

# Chapter 2

# Introduction to HTML5

# The Design Process

- What does the company do?
- What is your role in the company?
- Does the company have an existing logo or brand?
- What is your goal in developing a web site?
- What information do you wish to provide online?
- Who comprises your target audience?
- Do its members share any common demographics, like age, sex, or a physical location?
- Who are your competitors and do they have web sites?

*TIP Client Meetings don't have to Take Place in an Office*

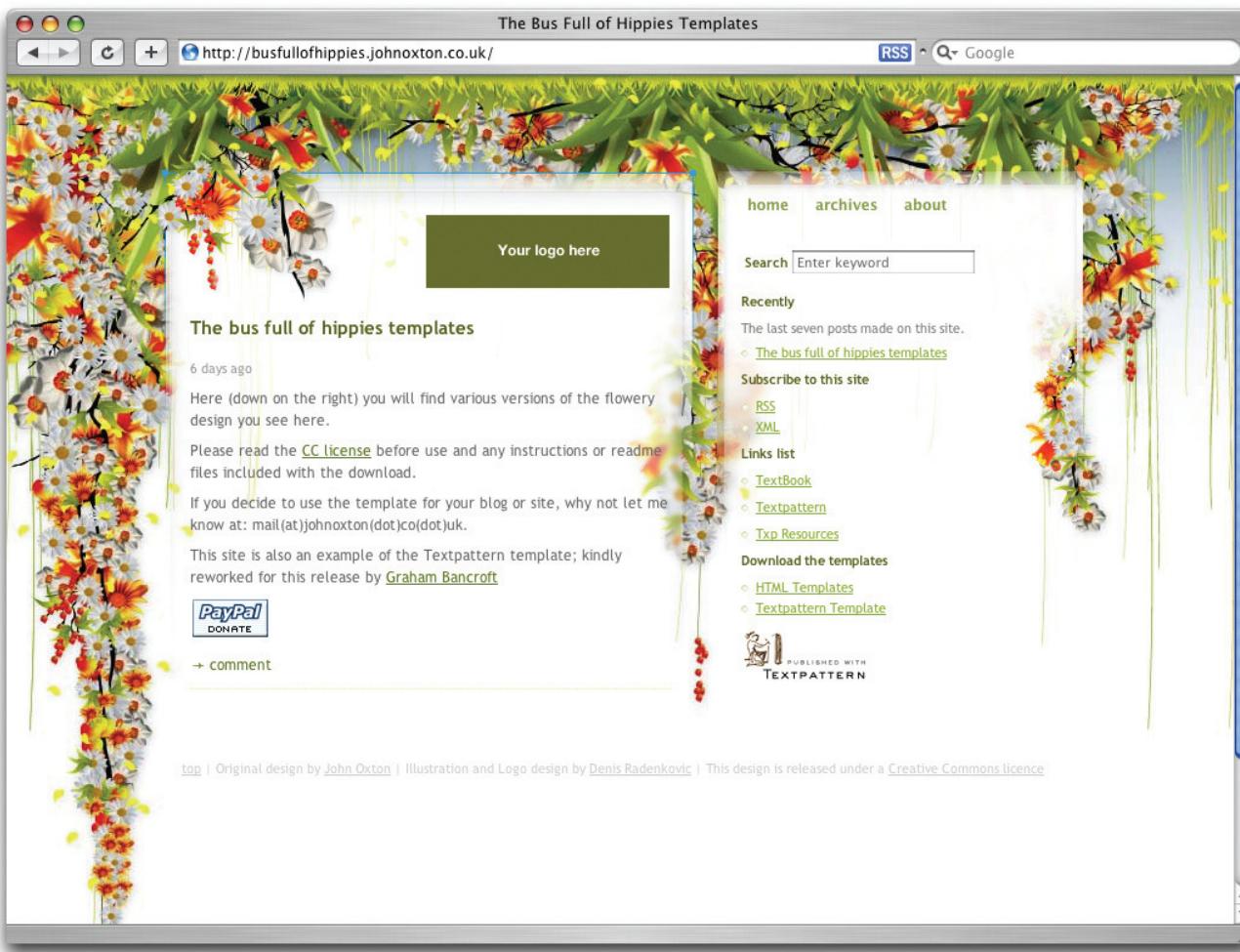


Even when I worked for a company with a big office, I had some of my most productive client meetings at a coffeehouse or over lunch. The feasibility of this approach depends on the client. If your contact doesn't seem like the informal meeting type, don't suggest it; in many cases, though, it's a good way to make a business meeting more personal.

## *NOTE What's a Comp?*

The word **comp** is an abbreviation of the phrase **comprehensive dummy**, and is a term that comes from the print design world. It's a complete simulation of a printed layout that's created before the layout goes to press. In translating this term to web design, a comp is an image of a layout that's created before we begin to prototype the design in HTML.

