweb2.com.

CHAPTER 1

Introduction

Since its invention by Tim Berners-Lee in 1990, the web has rapidly transformed the means by which information can be published and disseminated. Central to the original ideal of the web was the ability to transfer data regardless of the platform on which it was viewed: so long as a visitor had a browser, it did not matter which hardware or operating system he or she used to get online.

Since about 2004, however, the ease and capabilities of the web have undergone considerable changes – what is commonly referred to as Web 2.0. This chapter will begin with an outline of the principles of Web 2.0 publishing, as well as the various options open to web producers.

THE INTERNET AND WEB 2.0

Web 1.0

There are plenty of books that have appeared in recent years on the history of the internet. As the focus of this title is recent developments that have been bundled together under the title 'Web 2.0', the context of web development throughout the 1990s can be dealt with very quickly.

The beginnings of the internet, as opposed to the world wide web, lie in the Cold War and plans to build a communications structure that could withstand a strategic nuclear attack. DARPANET (the US Department of Defense Advanced Research Projects Agency) launched the first network in 1968, and through the 1970s and 1980s various research and military institutions connected to this backbone.

Until 1990, however, the internet was still very much an esoteric and restricted concern. What changed this was the work by Tim Berners-Lee, a consultant at the European Centre for Nuclear Research (CERN), who wrote a short program, Enquire-Within-Upon-Everything, or ENQUIRE, that enabled electronic documents to be linked more easily. A year later, he developed the first text web browser, NeXT, and so launched the world wide web.

CERN continued to develop the web as an academic tool, but by the end of 1992 only 26 hosts were serving websites, growing slowly to 1,500 by 1994. The boom in web (and internet) usage came that year when Marc Andreessen, at the National Center for Supercomputing Applications, developed a graphical web browser, Mosaic, and then left to form a new company, which was to become Netscape Communications.

At the same time, developments in personal computing, such as the decline in price of PCs and the launch of a new operating system, Windows 95, meant that more people than ever before were starting to use computers as part of their daily lives. While Microsoft had originally been dismissive of the internet, by

1997, with the launch of Internet Explorer 4 as part of the Windows operating system, they began to pursue this new market much more aggressively (too aggressively according to the US Department of Justice).

The late 1990s saw the dotcom bubble expand – and then burst. Paper millionaires appeared and disappeared in the space of a few months, and a post-millennium malaise set in when it seemed for a few years that nothing good could come out of the overvalued medium.

Yet the investment and innovation that took place in those years did have some incredibly important consequences. While many half-baked websites (quite rightly) disappeared without trace, some such as Amazon, eBay and Google became household names. Internet usage generally, and the web in particular, had become completely normalised in many instances, for some users displacing traditional media altogether as faster broadband connections rolled out in different parts of the world. At the same time, the often difficult process of getting content online was becoming increasingly simplified through such things as blogs, wikis and **social networking** sites, leading some commentators to remark on a new phase of web publishing – Web 2.0.

Core principles of Web 2.0 include using the web as a platform to run applications, rather than relying on the operating system, allowing users to take control of their content, and employing new methods to share that content more easily.

What is Web 2.0?

Web 2.0 is a term coined by Dale Dougherty of O'Reilly Media and Craig Cline of MediaLive prior to a conference of that name which took place in 2004. It is a rather loose term that refers to a collection of platforms, technologies and methodologies that represent new developments in web development.

The term itself has generated a considerable amount of controversy, most notably from Tim Berners-Lee, the inventor of the world wide web, who, in an interview for IBM in 2006, remarked that 'nobody even knows what it means'. Berners-Lee pointed out that the innovations implemented by Web 2.0 applications, for example simplifying the sharing of data and making online media much more inclusive, were actually pioneered as part of the development of the supposedly outdated Web 1.0. Likewise, Steve Perlman (the man behind **QuickTime** as well as many other innovations) more recently observed that many so-called Web 2.0 sites were really very static in their approach to content and lacked real multimedia support; many such applications, he observed in an interview with CNET, currently touted as cutting edge will be obsolete in only a few years.

In addition to such criticism, a more general observation is that certainly much Web 2.0 commentary is little more than internet marketing hype familiar from the dotcom bubble at the end of the 1990s. Despite these reservations (all of which are extremely valid), Web 2.0 is a convenient label to distinguish some real innovations that have taken place since the turn of the century. More than this, however, it recognises that recent years have seen a remarkable change in the applications of new technologies driven (among other things) by revolutions in computer usability and bandwidth.

In a blog entry in September 2005, Tim O'Reilly offered a succinct overview of what Web 2.0 was meant to achieve, observing that 'like many important concepts, Web 2.0 doesn't have a hard boundary, but rather, a gravitational core. You can visualise Web 2.0 as a set of principles and practices that tie together a veritable solar system of sites that demonstrate some or all of those principles, at a varying distance from that core' (O'Reilly, 2005).

