Planning and Structuring Content

Sometimes, people and businesses get lucky on the Web. They create a Web presence that garners exposure and a high volume of traffic. Their Web presence then goes beyond log files or hit meters to connect with people. A site becomes a destination when it connects others with personal stories—a respectable place for instant social commentary or the most respected brand for online shopping for everything under the sun and then some.

However, without proper planning, any site that does become successful will not be able to leverage that kind of luck for long. In fact, you are more likely to achieve the success you want online through hard work, planning, and structuring of your message than a lottery ticket of a Web site. To ensure that your Web site meets its goals to become a success, it's important to strategize as to the rationale behind its existence.

In this chapter, we will look at factors that will help you reach your intended audience with your message before you put pen to paper for sketching out designs or code markup between a body element. Afterward, we will look into how to get your content ready for Web delivery and the application of design, the presentation of your message, through Cascading Style Sheets (CSS).

Know Your Audience

Design consists of more than visuals, graphics image formats, sound files, and typography. Design requires planning, and one rule binds print designers, architects, movie directors, car manufacturers, politicians, and Web designers. That rule is: "Know your audience." Because one kind of person doesn't make up the millions of Web surfers, not one manner of Web site design can reach this audience.

When a client hires you to build a Web site, you should solidify the goals the client has for their site. Determine the site's intended functionality—the client's needs against his wants—so that you can examine your client's competitors. Examining competitor sites in this context is an expected exercise of design strategy. Exploring how a client's competition reaches its audience is a good way of determining the status quo for doing business in your client's industry. However, if you apply a similar design to a client's site as a competitor, you are shortchanging the client. Doing so means that you are failing to differentiate the client's brand and to attract the competition's customers. In addition, you might just be adding to the stockpile of bland Web sites. In essence, you are wasting money and time instead of approaching the Web site's design from the audience's point of view.

People make judgments in the way they communicate with others, from wrongful discrimination to giving genuine courtesy. Their perceptions of whom they are talking to dictate interactions, such as conversations (or the lack of conversations) in every part of social life. First impressions are important in every occasion from job interviews to blind dates. The same can be said for how you craft the message for your Web site. If the design does not reach out and inform at the start, it's not effectively doing its job. To quote Jan V. White's *Color for Impact: How Color Can Get Your Message Across or Get in the Way* (Strathmoor Press, 1997), "'First-glance value' is not just a catch phrase. It is the very kernel of functional communication, given today's frantic competition for attention… Content and form are one. Design is a lubricant for ideas."

Print designers, architects, film directors, and Web designers, for example, work through a design problem like a consummate negotiator who is trying to create paths for understanding the material. By negotiating a compromise between art, function, and experience, designers and developers work on the visuals, content, and backend portions of a Web site, which make up the inherent experience of surfing the Web site. (This isn't your father's graphic design job.) The kinds of visuals and tools that designers use to successfully bring that experience to users depend on with whom they are going to communicate on behalf of their clients.
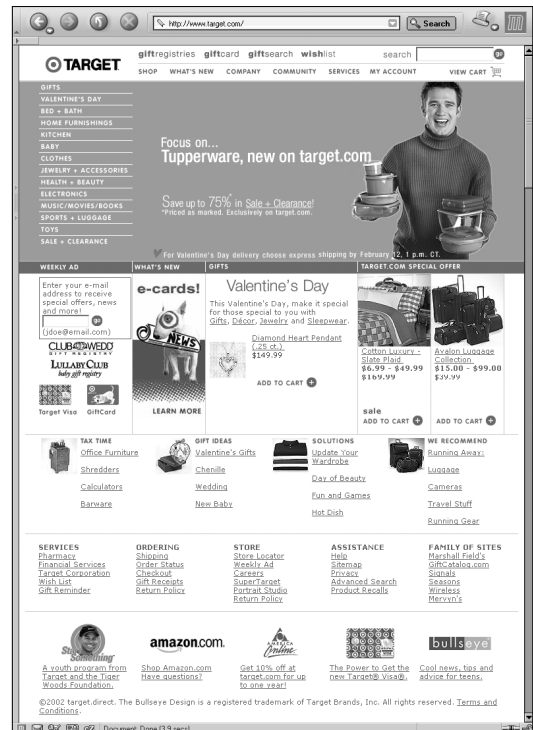
Customer Types: Tomāto or Tomäto

Even when an audience shares similarities for two competitors in business, their Web site designs and development produce two different types of sites. For example, both Barneys New York and Target share an audience that is interested in purchasing consumer goods (see Figures 1.1 and 1.2).

Figure 1.1

To view Barneys' e-commerce site, you need a high-end computer and a fast Internet connection to handle the presentation.

Figure 1.2

Target provides a Web site that almost all Web-enabled computers that have a current generation browser can reach.

Both Target and Barneys New York have physical "brick and mortar" stores that sell their products to customers. What separates the businesses is the type of customer each is targeting.

Target aims for the bargain shopper customer type, or the sensible, middle-class society. It has a wide selection of everyday and every-so-often goods (such as snack items, drinks, contact solution, and so on, as well as stereo equipment, fake Christmas trees, Halloween costumes, and the like). Barneys of New York targets and markets to the richer segment of society, or the full ticket price customer.

These differences in types of audiences are easily identified in the companies' respective Web sites. Barneys New York has few HTML-based forms on the site, but it almost exclusively uses a multimedia experience with Macromedia's vector-based tool Flash. To get the full experience of Barneys's Web site, users need to have fast Internet connections to handle the large file sizes, and they need a high-end computer system to render the Flash presentation. Target's site, on the other hand, has a simple, straightforward design that targets the majority of American Web surfers. On Target's site, people can shop for reasonably priced goods without interruption on a fairly low-powered computer with one of many available browsers.

Informing the world unlike its print counterpart, which informs only the United States. The aim of this site is to inform its readers about the latest in national and international news.