# Defining Good Design



➤ Users are pleased by the design but drawn to the content

# Defining Good Design



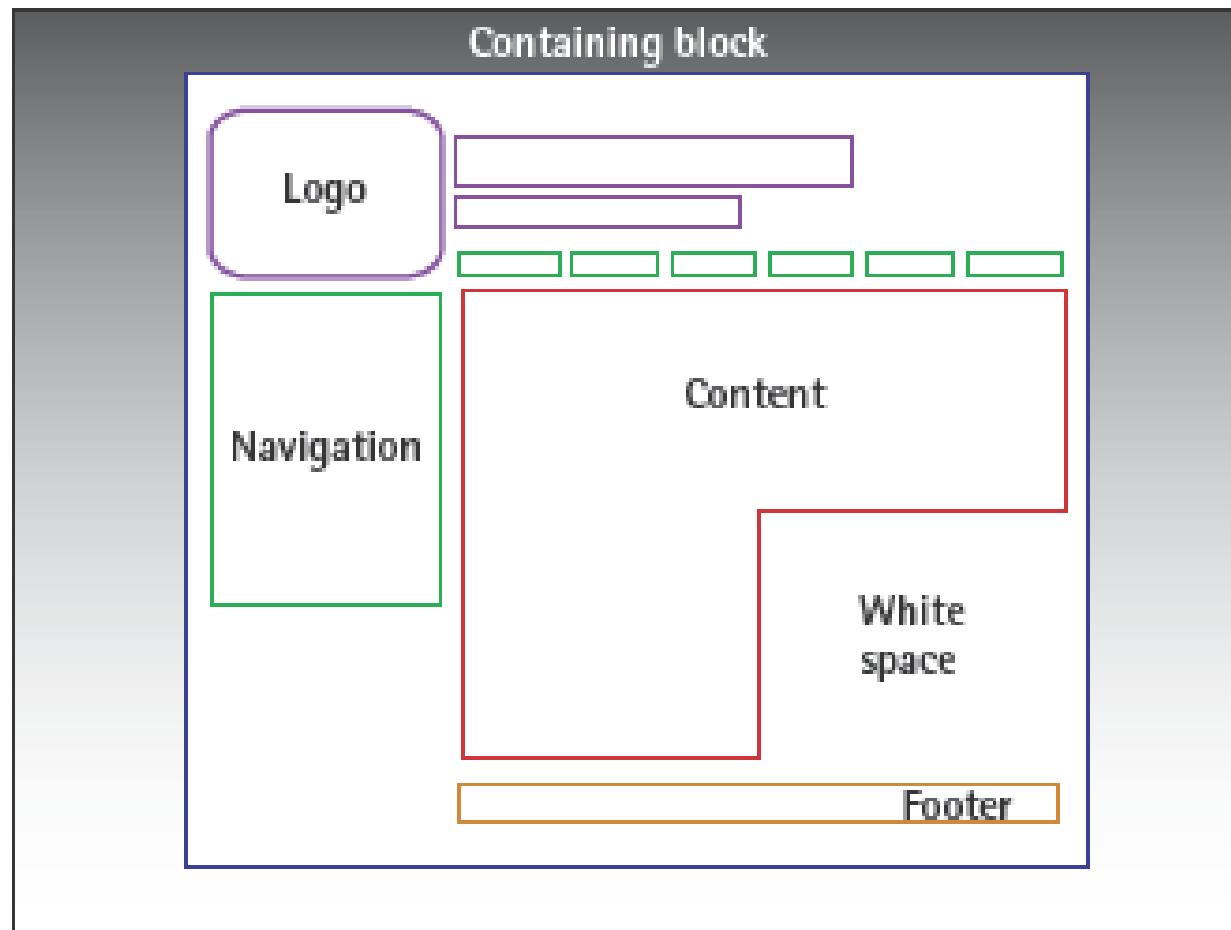
- Users can move about easily via intuitive navigation

# Defining Good Design



➤ Users recognize each page as belonging to the site

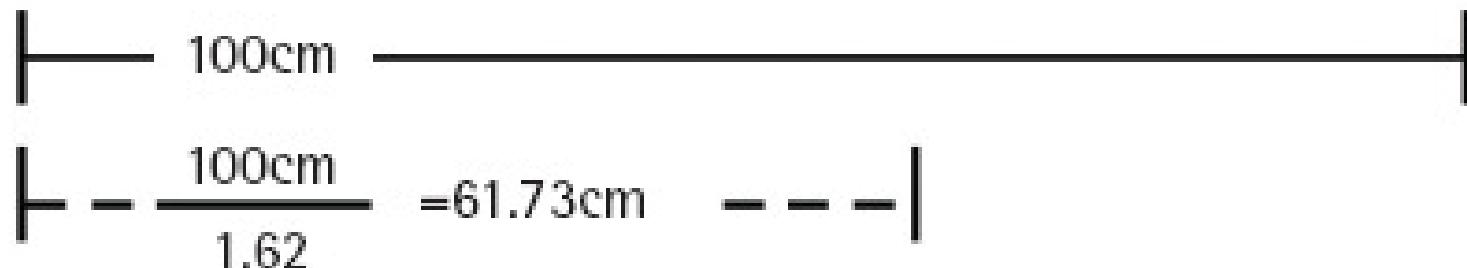
# Web Page Anatomy



## *NOTE Above the Fold*

The fold, as some usability experts call it, is the end of the content users can see on a page before they scroll down. This metaphor is derived from the concept of a fold in a newspaper. If you look at the cover of a folded newspaper, most of the headlines and important news will appear on the top half, so that you can see the most important news items at a glance when the newspaper is folded. The location of the fold on a web page depends on the browser dimensions and the user's screen resolution. At a resolution of 800 x 600 pixels, accounting for browser chrome, the address bar, and a bottom status bar, the fold is usually just over 400 pixels from the top.