Key elements of this 'gravitational core' include:

- using the web as an applications platform,
- democratising the web, and
- employing new methods to distribute information.

The implications behind a 'democratisation' of the web are contentious to say the least, and this idea is better limited to considerations of usability and participation rather than any implied political process (although that is often invoked), but these three bullet points in some shape or form do identify the nucleus of what Web 2.0 is meant to achieve with regard to platforms, participation, and data as the focus.

The web as platform

O'Reilly observes that the notion of the web as platform was not new to Web 2.0 thinking but actually began with Netscape in the mid-1990s when it took on Microsoft with the assertion that online applications and the web would replace Windows as the key operating system (OS). As long as users could access programs and data through a browser, it did not really matter what OS or other software was running on their desktop computer.

Several factors indicate the difference between online services in the 1990s and those currently labelled as Web 2.0, and also indicate why Netscape failed at the time:

- Limited bandwidth: for processing and delivering data, online services simply lagged behind and/or were too expensive in comparison to desktop applications at the time.
- Limitations of the 'webtop': Netscape's alternative to the desktop, the 'webtop', was much closer to Microsoft's core model than it assumed; achieving dominance by giving away a free web browser was meant to drive consumers to expensive Netscape server products rather than allowing them to plug into a range of services. The software application had to succeed for Netscape to be viable.

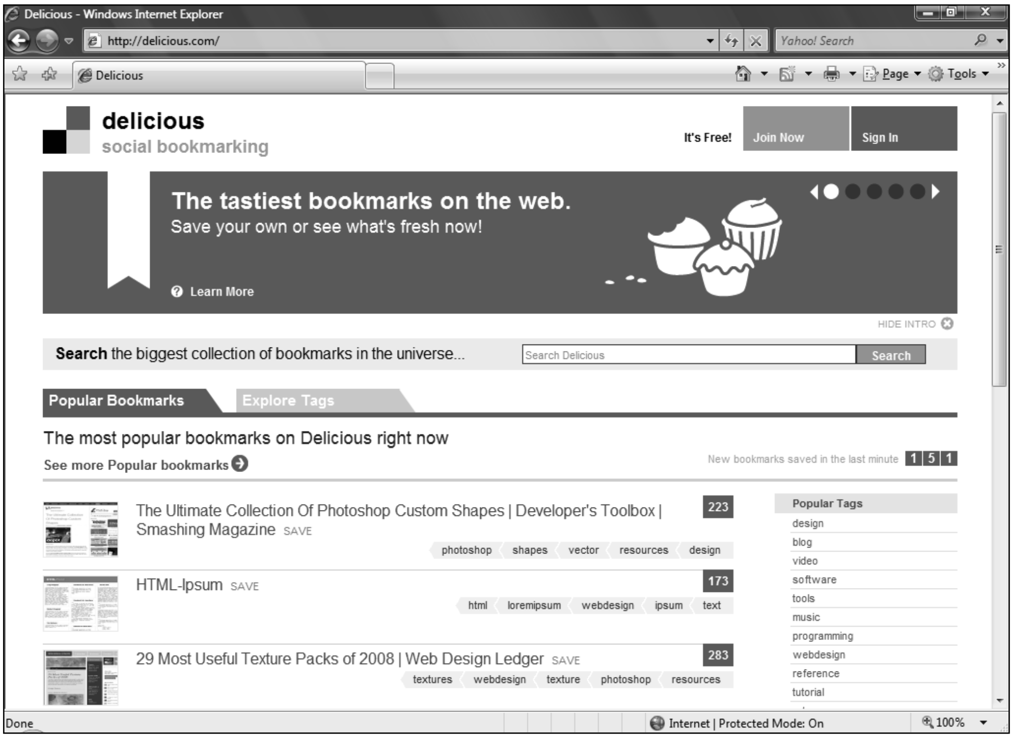
O'Reilly contrasts this approach to that of Google's, which began life as a web service: the ultimate difference, argues O'Reilly, is that with Google the core service is combined with the delivery of data: 'Without the data, the tools are useless; without the software, the data is unmanageable' (2005). Software does not need to be sold and licensing of applications is irrelevant, because its only function is to manage data, without which it is redundant: that is, it 'performs' rather than is distributed. As such, 'the value of the software is proportional to the scale and dynamism of the data it helps to manage'. Furthermore, data should be as easily exchanged as possible between different applications and sites.

Web 2.0 platforms are designed to make data accessible, regardless of its location, so that it can be exchanged as seamlessly as possible, providing services previously carried out on the desktop.

Rather than simply providing static information as was common to many (but by no means all) Web 1.0 sites, Web 2.0 services make much greater use of **applets** to use that data dynamically, for example to send messages to large numbers of users via a simple interface (Twitter) or share favourite links (delicious). Simplicity of use to the end user often belies very complex technology behind the scenes, and core technologies include server software, content syndication, messaging **protocols**, standards-based browsers (non-standard **plug-ins** are to be avoided as they cannot necessarily be installed on different devices) and client applications accessed through the browser.