Reader Types Versus Customer Types

Knowing your audience isn't just based on customers, regardless of what the talking heads on financial networks and television shows say about new media. Believe it or not, people surf the Web for information. (There's a reason that the Internet is referred to as the information superhighway.) These people are reader types. They can check out *USA Today*'s Web site for coverage of American-centric interests (see Figure 1.3) and compare the news with what's in their local newspaper's Web site as well as other national or even international news sites. Although the online publication of *USA Today* reflects the content of its print counterpart, the site also supports online-only content and publishing updates that occur five days a week.

Web sites that reach out to customer types need to provide a different type of atmosphere from those that cater to readers. With shopping sites, they need to provide a sense of trust with your credit card information and timely fulfillment of your order. Sites that cater to reader types are concerned about meeting publishing deadlines on a regular schedule to retain and grow their readership. Having something original or interesting to say is also a good idea.

The American right to freedom of the press does not belong only to people who can buy a printing press. Justin Hall's Links site, an independent online publication chronicling his life in his own words, images, and programming, is an example of a one-man publishing empire (see Figure 1.4). Established in 1994, Justin covers the convergence of new media and society, and his audience is anyone interested in things that matter to him and those that get caught up in the journal of an expressive person. Justin's publishing schedule is his own. (Usually, it's a once-a-day writing schedule.)
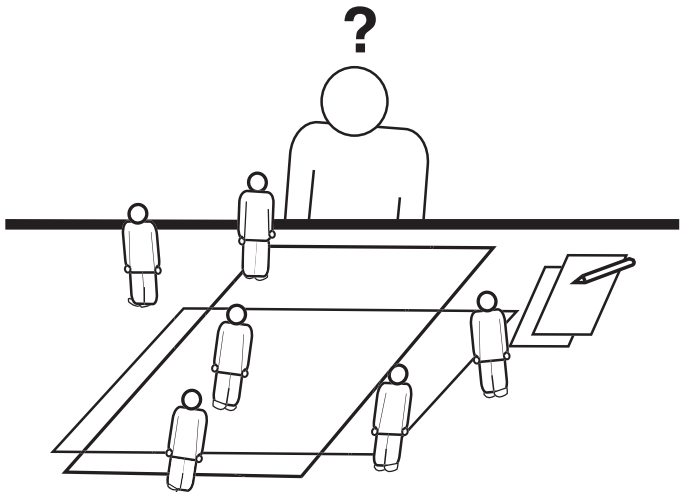
Figure 1.4

Justin Hall's personal site is an easy-to-read personal journal of one man's perspective through technology and its effect on culture. Yes, it has ad banners, too.

Experience Types

Surfing the Web is an experience unto itself. Users can engage in online artwork that only exist on the Web. And although they can burn art pieces onto CD-ROMs, the delivery of the artwork is the Web. Instead of going to a museum to experience artwork in person, the art is available wherever a person can go online to find it. Web artist Richard Grillotti's PixelJam takes full use out of a browser's capability to stretch images and the animation specifications in GIFs (see Figure 1.5).

Museums, on the other hand, are leveraging the power of the Web to advertise their shows (see Figure 1.6). Although the museums might have shows for their electronic artwork, the user has to physically go to the museum to click a mouse, tap keyboards, or use other input devices that the artist has created to interact with the artwork.

However, both SFMOMA and PixelJam are out to create a unique surfing experience for their visitors. Grillotti's site is the experience, whereas SFMOMA's is an attractive marketing ploy that tries to lure paying visitors to their museum (and stop in the gift shop on their way out).

Figure 1.5

Richard Grillotti's PixelJam uses the nature of GIF animations that are small and then stretches them like a canvas over a browser frame. In short, it's Web art.

Figure 1.6

The San Francisco Museum of Modern Art (SFMOMA) tries to convince visitors to check out their "brick and mortar" art museum.

Through audience types, we realize that various types of sites are available. For your site development, you need to realize up front if you're building for one, two, or all three of these site types for your audience. It's not just a matter of knowing the audience, but building the right type of site for your audience.

Liberation Through Audience Limitations

If you don't know who you are talking to, will your audience know what you're saying through your Web site? Probably not. Your audience will probably be as lost and confused on the site when they reach your home page (see Figure 1.7) as you are. Without thinking about the audience for your Web site, the people who do show up to see your Web site won't be catered to effectively like they would be at a site that was built from the ground up to help them.
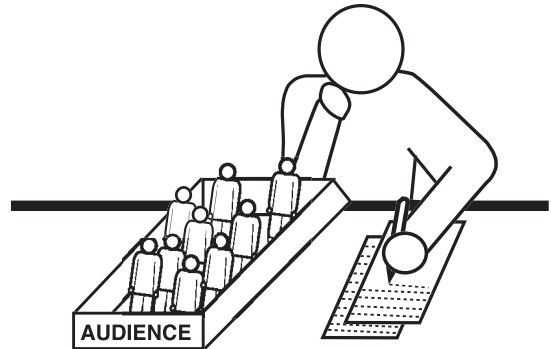

Figure 1.7

Starting a Web project without defining your audience leads to a Web site without a beneficial purpose.

One of the great things about taking an audience-oriented design approach is that you are actually reducing design possibilities, or the options that would be fruitless. You don't take unneeded steps. Instead, through research and strategic thinking, you can determine what the best ways are of *not* wasting time, money, and other resources at your disposal (see Figure 1.8).

Figure 1.8

By boxing your audience, you can put them under observation and testing to determine their likes, dislikes, and what they want from a site like the one you are going to build.

AUDIENCE

It is also important for you to know who *isn't* part of your audience. Sometimes, the best way to determine the best solution is to eliminate what obviously *won't* work for your Web site. Eliminating different groups of audiences allows you to narrow in on your audience (see Figure 1.9).

Figure 1.9

Getting rid of audiences that don't serve your mission brings you a step closer to bringing your message to the right audience.