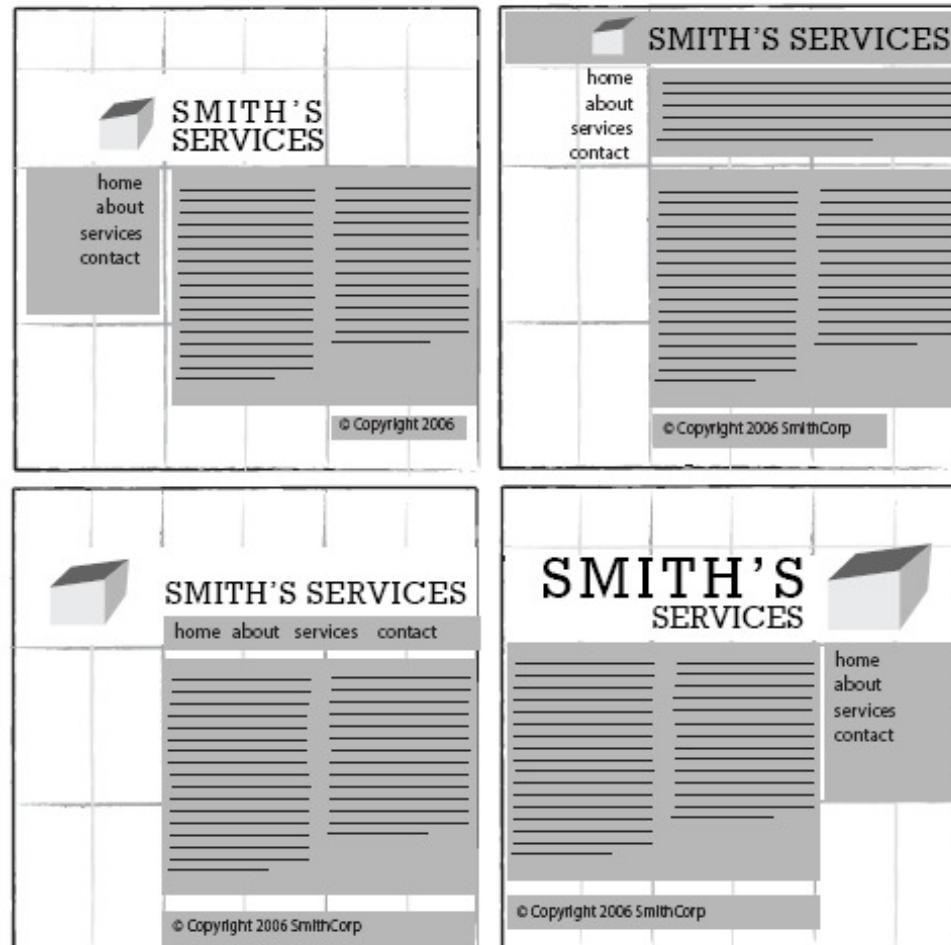
# Grid Theory



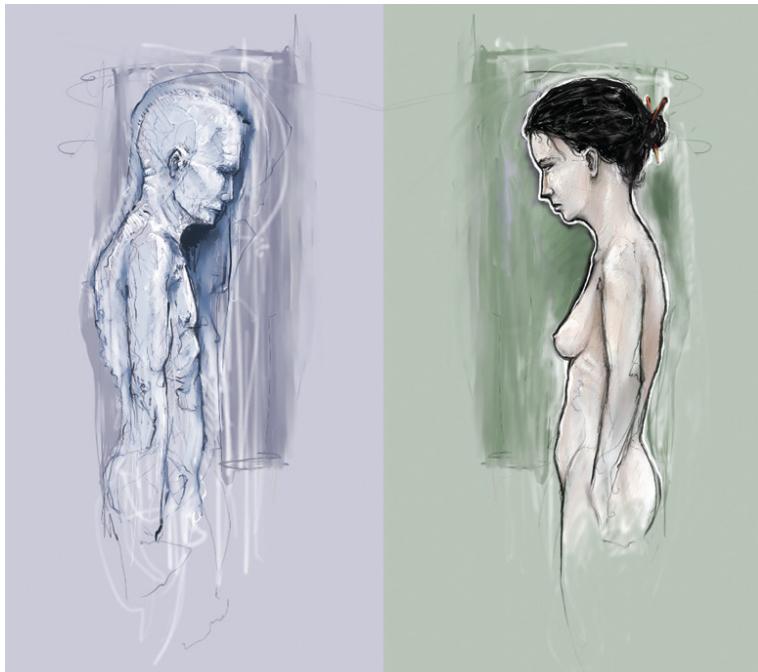
Golden ration



# The Rule of Thirds



# Balance



Symmetrical balance

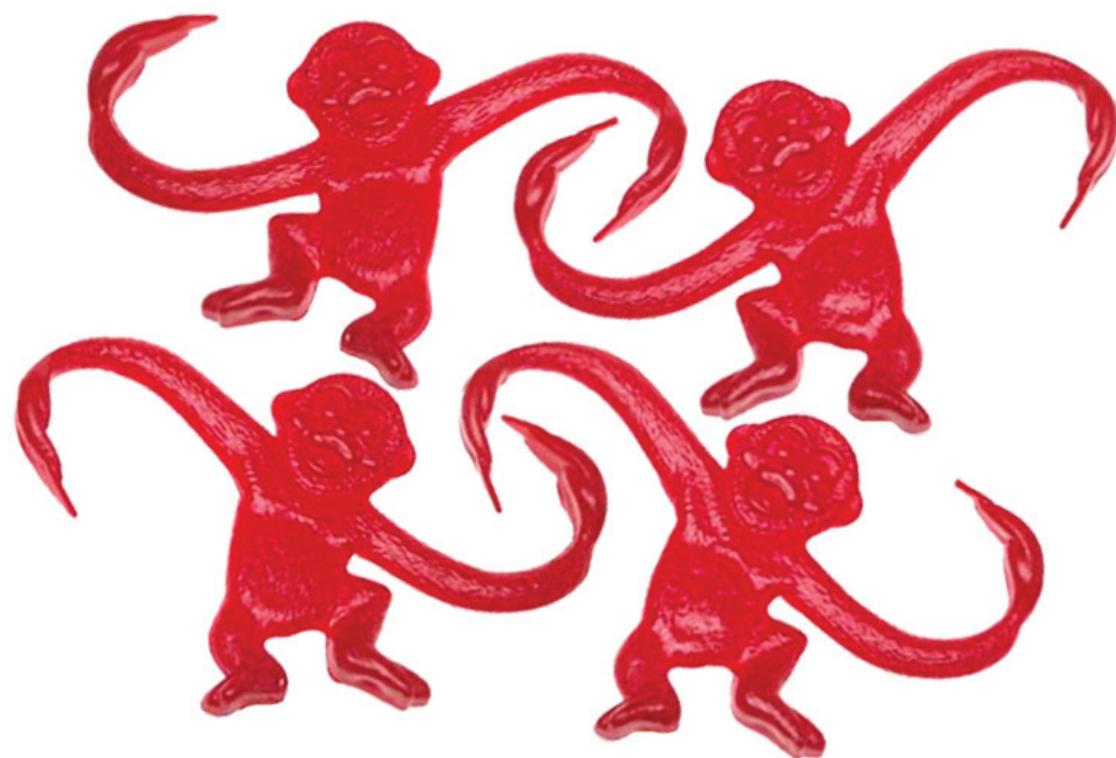
A screenshot of the Grow Collective website. The header features the 'grow' logo and navigation links for 'who we are', 'articles', 'about design', 'about AMOS', 'about hosting', and 'talk to us'. Below the header, there's a decorative graphic of a plant growing across the page. The main content area has three columns: 'about us' (describing the company as a co-operative based in Bristol), 'new growth' (mentioning CSS Mania awards), and 'from the library' (links to 'Work Book' and 'Hosting Book'). There's also a 'selected work' section with thumbnail images and a 'more growth' section about Scooch Slide Show 1.1.