The important features of the web as platform are as follows:

- to make data accessible from any platform connected to the internet, regardless of its location or the operating system,



delicious
social bookmarking

The tastiest bookmarks on the web.
Save your own or see what's fresh now!

Search the biggest collection of bookmarks in the universe...

The most popular bookmarks on Delicious right now
See more Popular bookmarks
New bookmarks saved in the last minute 1 5 1

The Ultimate Collection Of Photoshop Custom Shapes | Developer's Toolbox | Smashing Magazine — 223 — photoshop shapes vector resources design

HTML-Ipsum — 173 — html loremipsum webdesign ipsum text

29 Most Useful Texture Packs of 2008 | Web Design Ledger — 283 — textures webdesign texture photoshop resources

Popular Tags
design
blog
video
software
tools
music
programming
webdesign
reference
tutorial

delicious.com, one of the new generation of social bookmarking sites.

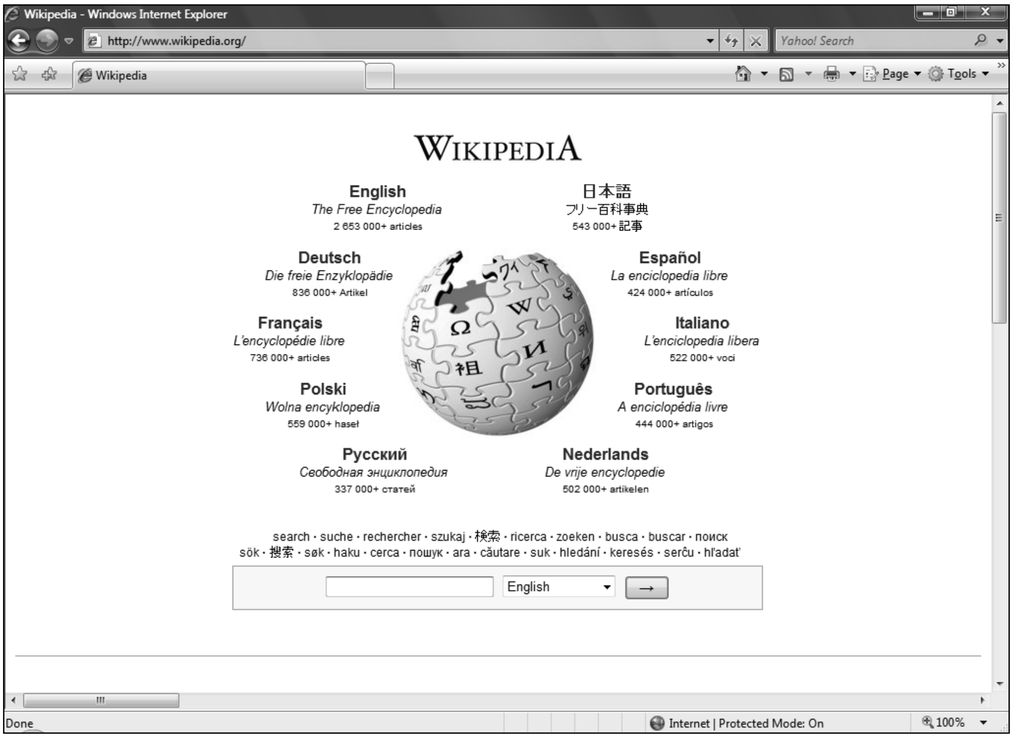
- to exchange that data as seamlessly as possible between different sites and applications, without the need for proprietary plug-ins,
- to carry out tasks previously carried out on the desktop via an online service and thus make them more easily shared,
- to use applets to provide a 'user rich experience'.

An architecture of participation

The 'democratisation of the web' is a phrase often used in conjunction with Web 2.0, but one that *Producing for Web 2.0* will avoid because of the assumptions it makes about democratic processes as ultimately being tied too often to consumption (this is not to deny a link between the two, but rather to draw attention to the limitation of such connections, a full analysis of which is beyond the scope of this book).

Rather, here we will use a more neutral term, again first used widely by Tim O'Reilly but with its roots in open-source software development and the ideas of Lawrence Lessig – an 'architecture of participation'. Such participation builds upon the interactivity that was an early part of web design, in contrast to other media which tend to emphasise passive consumers (not necessarily in interpretation, but certainly in terms of production) versus active producers. Despite the fact that Berners-Lee believed that early users would be authors as well as readers, and nearly every site had pages that required users to click hyperlinks to navigate a site, most such pages were very static.