TRASH

With an audience-centered approach, you won't spend time trying to reach millions of Web surfers. Instead, you'll be able to focus on the people who matter to your project being a success. Just because a million people could potentially visit your Web site, the likelihood of your ISP's servers melting down from too many users won't happen without careful planning.

Of course, serving a specific audience at the beginning of the project does not imply that you will be serving the same audience at the end of the project or during the maintenance and content updates. During the development phases of a Web site, a client or a project manager will add his opinions about how to improve the Web site. Take these opinions with a grain of salt. These opinions are coming from one person's perspective of Web surfing and what he thinks is "cool." By adding only this client's/manager's philosophy of approaching the functionality and design, chances are that you are cheating the audience instead of making the site a quality environment.

Of course, blindly ignoring the person who is paying for the Web site (and therefore you) isn't a good thing either. If possible, you need to reach a consensus about the best way to approach the project. Remember: You've researched the best way to build the client's Web site so far, and you can back up your claims. Ask the client to do the same for any change or modification that doesn't seem to help. See what the client has to say and reach compromises.

Gathering Content

Web builders should always determine the who, what, when, where, and why before writing a sentence of copy or applying the most basic of Photoshop's filters to an image. By approaching the project as set of questions that need to be answered, the final product will be better. You'll have fewer questions about what the site should be and should do at the end of this kind of Q&A. You will have a road map to getting the right content before you design. However, the content, the essential part of the Web site, has a strong chance of being mishandled at the expense of trying to land and turn around business.

If you build a Web site for yourself, either as an independent publication like an e-zine or homage to the *Superfriends* television show, the burden of filling up Web pages with content falls to you. You manage the content flow and publication schedule. However, when a client hands over his time and money to you for a Web site, he wants his intended audience to use his Web site and he wants his super-heroic message conveyed.

Where Things Go Wrong

Typically, Web design studios are still vying for a project when the design of Web site is determined. This usually happens when a client is presented an image of what his new Web site will look like. The image is called a *comp*, which is a concept artwork depicting what they see a client's site could potentially look like if they hired the designer or designers. Regardless of what this artwork turns out to be, the comp is still an artist's rendering, a weather forecast that always shows a beautiful day.

The goal from the developers' perspective is to get the client's business, and if providing eye candy will get the job, so be it. The effectiveness of

the site depends on how well the designer or designers who are working on the initial artwork are at forecasting the content requirements for the project.

This method of creating Web sites could be satisfactory for sites that are destined to stimulate art pieces at the whim and desecration of the designer. Firms, boutiques, or freelancers who build the sites are usually paid to fulfill real business objectives and reach intended customers.

I am not implying that clients just grab the Web designer with the best knack for keen eye candy. A client typically will want to ensure that the project can be handled correctly. Most clients will approach their Web site with a clear head and have a list that is a mixture of wants and needs for their business to succeed online. However, if a client is presented a comp, he forms preconceived notions of how the site will look and be structured.

Instead of the audience being the approach to the end product, the Web site shown in the artistic rendering becomes the approach. If a client chooses a design presented to him (and likes other aspects of the job proposal), the Web site studio wins a check and signed contract. The work of "strategic development" begins then.

The tasks in strategy development include competitive analysis, content development, strengths/weaknesses assessments, functionality reports, and maybe even target demographics. You should address these missions to answer key questions at the start of a Web project so that you can have a pure understanding of where the design of the site should go.

Additional Resources

For more reading about the process of Web development and client relationships, I recommend these books:

- *Collaborative Web Development: Strategies and Best Practices for Web Teams* by Jessica R. Burdman. (Addison-Wesley, 1999) ISBN: 0201433311.

- *Secrets of Successful Web Sites: Project Management on the World Wide Web* by David Siegel. (Hayden, 1997) ISBN: 1568303823.

- *Web Redesign: Workflow That Works* by Kelly Goto and Emily Cotler. (New Riders Publishing, 2001) ISBN: 0735710627.

In the place of an appropriate site is an appropriate mess. The once beautiful site that the client signed off on (see Figure 1.10) now will have been produced and then diluted to fit the content and functionality. In time, the site will grow into a monster of links, badges, and animations that will override any key messages that the client wants to convey (see Figure 1.11).

Figure 1.10

In this redesign for Datamax Corporation, the layout is clean and simple.

Figure 1.11

A few months later, the home page for Datamax Corporation is awash in links to various subsections and graphics elements that conflict with each other for the user's attention.

These types of sites are often hard to navigate, long to load, and lacking in key content. The content that *is* available will miss the intended target audience. As if an underperforming site is not enough, the client then spends even more money and time to the same designers to "fix" the problem with a new redesign. Thus, the cycle begins anew with a new comp to land the redesign.

Additional Information

For more information on the subject of evolving tools, see Stewart Brand's book, *How Buildings Learn: What Happens After They're Built* (Penguin USA, 1995). The books tells you how buildings adapt to the owners of a home or an office.

To have a successful Web site, your visitors need to feel that they "own" the site. You can do this by reaching out to them by continually asking for ways to improve the site or asking them to participate in discussions on the site. You can draw a parallel to people who move into a brand new house. The architect might have an idea in mind for how this house would be best used: an environment for a mother, father, and their 2.3 kids for the rest of their lives. However, what happens to the house when a single parent who has a child of her own and several foster kids moves in? The house changes to the new needs and wants of that family. The house changes on the inside and outside, but the framework stays the same. For example, rooms that were designed to entertain houseguests are transformed into ad hoc bedrooms, and the backyard garden turns into a mini football field.

You can avoid having a house that changes often by making sure you are building correctly for your audience. Of course, it's not the end of the proverbial world if you realize that your Web site isn't making the most of its resources. Simply think of yourself as a live-in architect: re-examine your message and ask your audience for input on how to make the house better.

Getting the Content Right: I've Learned This Lesson Before, But Never Like This

Then there's the pesky task of acquiring the content. If you have worked in the industry, you know how hard it is to get content from the client. Working on content with a client is one of the hardest things to do. If

content management does not share a mention in your proposals to client, make the importance of crafting the content to fit the audience a focus from now on in your pitches to clients.