# Asymmetrical Balance



A screenshot of the hicksdesign website. The header reads 'hicksdesign: design for print and new media'. The main content area has a white background. To the right is a sidebar with a dark brown background featuring a colorful circular logo. The sidebar contains links for 'recent work' (with a grid of thumbnail images), 'our clients' (with a list of clients like Mozilla Foundation, Westciv, etc.), 'contact' (with phone number +44 (0)7917 391 536 and a 'Download my vCard' link), and 'work' (with sub-links for 'about', 'journal', 'extras', and 'contact'). At the bottom of the sidebar, there are links for 'Turn off styling', '©2003-06 hicksdesign', and 'RSS Feeds | Site map'.

# Unity



# Proximity



augue duis dolore te feugait nulla. Ea commodo consequat. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

## Un gnome

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequatur, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla. Ea commodo consequat. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.



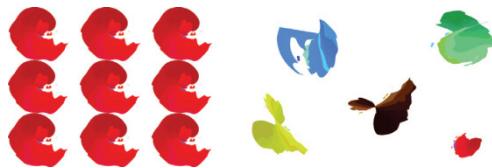
Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequatur, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla. Ea commodo consequat. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

## Gnomencature

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequatur, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio

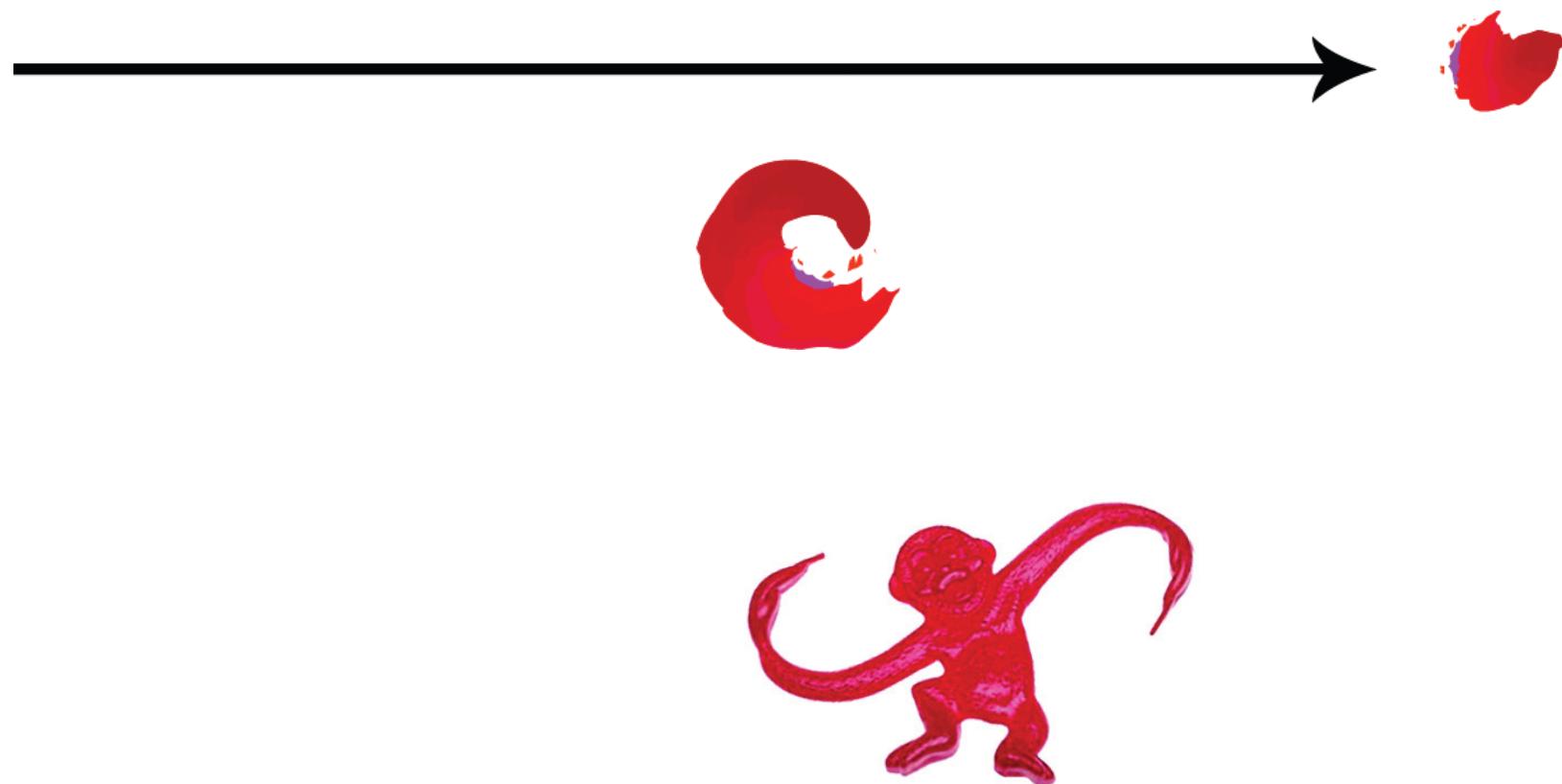


# Repetition

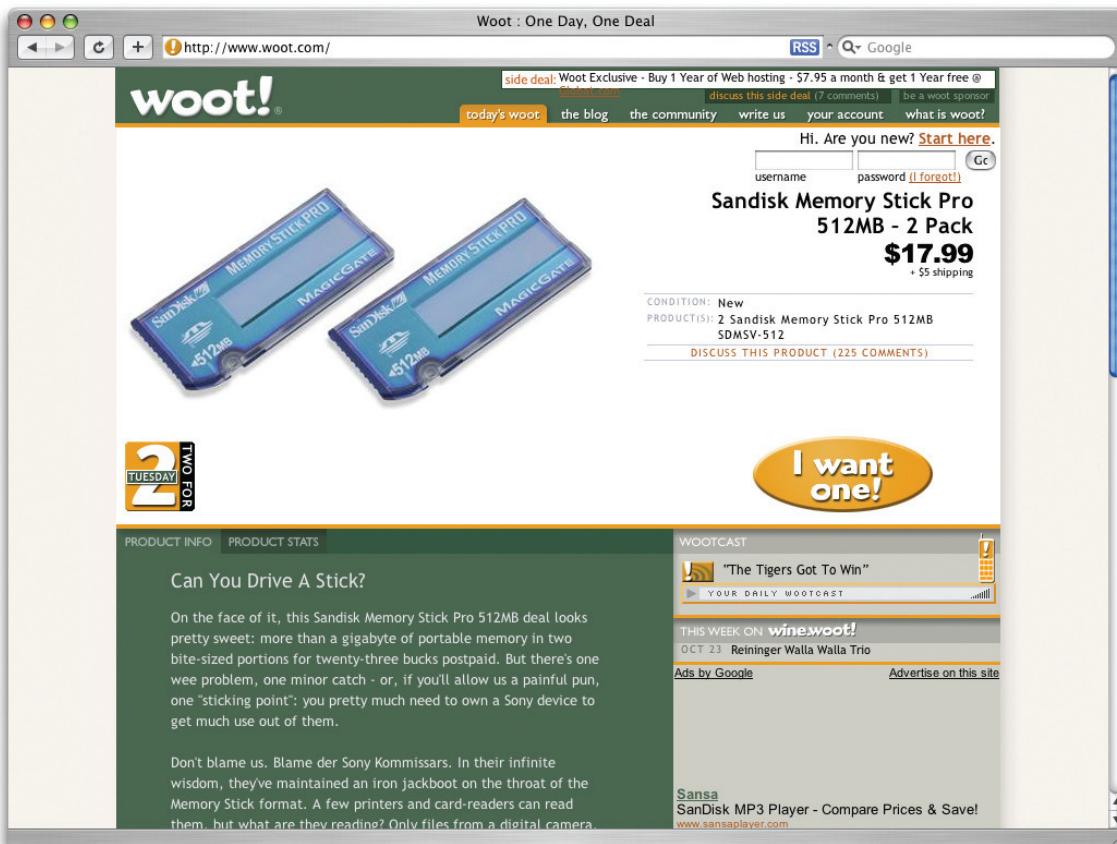