In the mid-1990s, however, companies such as Amazon were actively encouraging visitors to post reviews of books, and of course there had long existed interactivity on such things as bulletin boards which started to transfer to mainstream sites. The development of audience interactivity, then, can be seen as an evolution of early forms of connecting people



Wikipedia is one of the most remarkable Web 2.0 sites to have emerged in recent years.

(as pointed out by Berners-Lee), emphasising the interaction of users with a site to a lesser or greater degree. At its most developed, this involves a much greater degree of trust in users so that, for example, with a wiki the process of editing and contributing is much more decentralised.

O'Reilly has spoken of the ability of sites such as Wikipedia to 'harness collective intelligence', although a consequence of this letting go of centralised control makes it much easier for users to enter incorrect information accidentally or deliberately. An important outcome of this simple fact is that for this to function correctly, it depends upon a community of informed and interested users to constantly monitor and moderate activity.

Attwell and Elferink (2007: 2) point out:

> the Architecture of Participation is not a software system as such – or even a collection of software tools – but rather a bringing together of various technologies and activities designed to facilitate and promote participation, communication and the active construction of meanings and knowledge.

Core to this is trust, that a site is not a 'walled garden' but something that should be as easy to enter and leave as possible (which in turn relates to issues of usability) and, to maintain this trust, that users' data belongs to them. In turn, this has drawn attention to the relationship between sites that involve some form of social networking or communal activity and ownership of intellectual property, with trends established by the development of open-source software and the role of organisations such as the Creative Commons (creativecommons.org) being important in developing new attitudes towards copyright, drawing a middle line between anarchic piracy that damages trust and over-restrictive regulations that stifle innovation.

The social impact of these architectures of participation is already proving itself to be immensely important, for example in the rise of the blogosphere (which encapsulates the successes and irritations of much of Web 2.0 capabilities). The combination of community and architecture draws attention to the main capability of Web 2.0 development: technology provides the framework to exchange data (the architecture) that should be as simple and seamless as possible, but without a community of users to produce that data in the first place the technology itself is redundant.

Data as focus

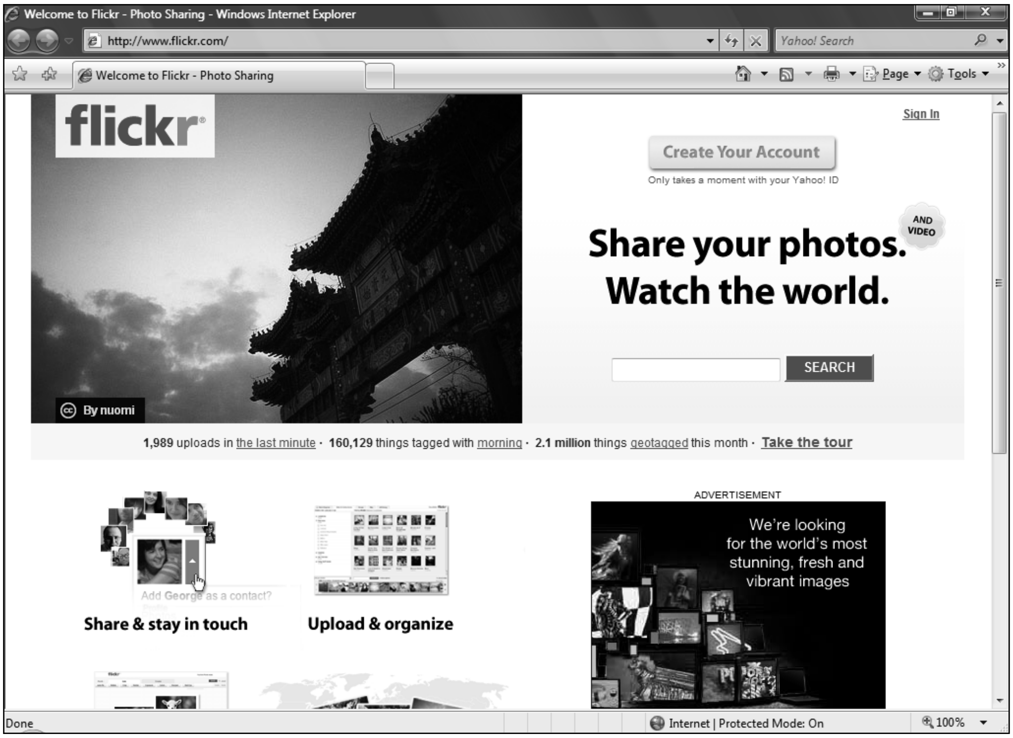
Web 1.0 was as much about information as Web 2.0, but the means for distributing data has changed significantly. One important consequence of creating participative architectures has been the growth of user-generated content (**UGC**), or consumer-generated media, referring to publicly available content produced by end users rather than the producers or administrators of a site.

Often UGC is only part of a site but in some cases, as with Flickr or YouTube, it constitutes the entire process (with attendant problems in terms of copyright with YouTube, for example). Some media organisations are therefore switching from being the providers of content to being the providers of frameworks and facilities where content can be shared.