Several metaphors emphasize the point that structuring is extremely important. Here's a list of some of them I've used or have heard others use:

- **You wouldn't build a house without blueprints. You plan the size, shape, rooms, utilities, and materials and get the best contractors** *before* **you start construction.**
- **You wouldn't buy a car your crazy Uncle had "just thrown together one day." You want to buy a car that has been planned, assembled, and tested.**
- **You wouldn't run for a high-level political office without making sure that you have enough money to campaign.**
- **You wouldn't just show up in Orlando for vacation without having decided where to go for lodging and outings.**

Now here's one more:

- **Designing a Web site without predetermined content is like making a Twinkie with the wrong filling.**

After several years in Web design and development, I've seen lots of white in customers' and project managers' eyeballs who roll their eyes when I'm evangelizing about the importance of content. Somehow, bringing up points using Ford cars, blueprints, and hypothetical vacations to Orlando bores people to tears. Or maybe it's the thought that producing copy is less attractive of a proposition than judging a beauty contest of potential Web site designs.

Regardless of the reason, it is vitally important to iterate the importance of content before the designing and building phases, and I do that by using the Twinkie metaphor. Because a Twinkie is small, inexpensive, and after several hours of troubleshooting Web production errors into the midnight hours, even downright desirable, the lesson tends to goes down faster with the Twinkie metaphor than with auto parts. And just like a sweet, high-fat Twinkie, I hope the point I'm trying to convey goes down easy, but sticks with you long after you sample it.
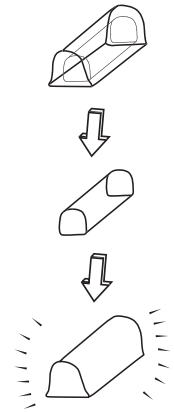
The Mistake of Design Before Content

If you look underneath a Twinkie, you will see three spots of cream filling. Hostess has a mechanical process of inserting cream filling into the cake shell. The outer cake shell is made first, and then the cream is filled into the Twinkie (see Figure 1.12).

The cake shell is like the comp that the designer creates for the client. By selling the visual first, the designer hopes to get the creamy content from the client to put into the Twinkie. The problem is that the content rarely fits into this pre-made shell. The designer might not realize that the client's filling is made of artichokes and sun-dried tomatoes. For a Web design firm to be good at selling Twinkies as Web sites, the firm needs to automate the process. Although a developed site will be available for the client and visitors to see, the real success from the site will come as a surprise rather than any serious planning and calculated risks.

Let's say we were to take Twinkie sites as a way of doing business. To make a substantial profit with "Twinkie sites," you would need to hone your craft into an automated assembling line. Just like Hostess does not make one Twinkie at a time, you shouldn't be going after one client at a time. Hostess makes several hundred Twinkies at once—they're not individually and lovingly hand-crafted.

In the Web development world, such businesses already exist. For example, VerisignSites, which VeriSign owns, allows customers to buy a domain name, pick a Web site template, and edit the content in their browser (see Figure 1.13). As another example, at Yahoo!'s GeoCities site, people can build Web sites if they don't need a vanity Web address (see Figure 1.14).

Figure 1.12

In a perfect world, you could build the cake shell and then insert the right amount of cream filling to make the perfect Twinkie.

Figure 1.13

VerisignSites is a one-stop place for buying a domain name, selecting a design, and filling it with content. That's good news if you find a site that matches your content perfectly—you'll have a successful Twinkie site!
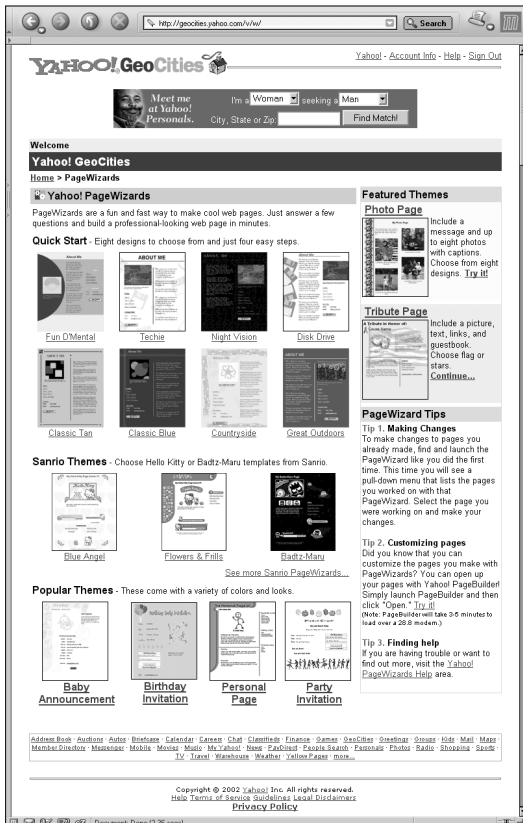
The GeoCities site offers free Web site designs. It even has a page design for birthday invitations.

In these situations, clients are stuck with the Twinkie shell, regardless of their recommended diet. And because you're filling your content into a prebuilt Twinkie cake shell, your content *must conform* to the inside of the shell, not the other way around. If you force the content into the preformed shell, you will get weird Twinkies (see Figure 1.15).
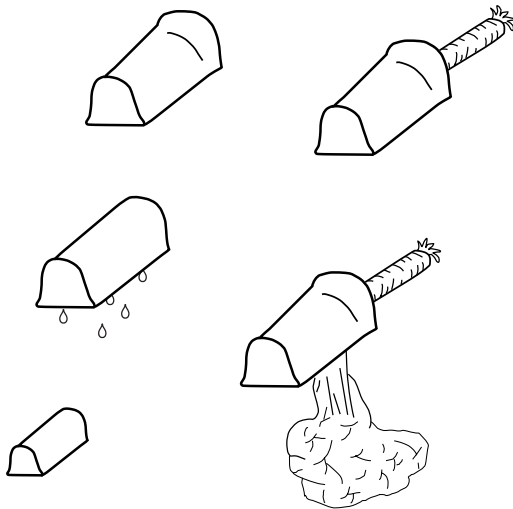

Figure 1.15

What happens when you have a Twinkie cake shell, but you really want to talk about carrots and broccoli? Your site becomes a Frankenstein—a sugar covering with a healthy center.