The screenshot shows a web browser window titled "Left Justified" with the URL <http://leftjustified.net/>. The page has a whimsical, hand-drawn aesthetic. At the top, there's a large illustration of a woman in a green bikini holding a fishing rod, standing next to a large green stapler. The title "Left Justified." is written in a large, white, cursive font. Below the title is a "WELCOME!" message and a small image of a bonsai tree. The main menu includes links for "home", "folio", "journal", "the lab", "contact", and "about". On the left, there's a sidebar with a "CATEGORIES" section containing links like "General", "Cascading Style", "Music", "Design", "Fun", "Web Nerdery", "Site News", and "Wordpress". The right side features a "LATEST ENTRY" section about moving to Melbourne, dated October 16, 2006, with 21 comments. The overall design uses a red wooden frame motif.

# Placement, Continuance, Isolation



# Contrast



# Proportion



Jasongraphix :: A journal of art, thoughts, and projects by Jason Baird.

Home Archive Folio Artwork About Résumé

**Step 'n Side**

**October 17, 2006**

About a week and a half ago, my friend Jeremy Darty posted the first episode of his new podcast - *Step 'n Side*. Back in the day Jeremy and I were the two renegade PC users in the graphic design program at UCF. As fate would have it, we both work on Macs full time now and I'm guessing he even produces his show in Garage Band. In his podcast, Jeremy is exploring and exposing some great alternative and independent bands and exploring a steady stream of new [podsafe](#) music.