The OECD (Organisation for Economic Co-operation and Development) emphasises that genuine UGC involves creative effort on the part of the contributor, rather than simply being material such as video or audio digitised from another source, and that it is produced outside the normal professional routines and practices.

In addition, whereas most sites until the dotcom crash tended to produce static

Flickr was one of the first sites to make sharing user content its core activity.


Welcome to Flickr - Photo Sharing - Windows Internet Explorer
http://www.flickr.com/
Yahoo! Search
Welcome to Flickr - Photo Sharing
Page
Tools
flickr
By nuomi
Sign In
Create Your Account
Only takes a moment with your Yahoo! ID
AND VIDEO
Share your photos.
Watch the world.
SEARCH
1,989 uploads in the last minute · 160,129 things tagged with morning · 2.1 million things geotagged this month · Take the tour
Add George as a contact?
Share & stay in touch
Upload & organize
ADVERTISEMENT
We're looking for the world's most stunning, fresh and vibrant images
Done
Internet | Protected Mode: On
100%

HTML pages that were relatively cumbersome to update, after this event many began to experiment with new ways to make data more interactive. One important development, which was initiated in the 1990s, was syndication or web feeds, whereby content on one site could be made available to multiple external sites or newsreaders on a user's computer. While the technology available for 'push' syndication (streaming data to subscribers) began to be developed around 1995, it was only from 2001 onwards that formats such as **RSS** (Really Simple Syndication, or RDF Site Summary) started to win widespread acceptance. The effect of syndication is that users can receive updates of changes to rapidly updated content, such as news feeds, blogs or forums, without the need to visit a site.

Really Simple Syndication (RSS) is an important way in which data can be shared and updated between multiple sites.

Two important foci, then, of Web 2.0 technologies are simplifying the process of creating information so that many people may contribute (sometimes referred to as crowdsourcing), and simplifying how data is shared. Blogs and wikis are good examples of the former, whereby users with minimal technical experience may enter or upload media much more quickly than was required when creating a personal home page. As well as syndication, a number of other methodologies have emerged for the sharing of data: tags are a means by which visitors may enter opinions and information about an article or piece of media which, in turn, will be picked up by search engines thus attracting other visitors.

In contrast to what O'Reilly distinguished as the Web 1.0 method of organising data – directories or taxonomies, which would require some form of centralised administration – Web 2.0 methods such as tagging create a **folksonomy** (also known as social indexing or tagging) whereby users apply their own categories to identify material of interest. Of course, the lack of control over terminology can create inefficient indexing (with plenty of synonyms – what is know as polysemy), but folksonomies have arisen due to the perceived inefficiencies of traditional web indexing or searches. In addition, the introduction of the **permalink**, a **URL** to point to a blog or forum entry after it has passed from the front page to an archive and which does not change, allows data to remain in circulation.

Another phenomenon associated with Web 2.0 is the long tail. Coined by Chris Anderson, editor in chief of *Wired* magazine, 2004, the long tail defines the process of focusing on less popular information or resources that previously were unviable because of some physical limitation. Also referred to as niche marketing, the long tail works by making available data to fragmentary audiences, in contrast to hit marketing, the process of constructing a few large-grossing hits because there is only so much space to show movies at the cinema, or carry DVDs, CDs or books on shelves. Anderson gives examples of the 1.7 million Indians living in the US who could only see Hindi films on a handful of screens, or the lack of documentaries available in Blockbuster. As Anderson points out, 'The average Barnes & Noble carries 130,000 titles. Yet more than half of Amazon's book sales come from outside its top 130,000 titles' (2006: 83). Digital distribution makes it possible to provide relatively obscure back-catalogues and, via search engines and other forms of organisation of data, connect that material to users on a wider scale – thus making them viable.

Always in beta

The adoption of the internet as a means for distributing software has resulted in new models that contrast greatly to previous cycles and which have consequences for distribution of other media.

Traditionally, a software concept would be proposed and written in alpha form, then distributed as a beta for testing and, after a release candidate was developed the final version would be distributed for sale. As applications became more complex, it became quite clear that bugs and improvements not picked up at the beta stage would need further patches to fix. By contrast, a great deal of software used by Web 2.0 sites is always in beta, drawing on the fact that open-source software in particular can be modified by a much wider range of developers and programmers than in traditional companies. The continuous development cycle for this new type of production process is often referred to as 'always in beta', with updates being released as and when they are created and/or tested, rather than waiting for a convenient date in the schedule.

The continuous cyle of development on new sites means that they are never final but 'always in beta', with users playing an important role in co-developing how they perform.

Another effect of this approach is to treat users as co-developers: just because a feature is available does not mean that it should be used. Overloading a site with additional abilities can simply make it confusing, while elements that are popular can be rolled out on a wider basis.