Ensure Proper Communication

Because most companies and clients do not have the resources to hire a copywriter full time—much less a freelance job—the act of writing will come from someone in the company who occasionally writes better than anyone else. When you finally receive the content from the client, you likely will need to rewrite or at the least revise some part of it.

You will need to go through the copy as if you were a site visitor. You need fresh eyes, as if you have never heard of this product or service. Although the Vice President of Product Development will no doubt understand his product inside and out, he probably won't be able to educate someone about the product who hasn't heard of it.

Guide, Don't Tell

Most companies who are attempting to stretch their marketing dollar will repurpose marketing copy from print collateral (print pieces such as business cards, stationary, envelopes, and so on) that carries the company's image by publishing it directly to the Web without revisions. This leads to an inability to capitalize on the inherent strengths of the Web medium: the linking of documents at related junctions, which is the nature of hypertext. At the very least, content that is destined for online distribution should take full advantage of what the Web has to offer.

When reviewing content that you have written or obtained from your clients, make sure to look at it from a descriptive angle, leading the user to the different sections of the site. Web sites need what is termed "guide copy," allowing for access and understanding of what is contained within the site and leading users deeper into the site experience.

A Web site's written content can and should be more than simple marketing fluff. Using the content to guide the site visitor toward the intended destination is an important element of any process.

Structured Markup Is Why We Are Here in the First Place

Yes, believe it or not, the Internet wasn't designed to make Amazon.com a household name. Rather, the Internet was built primarily to share scientific papers with scientists. The Internet was initially established in 1962 by the Defense Advanced Research Projects Agency (DARPA) (see `http://www.arpa.mil/`), and they held a public demonstration in 1972. Tim Burners-Lee announced the Web as an academic exercise in 1990 at Conseil Européene pour la Recherche Nucléaire (CERN), the high-energy particle physics laboratory in Geneva, Switzerland.

The building block of the Web, Hypertext Markup Language (HTML), was simple enough for the focused and scatterbrained individuals to encode their work in simple tags. When a headline was marked with the `<h1></h1>` container tags, it flowed through on its own line with the text in bold and larger than text marked in `p`. A headline that was wrapped in `h2` usually would appear to be larger than a headline text that was wrapped in `h6`. Also, if a passage of text was quoted from another source, then the `block-quote` tag could be used to create a passage of text that was indented.

With such a low barrier to publishing material, more than scientists got in on the action. People from all walks of life soon joined the online world.

HTML Terrorists

When the precursor to Netscape's Navigator, NCSA Mosaic (see `http://archive.ncsa.uiuc.edu/SDG/Software/Mosaic/`), came out with its HTML extensions, such as the `font` tag, the `blink` tag, and the ability to change background colors, Web builders started using them. In doing so, Web designers were slowly becoming HTML terrorists while they were thinking they were simply doing their job.

David Siegel's *Creating Killer Web Site*s (Hayden Books, 1997) was the first book that approached Web design as a serious issue. That's when the abuse on structured markup became common practice. If you told designers in the late 1990s that they were hurting instead of building a strong Web, they acted puzzled and asked if you had seen their portfolio lately.

Designers took appearance as a paramount issue (which is their job) in making Web sites successful. Following are some typical Web designer thoughts of the time:

- **"Who cares if we indented text with `blockquote`? Of course people will know I'm not quoting the entire Web page. That's downright silly. Who thinks that?"**
- **"Tables are to be used for numerical information? What does that mean? If I couldn't use tables for placing my images and text, the browser wouldn't let me do it, right?"**
- **"I need my designs to be pixel perfect, just like they are in Photoshop. I used to use single pixel GIFs as spacers, but now I use Netscape's `spacer` tag for that."**

I won't even mention what happened when Web builders discovered when Netscape began to support the animation part of the GIF specification.

However, to be fair to the designers (especially because I was once an HTML terrorist, too), sometimes the designs called for visual presentation. Web sites for Hollywood movies that dealt with images and were time sensitive, for example, could not achieve marginal success if they didn't break the rules of structured markup.

Blame the Browsers

The browsers in the late 1990s were poor in their support for Web standards. Both Netscape and Microsoft, the companies that own the major portion of the browser market share through their respective browsers, were big companies. And big companies are used to getting their way. Why would they want to stop making their own extensions to the Web?

Well, thanks in part to a group of developers and designers who make up the Web Standards Project (WaSP), the builders of the Web started demanding that sites perform to the recommendations made by the World Wide Web Consortium (W3C), which was founded by Tim Berners-Lee. Although it was okay for browser vendors to promote their exten-

sions to HTML, WaSP members argued, they should adopt the standards that the W3C set forth. Slowly, the major browsers came to use the angle of standards support in future releases as marketing material. This was the start of the change in how Web design was accomplished, but something terrible happened.

The initial browsers that implemented CSS did so poorly. What killed designers' and developers' early adoption of CSS is that the initial browsers that supported CSS did not work well enough to have meaningful use. The marginal benefit from using CSS that didn't affect rendering was mostly selecting margins for the entire body page and requesting fonts. We wound up getting ways to get rid of the `leftmargin`, `topmargin`, `marginwidth`, marginheight, and `face` attributes for the `font` tag.

If you want to incorporate CSS into your Web pages and be conscious of how your designs appear in as many browsers as possible, you can include only a few CSS rules. Listing 1.1 is an example of a "Web-safe" snippet of CSS code that will work in almost any browser in which support for CSS is vague, to say at best.