On Monday, Jeremy posted the second episode of his podcast and I have to say that there hasn't been a single song in either episode that I didn't like. So far, he has played music from Brother Love, the Collision Junk, Darren Geffen, Ravko Pipes and the Kenny Ryan Band, Megaphone, and Seven Mary Three. Of all those bands, I had only really heard of Brother Love and (of course) Seven Mary Three. It's pretty crazy that he got Seven Mary Three's management to ask him to play on their show. I am so stoked. They're a big name and have been around for about 10 years now. Back in 2002 Jeremy and I got to see them play at *Spirit Splash*, one of UCF's annual homecoming traditions. When they got on stage they announced that although they had never canceled a show before, they almost canceled ours because lead singer Jason Ross was sick and had pretty much lost his voice. When they started to play "Cumbbersome", Jeremy and a few other guys jumped up on stage and got to help Jason belt out the chorus. Good Times!

Be sure to check out [Step 'n Side](#) and let Jeremy know what you think about the music. He might even play your comment on the air and call you out like he did to me. :)

07:49 AM/ Music - No Comments

Making this place feel like home.

**jasongraphix**

Select Style  Submit

Featured Work amesnjas v2

Friends Amesnjas Birdman Bojan Janjanin Brian Reese Christilina David Lanham Dustin Diaz Elliot Swan Jeremy Darty Justin Perkins Marko Dugonjic Ray Williams

Cancelled opening the page

# Proportion



Jasongraphix :: A journal of art, thoughts, and projects by Jason Baird.

Home Archive Folio Artwork About Résumé

**Step 'n Side**

**October 17, 2006**

About a week and a half ago, my friend Jeremy Darty posted the first episode of his new podcast - *Step 'n Side*. Back in the day Jeremy and I were the two renegade PC users in the graphic design program at UCF. As fate would have it, we both work on Macs full time now and I'm guessing he even produces his show in Garage Band. In his podcast, Jeremy is exploring and exposing some great alternative and independent bands and exploring a steady stream of new [podsafe](#) music.

On Monday, Jeremy posted the second episode of his podcast and I have to say that there hasn't been a single song in either episode that I didn't like. So far, he has played music from Brother Love, the Collisionink, Darren Geffen, Ravko Pipes and the Kenny Ryan Band, Megaphone, and Seven Mary Three. Of all those bands, I had only really heard of Brother Love and (of course) Seven Mary Three. It's pretty crazy that he got Seven Mary Three's management to ask him to play on their show. I am so stoked. They're a big name and have been around for about 10 years now. Back in 2002 Jeremy and I got to see them play at *Spirit Splash*, one of UCF's annual homecoming traditions. When they got on stage they announced that although they had never canceled a show before, they almost canceled ours because lead singer Jason Ross was sick and had pretty much lost his voice. When they started to play "Cumbbersome", Jeremy and a few other guys jumped up on stage and got to help Jason belt out the chorus. Good Times!

Be sure to check out [Step 'n Side](#) and let Jeremy know what you think about the music. He might even play your comment on the air and call you out like he did to me. :)

07:49 AM/ Music - No Comments

Making this place feel like home.

**jasongraphix**

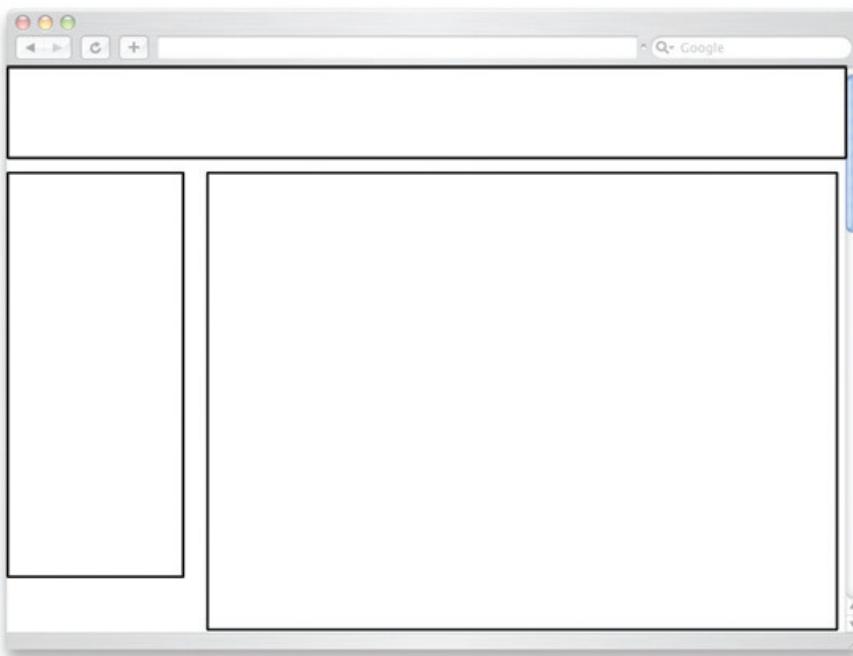
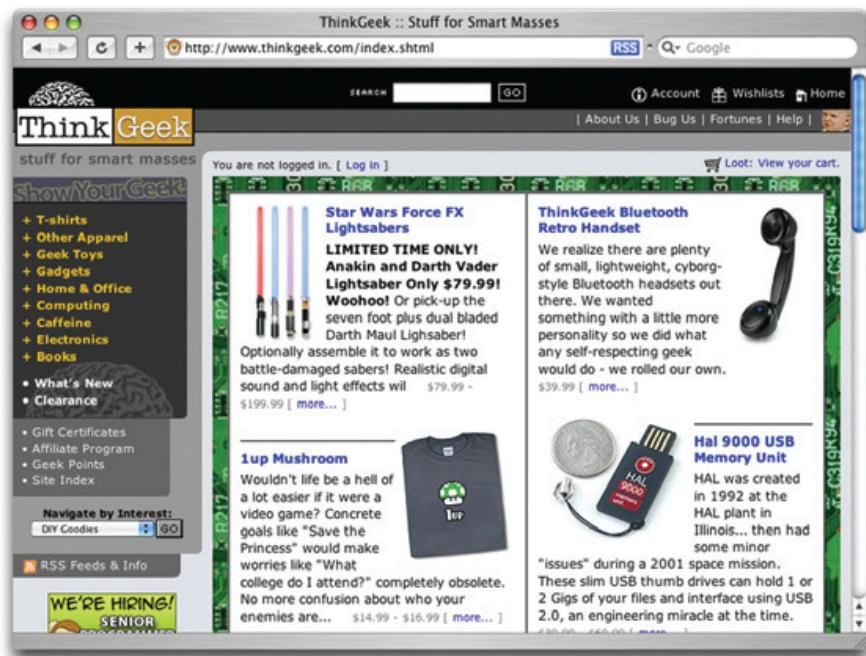
Select Style  Submit