Simplicity is often key, and this also applies to development and distribution of data: as O'Reilly remarks, the best way to think of information is via syndication not co-ordination – that is to make information available as quickly and cleanly as possible rather than attempting to control what use of it is made at the other end. Likewise, data of all kinds should be designed to be hacked and remixed into new forms – with 'some rights reserved' becoming a useful contrast to 'all rights reserved'.

The always in beta model for content has had its greatest impact in areas such as rolling news, which in some senses may be said to have predated software development: hourly bulletins offering new information as and when it became available (although, in pre-Web formats, still dictated by a schedule). In regard to this, Paul Bradshaw (2007) has suggested a publishing model that although really devoted to online news has some crossovers to other types of new media publishing. Bradshaw identifies the speed with which online publishing can take place, with an alert allowing an author to produce a speedy response in terms of a blog entry or article identified as an initial response (such as a news story). More than this, however, online publishing can also emphasise depth as stories are returned to and more detailed accounts presented, through multimedia or – and this is where Web 2.0 tools become useful – interactive responses from multiple users.

What is particularly appealing about this model is the fact that it also indicates the different styles of discourse which users and producers (terms that become increasingly blurred in this format) are starting to respond to: we expect a blog to be more informal than, say, a video package, and so may assess and evaluate it accordingly. In addition, the genesis for stories and information does not necessarily begin with the producer but may instead originate with the end user who identifies a particular event.

Distributed applications

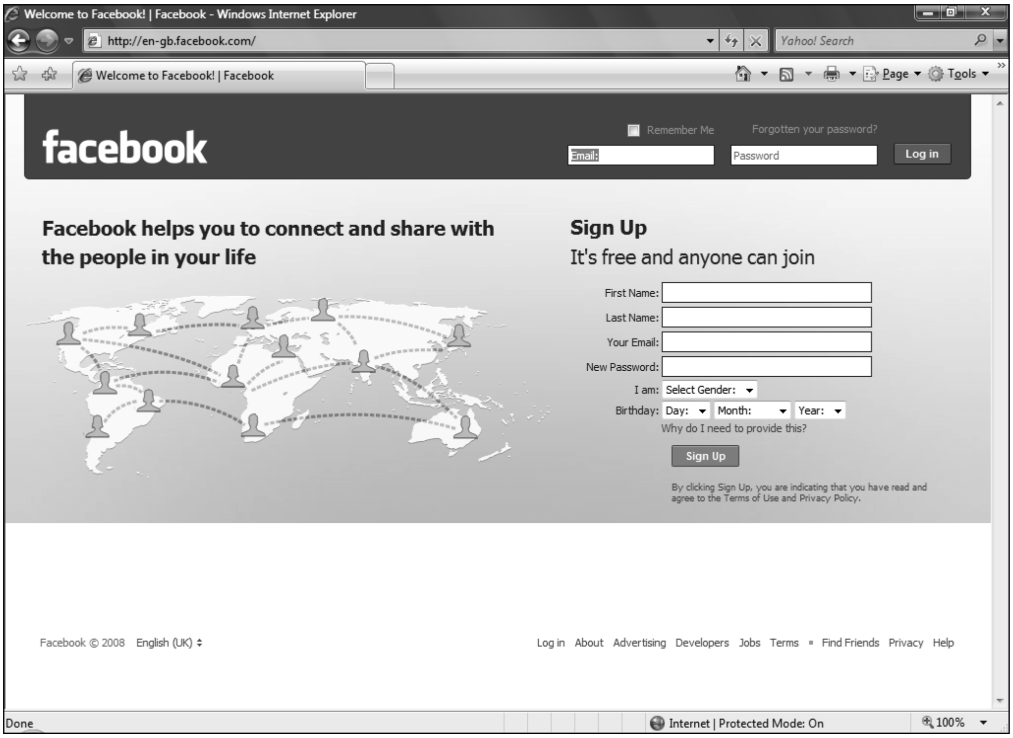
A consequence of the focus on the web as platform is that software increasingly needs to be written above the level of a single device. This is certainly not new to Web 2.0, and was fundamental to Berners-Lee's original idea for data interchange between a number of different computing platforms. However, with the proliferation of devices that can now connect to a network, such as TVs, mobile phones and media players (such as the iPod), the ability to create services that can serve data to multiple end users with different applications, some of them probably not conceived of at the time of design, reinforces the importance of creating open rather than proprietary standards for data exchange. This was the problem for Netscape's webtop model, which ultimately wished to lock users into a vertical market from their browsers up to their servers.