This snippet is given as a sign of the weak implementations in browsers. With the current generation of browsers taking over the share of the browser marker, we now are able to make use of CSS capabilities more.

Listing 1.1   Web-Safe CSS

```
body  {
   /* Gets rid of margins in space around page's content */
   margin: 0;
   /* Selects font for the page */
   font-family: Verdana, Arial, Helvetica, sans-serif;
   /*Selects color for body text */
   color: #333;
   /* Selects the background color of the page. */
   background-color: #CF0;
   }
p, td  {
   font-family: Verdana, Arial, Helvetica, sans-serif;
   }
```

CSS: Terrorist Rehabilitation

Because the browsers took an early adoption to CSS, designers have plunged to use the power of this technology in Web design. That's a good thing. Due to the shoddy implementations of CSS, practical use was years away until today. Now we can start shedding the old habits thanks to standards-based browsers from the major browser vendors, but you will need to know a couple of things:

1. **Forget how you used to develop Web pages in the traditional "HTML Terrorist" fashion.**

   Just so there isn't confusion, this is what you will be giving up for contemporary Web design:
   - **Single pixel GIFs for positioning**
   - **HTML tables for page layouts**
   - **Non-breaking spaces for leading, propping table cells**
   - **The ability to put important text into images**
   - **Framesets**

2. **If you've built for the Web before through traditional "HTML terrorist" methods, you will need to reinvestigate HTML.**

3. **Your content shall be clean of excessive markup that's purely for visual presentation, such as slicing images and placing them into an HTML table.**

4. **Take the time to learn CSS. Buying this book is a good start; however, you must also try reading the specification that the W3C set (see** `http://www.w3.org/Style/CSS/`**) or Eric Meyer's** *Cascading Style Sheets 2.0 Programmers Reference* **(McGraw-Hill Professional Publishing, 2001).**

5. **Validate your markup against the spec during the development of your Web site. Valid code will allow your content to be compatible with browsers that follow standards and for browsers that haven't been released yet.**
   - **For CSS validation, see** `http://jigsaw.w3.org/css-validator/`**.**
   - **For HTML or XHTML markup, see** `http://validator.w3.org/`**.**

HTML, the markup language that was used to make most of the Web pages in the 1990s, is actually an implementation of SGML. To the average Web designer, SGML is not that all that interesting. Designers should know that SGML depends on having documents be very structured. The root in HTML is structured documents. Just because you can code malformed documents doesn't mean it's the right way.

Why bother with structured content? Well, forward compatibility is one reason. *Forward compatibility is* the ability to mark up your content in a manner today so that it will ensure proper display in future browsers.

If you use W3C recommended markup, *in theory* you won't need to fear browser vendors' extensions. As Web builders, we generally hope that the W3C and browser manufacturers keep backward compatibility as a primary goal as they work on refining and creating Web technologies. It's not a perfect situation to be in, but your Web sites will be in a better predicament for following W3C recommendations than not.

By following W3C recommendations, you will save time and money in future redesign costs. If properly set up, revised CSS will automagically update your site's appearances faster than a competent text editor can search and replace.

Also, remember the Twinkie sites? With preformed cake shells that are common in such sites, you run into another problem. Often, you will need to place "body copy" into text-area form fields (see Figure 1.16). The inherent context of marking up the content in HTML is lost when you're using systems like these. The integrity of the site is at the hands of the people who might not know how best to maintain the site.

Figure 1.16

Yahoo!'s GeoCities page builder asks for content through a Web form.

In addition, by producing your content in valid HTML or XHTML and applying the style through CSS, your pages will be more accessible to people who have disabilities. If you run a Web site for an American government and an American-funded agency, it's more than a sensible approach to your Web site project to follow the Americans with Disabilities Act and its Section 508. It's the law in the United States.

Content that is trapped in HTML that has been marked up in traditional Web design methods will not be able to be leveraged down the road when changes in browsers or Web technologies occur. In theory, structured content that takes advantage of CSS will be able to take advantage of these changes, like future versions XHTML or CSS allowing for greater control of the page layout and style. By using Web standards, you have a way to update the content more easily than if you have used single-pixel GIFs, GIFs for text, or numerous nested tables.

Content Structuring Exercise

Notice in Listing 1.2 that I've declared the document type through the DOCTYPE tag in the first two lines of markup. A valid, well-structured document will declare what version of HTML the document type definition (DTD) is using.

Listing 1.2   Sample of Properly Marked Up Content

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Web Design Pad - a better mousepad by design for Web designers,
    developers, and builders</title>
  </head>
  <body>
    <h1>Web Design Pad</h1>
    <p>A better mousepad by design for Web designers, developers, and
    builders</p>

    <ul>
      <li><a href="benefits.html">Benefits</a></li>
      <li><a href="faq.html">Frequently Asked Questions</a></li>
      <li><a href="press.html">Press and Testimonials</a></li>
      <li><a href="https://secure.designpad.com/order.html">Order Today!</a></li>
    </ul>

    <h2>Benefits</h2>

    <p>Not just another mousepad. We packed this unique mousepad with functional-
ity by placing those sacred Web colors in an easier and faster to comprehend
arrangement. And the Web Design Pad looks awesome on anyone's desk.</p>


    <h3>The Designs Go Faster on Color Wheels</h3>
    <p>The color wheel format presents colors in an orderly progression, enabling
the user to visualize the sequence of color balance and harmony. Precise hexadeci-
mal values are printed directly on each of the colors. The RGB=HEX conversion
scale is wrapped around the edges of the mousepad. This helps Web developers
choose compatible Web-safe colors and reference correct hexadecimal coding
instantly.</p>


    <h3>Keep Tabs on Your Colors</h3>
    <p>Ever try to find a simple shade of green in other Web color devices? We
did on several occasions for a recent Web project. What took seconds with the Web
Design Pad took us ages on other mousepads and Web editors.</p>