Featured Work amesnjas v2

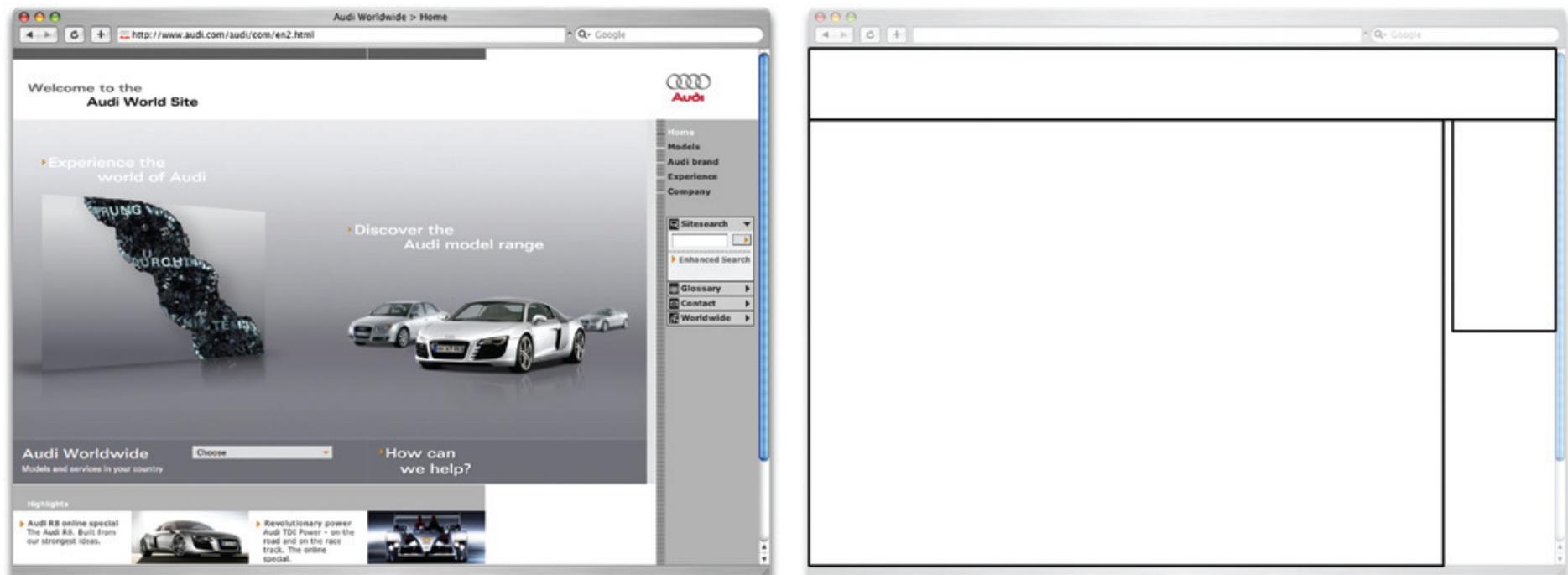
Friends Amesnjas Birdman Bojan Janjanin Brian Reese Christilna David Lanham Dustin Diaz Elliot Swan Jeremy Darty Justin Perkins Marko Dugonjic Ray Williams

Cancelled opening the page

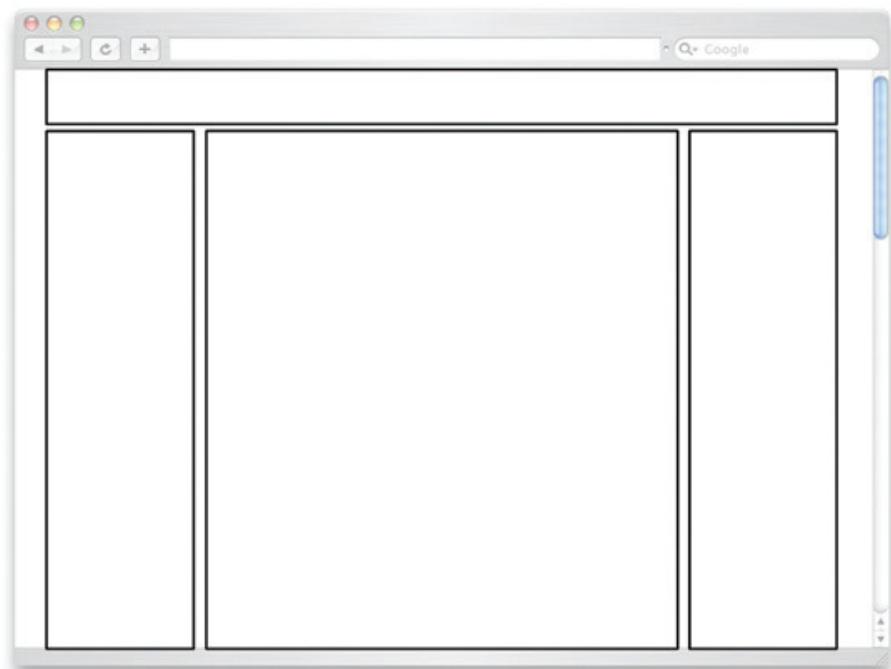
# Navigations



# Navigations



# Navigations



# Getting Inspired

- CSS Zen Garden at

<http://www.csszengarden.com/>

This site is the original showcase of fresh ideas for CSS.