As with content, so the principles of being able to hack and remix data is important to software, the new model for which is to create applications via ‘snap-ins’ or plug-ins, taking source material distributed elsewhere on the web and presenting it in an innovative fashion. Key to this has been the development and distribution of open **APIs**, or application programming interfaces. An API is a standard set of instructions (the interface) to allow different elements to communicate, for example a website and a remote service provided by a third party. An important element of Web 2.0 design has been that certain companies, such as Google, Facebook and Flickr, have made their APIs freely available for non-commercial use so that, for example, web designers can call on Google Maps, friends lists in Facebook or photo albums in Flickr to make them available on their own websites.

The API usually works seamlessly, so a visitor to a site is not even aware that content is being provided from another domain. APIs are usually employed to create **mashups**, web applications that combine data from more than one source to create a single tool, for example by combining data from Google Maps with data from estate agents, thus creating a service that was not imagined by either party. Mashups editors also often make use of RSS feeds to source data from other sources, something that is made possible by the fact that information is increasingly separated from presentation, allowing it to be re-used in novel forms.

As Jesse James Garrett (2005) says, **Ajax** is a good example of this new approach to producing software: ‘Ajax isn’t a technology. It’s really several technologies, each flourishing in its own right, coming together in powerful new ways’. It incorporates **XHTML** and **CSS** for presentation, dynamic interaction and data exchange via

Facebook is one of many sites that has made its API available to third-party developers.


Welcome to Facebook! | Facebook - Windows Internet Explorer

http://en-gb.facebook.com/

Facebook helps you to connect and share with the people in your life

Sign Up

It's free and anyone can join

By clicking Sign Up, you are indicating that you have read and agree to the Terms of Use and Privacy Policy.

Facebook © 2008 English (UK)

XML, **XSLT** and the document object model (**DOM**), and asynchronous data retrieval via **XMLHttpRequest**; and binds everything together with **JavaScript** (all these are technologies that will be explained in the next chapter). What appears to be one piece of software is in fact a hybrid of many other forms that can be customised and re-adapted much more quickly.

WEB PRODUCTION SKILLS

The skills required to produce a website can be incredibly varied. A lot of talk in media industries in recent years has focused on the notion of convergence, that different production techniques as well as the means for consuming media will come together in a single platform. The internet and computing are obviously the most important technological driving forces behind this phenomenon, although economic, social, cultural and even political factors have an important role to play.

The complexity of setting up a site should not be overemphasised. If you need a quick and easy route into publishing your ideas and thoughts, then a blog can be as easy as using a word processor – with the ability to embed multimedia instantly. However, for a complex, multi-user content management system a raft of programming, design and content production skills can be called into play, nearly all of which will be covered to a lesser or greater extent in this book.

A knowledge of core technologies such as HTML and CSS will enable you to modify and customise many of the free services that are provided to users online.

How to plan for a website

Before creating anything technical, a web producer needs to think ahead with regard to planning for their site. What content will it contain? Who will use it? Will you need multimedia? How will it be tested and maintained?

As part of the pre-production process covered in the next chapter, we will outline some of the ways in which you can prepare for your website in order to manage it effectively from the initial idea to final testing and deployment. Project management can be a slightly daunting term for small-scale developments, but having clear ideas and objectives about what you wish to achieve will help you when it comes to the design stage.

For the first stage of planning, it is important to know what you want –perhaps a blog for fast and easy publishing or a Facebook group if you want interaction from lots of users. In some cases, designing a website from scratch may be the least efficient way of reaching your target audience.

Knowledge of core technologies

Even if you are planning to take advantage of the many Web 2.0 publishing options available online, an understanding of at least some of the core technologies will be immensely useful in customising what is on offer. While there are, for example, plenty of templates available on sites such as Blogger, knowing how to modify that template's HTML structure and its appearance via CSS (cascading style sheets) will provide you much more scope in getting your message across. One of the ironies of Web 2.0, noted in the preface, is that the simplicity of publishing online has made a knowledge of underlying code much more important in many ways than learning a complex web design application. Throughout this book we will return repeatedly to the essentials of code needed to customise as well as build a site.

The most important core skills are a knowledge of HTML (or, rather, XHTML, rewritten in accordance with XML principles) and CSS, but you will also be introduced to client- and server-side **scripting**, in the form of JavaScript and PHP, as well as XML, the database programming language **SQL** and some other new technologies such as Ajax that draw together these languages. In some cases, what is offered is a taster of such languages rather than a comprehensive overview, but just about anyone should be able to modify an existing platform or begin to create dynamic sites from scratch.

Web design skills

An understanding of the underlying code is immensely important to being able to create and fix websites. Indeed, one of the key skills that is required for design is the ability to identify and solve problems, debugging your site when things go wrong and do not work quite as you intended.

Knowing how to code, however, is not the same as being able to create an efficient design. How will users navigate around your site and find the information they need? What is the structure of the site and how do pages relate to each other? What will those individual pages look like, and where will content go? Paying attention to such things, as well as knowing how to work with colour, typefaces and design layout, will make a huge difference to how your site is received.