  </body>
</html>
```

This code allows a browser that is rendering the page to know how to properly render the page. Remember that we are breaking away from traditional design, and one of the things we have to do now as Web builders is ensure that our Web pages validate. To validate our code, a validator like the one at W3C (see `http://validator.w3.org/`) needs to know what HTML specification you are coding in. The same goes for the browser. As the browser tries to render your Web document, it uses the DOCTYPE tag as a guide for tips on how to render your content.

The current generation of browsers actually has two rendering engines. These browsers render out a page in two modes: Strict or Quirks. If you use the `DOCTYPE` for current flavors of HTML or XHTML, you are going to get your content rendered in the way that's more closely spelled out in Web standard, even if HTML hacks stop working. This state of rendering content is called Strict mode, or Standards mode.

On the flip side, if a browser doesn't understand the `DOCTYPE`, the rendering engine goes to Quirks mode—a state in which the browser renders content in a manner that Web developers are used to seeing from older browsers, warts and all.

For More Information

For more information about how `DOCTYPE`s turn on a browser's Strict mode or Quirks mode, look at these Web pages:

- DOCTYPE grid showing which mode occurs in browsers (IE, Navigator, and Mozilla) with a specific DOCTYPE. See `http://gutfeldt.ch/matthias/articles/ doctypeswitch/table.html`.

- Microsoft's documentation about implementations in IE 6. See `http://msdn.microsoft.com/library/default.asp?ur l=/library/enus/dnie60/html/cssenhancements.asp`.

- Mozilla's documentation about DOCTYPE sniffing. See `http://mozilla.org/docs/Web-developer/quirks/ doctypes.html`.

Knowing the DOCTYPEs to use becomes important when considering how we want browsers to render our content. Here is a listing of valid DTDs for HTML 4.01:

- **This DTD contains recommended HTML elements without the deprecated elements, such as FONT:**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
```

- **This DTD contains the deprecated tags (see Appendix C, "HTML 4.01 Reference"):**

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

- **If you use a frameset, this is the DTD you want to use:**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
  "http://www.w3.org/TR/html4/frameset.dtd">
```

For XHTML, which requires stricter markup syntax, these are the valid DTDs:

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

To check whether the code validates, download the file from the book's Web site and see http://validator.w3.org/. Better yet, if you have your own Web site, run your Web pages to the validator.

Include Navigation

After you have marked up your content, you need to provide the page navigation. Do not confuse this with site-wide navigation. This page navigation allows users who can't skim over the navigation menu a chance to skip through the repetitive site-wide menu that's on every page.

A page typically consists of a header, some content, and a footer. In your content, you might have subsections of information besides the main content. Main content could be an article, whereas subsections of content could be a subscription form for a newsletter, advertising banners, or related links to the main content. The footer contains the copyright information, links to privacy policy, and other legal information.

Page navigation should fit into a page layout in this manner:

1. **Header**
2. **Page Navigation**
3. **Site Navigation**
4. **Content**
5. **Footer**

A page navigation contains the following links:

```
<div class="pagenav">
  <ul>
    <a href="#content">Content</a></li>
    <a href="#navigation">Navigation</a></li>
    <a href="#footer">footer</a></li>
  </ul>
</div>
```

We apply the following anchors to the previous markup along with the page navigation (see Listing 1.3).

Listing 1.3   Applying the Page Navigation to the Web Document

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Web Design Pad - a better mousepad by design for Web designers,
    developers, and builders</title>
  </head>
  <body>
    <h1>Web Design Pad</h1>

<p>A better mousepad by design for Web designers, developers, and builders</p>

<div class="pagenav">
  <ul>
    <a href="#content">Content</a></li>
    <a href="#navigation">Navigation</a></li>
    <a href="#footer">footer</a></li>[sr]
  </ul>
</div>


    <a name="navigation"></a>
    <ul>
      <li><a href="benefits.html">Benefits</a></li>
```

Listing 1.3   Applying the Page Navigation to the Web Document   (Continued)

```
      <li><a href="faq.html">Frequently Asked Questions</a></li>
      <li><a href="press.html">Press and Testimonials</a></li>
      <li><a href="https://secure.designpad.com/order.html">Order Today!</a></li>
    </ul>


    <a name="content"></a>
    <h2>Benefits</h2>

    <p>Not just another mousepad. We packed this unique mousepad with
functionality, by placing those sacred Web colors in an easier and faster to
comprehend arrangement. And the Web Design Pad looks awesome on anyone's
desk.</p>

    <h3>The Designs Go Faster on Color Wheels</h3>
    <p>The color wheel format presents colors in an orderly progression,
enabling the user to visualize the sequence of color balance and harmony. Precise
hexadecimal values are printed directly on each of the colors. The RGB=HEX
conversion scale is wrapped around the edges of the mousepad. This helps Web
developers choose compatible Web-safe colors and reference correct hexadecimal
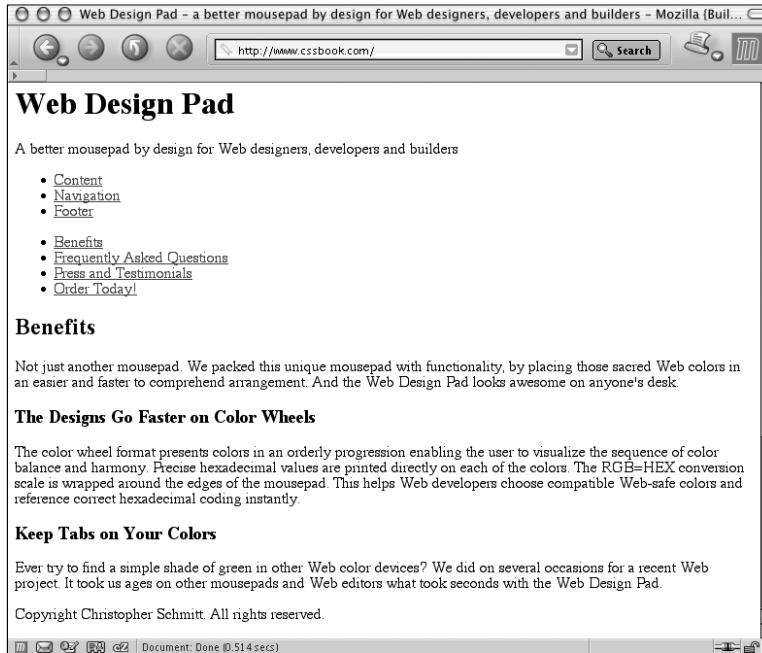coding instantly.</p>

    <h3>Keep Tabs on Your Colors</h3>
    <p>Ever try to find a simple shade of green in other Web color devices? We
did on several occasions for a recent Web project. What took seconds with the
Web Design Pad took us ages on other mousepads and Web editors.</p>



    <a name="footer"></a>
    <p>Copyright Christopher Schmitt. All rights reserved.</p>

  </body>
</html>
```