In addition, while the emphasis in this book lies with knowing core web coding languages (at least enough to customise and modify a pre-existing template), there will be plenty of examples of using a visual web editor such as Dreamweaver to create your own site. Such editors should never be used as a simple replacement for any knowledge of HTML in particular, but can be a very handy supplement for creating your own sites.

Multimedia skills

One of the effects of convergence is that the web is becoming the ideal platform for publishing a wide range of media. Although the very first web pages were purely text, within a couple of years the inclusion of images had transformed the presentation of online documents, enabling them to use magazine-style layouts. For most of the 1990s, this was the standard form for web design, music (in the form of **MP3** especially) being added to the mix by the end of the decade. Sound was available to designers prior to this, but bandwidth did not really support high-quality audio and the less said about the horrors of embedded midi files in pages in the mid-1990s the better. It was not really until about 2004–5 that broadband access to the web became widely enough available for video to be added to the mix, but since then visuals have become an important part of any web producer's repertoire.

One of the consequences of this is the impact that has been felt in a number of established disciplines and professions, such as journalism. Where once it was enough for a journalist to master a pen (followed by the typewriter and word processor), increasingly he or she will also need to be familiar with at least basic skills in handling a camera, audio recorder and video camera to succeed. It should be noted, however, that there is often a considerable difference between the quality and skills of audio and video work required for many sites compared to high-definition radio and television broadcasts. This is not an excuse to be shoddy, but in practice people listen to or watch short clips online when embedded as part of a website (video on-demand and online radio being two exceptions of course) in contrast to the ways in which they consume more traditional media.

News values and writing skills

The reference to journalism draws attention to the final vital skill for an effective website: video, audio and interactive technologies such as Flash may have made the web a much more exciting place, but the fact remains that a great deal of what we do online is read copy – from captions and comments to full-blown articles. Many professional websites can be let down by spelling mistakes and solecisms – or simply by containing content that is far too dull.

The ability to spot a story and craft copy is an essential skill for the producer of a successful site, one which should never be underestimated. In many respects, the traditional skills for writing news – compressing as much information as possible into the first paragraph, ordering it by relevance and using the inverted pyramid – are eminently suitable for the web. Readers typically scan only the first few lines of an article before deciding to continue or move on, and so being able to convey what a page is about immediately and vividly is extremely important.

What this all means is that the ideal candidate for a web producer is an impossible figure: one who knows how to code, can handle multimedia equipment, has an eye for design and can also craft the perfect story. In practice, for all the talk of convergence and multi-skilling, large professional sites still rely on some division of labour where writers write, photographers take pictures, broadcasters produce audio-visual materials and web designers code and handle layout. And yet the ability to work across these different areas can enhance and improve the core area that you, as an individual, decide to focus on. If you are a writer, a good understanding of web design will make you appreciate why certain types of copy work better online than in print, while good literacy will improve the professionalism of your designs.

The ability to spot and craft good news copy is in many ways more important than ever as readers scan only the first few lines or words of a story before deciding whether to read on.

CHAPTER 2

Pre-production

In this chapter, we will look in more detail at the decisions that need to be made before you begin to construct a website, including selecting the right platform for your content, whether it's a personal website or a multi-user content management system (CMS). We also outline the relevant technologies that go into making the web producer's 'online toolkit', and give advice on selecting and setting up a web server.

PLANNING A WEBSITE

The temptation when creating a website is to jump straight in, but if a site is to be a success some planning ahead will make a significant difference. Resources must be allocated, deadlines observed and tasks established. What this consists of will obviously vary from project to project: if you are a student engaged in a college project, there will be specific outcomes that you have to achieve, of course, while it may be that you are setting up a site that will be used by multiple content managers as part of a business or company project.

Managing a project

Anyone involved in project management will be presented with a job for which they have limited workers, time and resources. Even if this is a sole project, for example to be submitted as college work, you must plan ahead to determine what you will need to do in a particular timescale. The first step in managing more effectively is to divide a complex project into essential tasks that can be assigned deadlines and set in order.

In addition to ensuring that time and workers are allocated to meet a deadline, effective management should consider the consequences on deadlines if budgetary constraints are applied, for example how much work can be achieved if a certain amount of money is cut or moved elsewhere, or which tasks will have to be prioritised if deadlines are changed.

To make the process easier, a project can be divided into four distinct sections: defining the project, creating a project plan, tracking the project and then closing it. In this part of the chapter, we are most concerned with the information that feeds into the first two areas. A project plan that maps out tasks and deadlines can be an indispensable tool for defining clearly the scope and resources available. The first step is to ensure that these are assessed realistically, ensuring that assumptions can be met. To help with this, a project plan breaks down the project into tasks that can be assigned different resources and workers, having identified who or what will fulfil each task.

A project plan can proceed by one of two ways: you can enter a start date and schedule the plan forward to determine the best deadline, or enter the completion date and schedule the