Figure 1.17 shows Listing 1.3 on a Web Design Pad.

Now that we have the page navigation in place with working anchors, let's hide the page navigation from the visual display by declaring a style (see Listing 1.4).

Listing 1.4   Declaring a Style

```
<style type="text/css">
  .pagenav  {
    display: none;
  }
</style>
```

Now we separate the areas of header, content, and footer (see Listing 1.5).

Listing 1.5   Separating the Header, Content, and Footer

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Web Design Pad - a better mousepad by design for Web designers,
developers, and builders</title>
  </head>
  <body>
```

continues

Listing 1.5   Separating the Header, Content, and Footer   (Continued)

```
     <div class="header">

       <h1>Web Design Pad</h1>

<p>A better mousepad by design for Web designers, developers, and builders</p>

       <div class="pagenav">
          <a href="#content">Content</a>
          <a href="#navigation">Navigation</a>
          <a href="#footer">footer</a>
       </div>


       <a name="navigation"></a>
         <ul>
           <li><a href="benefits.html">Benefits</a>
           <li><a href="faq.html">Frequently Asked Questions</a>
           <li><a href="press.html">Press and Testimonials</a>
           <li><a href="https://secure.designpad.com/order.html">Order Today!</a>
         </ul>

     </div>

     <div class="content">

      <a name="content"></a>
        <h2>Benefits</h2>        <p>Not just another mousepad. We packed this unique
mousepad with functionality by placing those sacred Web colors in an easier and
faster to comprehend arrangement. And the Web Design Pad looks awesome on any-
one's desk.</p>

        <h3>The Designs Go Faster on Color Wheels</h3>
        <p>The color wheel format presents colors in an orderly progression,
enabling the user to visualize the sequence of color balance and harmony.
Precise hexadecimal values are printed directly on each of the colors. The
RGB=HEX
conversion scale is wrapped around the edges of the mousepad. This helps Web
developers choose compatible Web-safe colors and reference correct hexadecimal
coding instantly.</p>

        <h3>Keep Tabs on Your Colors</h3>
        <p>Ever try to find a simple shade of green in other Web color devices? We
did on several occasions for a recent Web project. What took seconds with the
Web Design Pad took us ages on other mousepads and Web editors.</p>
```

```
    </div>

    <div class="footer">
      <a name="footer"></a>
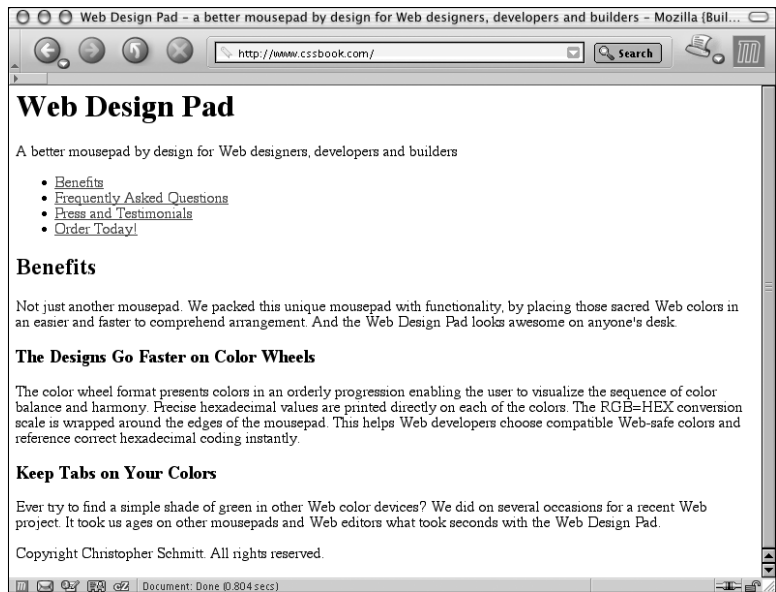      <p>Copyright Christopher Schmitt. All rights reserved.</p>
    </div>

  </body>
</html>
```

Figure 1.18 shows Listing 1.5 on a Web Design Pad.

Figure 1.18

An example with page navigation in place, but hidden through CSS.

Review Content for Design Possibilities

After you have marked up the content, it's time to take a look at the markup look for visual design. Instead of beginning with the design as a means to get to the content, stylize the information you are trying to convey in terms of visual presentation by examining intensely at the message that got to you this point. Look at the tone of the material you have and how a visual design will best serve your client, your client's audience.

Note

For more examples of proper versus improper content structure through HTML, go to the "Incorporating CSS Ideally" section in Chapter 3, "Setting Up Style."

After you have the content marked up structurally, you can look for CSS to design the content for visual presentation a number of ways (see Figure 1.19) without having to rearrange your content. No more will we have to mess with single-pixel GIFs, relying on HTML tables for layouts, or putting up with the FONT tag.

Figure 1.19

Potential design possi-
bilities for one Web
document, leveraged
through the power
of CSS.

In the next chapter, we will look into what makes up contemporary Web design and how to better approach our Web designs.