- CSS Beauty at <http://www.cssbeauty.com/>

CSS Beauty is both a gallery of well-designed CSS web sites and a portal to the CSS design community.

# Expansive Footer Navigation

The screenshot shows a Mac OS X-style web browser window displaying the JeffCroft.com archive. The main content area lists three blog posts:

- Has accessibility been taken too far?** (Aug 21st, 2006) - 89 comments
- A helping hand for Dvorak** (Aug 7th, 2006) - 28 comments
- What I want to see in Leopard** (Aug 6th, 2006) - 18 comments

On the right side of the page, there is a sidebar with a large list of links under the heading "socialbookmarking". Other sections include "About the author", "What next?", and "Feeds".

**About the author**

I am a web and graphic designer focused on web standards-based development. I currently live and work Lawrence, KS. As the senior designer at World Online, I work on such award-winning standards-based sites as [lawrence.com](#), [kusports.com](#), and [ljworld.com](#).

I have been working full-time on the web since 1995. In addition to my work with World Online, I have worked at two major Universities in an effort to bring web standards to the education sector, and completed many freelance and contract jobs for varying clients.

When not hunched over a computer, I enjoy photography, music, film, television, and a good night out on the town.

[FULL BIO](#)

**What next?**

If you've come this far, you may also like:

- [Blog archive](#)
- [Blog tags](#)
- [Photos archive](#)
- [Photos tags](#)
- [Links archive](#)
- [Links tags](#)
- [Comments archive](#)

**Feeds**

The following feeds are available to meet all your syndication and aggregation fantasies. There are also per-post comment feeds available on each blog entry page.

- [Full blog posts](#)
- [Blog post summaries](#)
- [Site comments](#)
- [Links](#)
- [flickr Photos](#)

© 1999–2006 Jeffrey A. Croft. Hosted by [Dreamhost](#) and [Gypsy Hosting](#). Views expressed are not necessarily those of my employer.

[A django JOINT](#)

# Three Columns with Content First

The screenshot displays a web browser window for the Vivabit website. The title bar reads "Vivabit | The Web's Cutting Edge". The address bar shows the URL "http://www.vivabit.com/". A search bar on the right contains the text "Google". The main content area features a large banner with a green and yellow leaf on the left and the text "THE WEB'S CUTTING EDGE." on the right. Below the banner, there are three columns:

- Welcome:** Vivabit brings events and the web together. We organise web design conferences, in the guise of [@media](#) and we produce [Event Wax](#), a web-based administration tool for event organisers.
- @media 2006:** 800 people from over 30 countries gathered in London in mid-June to see some of the world's most highly respected web experts speak at [@media 2006](#).
- Smarter Event Management:** [Event Wax](#) has just the right amount of powerful features snuggled inside an easy to use platform to be a massive benefit both for yourself and for your event's attendees.
  - Searchable, sortable access to all booking details
  - Multiple ticket types
  - Affordable packages based on capacity
  - Free to use for small events
  - Targeted mailing lists
  - Hook up to your own payment provider
  - Optional hosted web site for your

**DESIGNING for the MIND**

**DO YOU KNOW WHAT MAKES A DESIGN GOOD?**

In it's many iterations, art is there something more to it? Creating design that is good, meaningful, and useful is a challenge. Designers have to constantly juggle creating user worthy interfaces and have natural abilities that can't easily be taught or learned. But what if there were principles that explained why design is good? As a designer, the second most important is that you can't make things up. As a designer, the second most important is that you can't make things up.

There is a real power that comes from being able to identify, and speak to what makes something meaningful. In the words of the author from *Design: The Craft* that once you recognize and understand something, it creates a new level of appreciation for it. This is true for design as well. By understanding the basic elements of design principles, you can explain them to others. The more you become conscious of how design works, the better you are able to communicate and judge design decisions.

**HELLO, I'M YOUR BRAIN**

Design is powerful because of the way we process information. Any project, whether it's a logo, a website, or a book cover, is put into categories and organized based on the brain's ability to process information. There are about 100 billion neurons in the brain, and each neuron has about 10,000 connections to other neurons. There are about 100 trillion connections in the brain. The brain's ability to process information is limited by the number of neurons it has, so there are many ways to organize them in the same way.

The principles below are based on research of the brain and neurological studies. These principles are not absolute, as they are not universal truths, but they are useful for design and always work for most people. These principles help us to understand how the brain processes information, and how it can be organized to be more effective.

**PRINCIPLES OF THE AESTHETIC EXPERIENCE**

**FedEx.**

**CINDERELLA STORY**

**FIRST ROUND**

**FINAL FINAL**

**AND YET, DESPITE IT'S LAY-BACK CHARM, FINAL FINAL PROVES NO MATCH FOR THE EVEN MORE CHARMING ERIC CLAPTON. WHO FINDS OUT WHAT HE MEAN FORTUNE CAN DOH.**

**THE BOLD NATE**

**CHART JUNK ISN'T AS BAD AS YOU THINK**

**INFORMATION OVERLOAD IS NOT THE PROBLEM**

**A discourse on the process of designing for real people.**

**CHART JUNK ISN'T AS BAD AS YOU THINK**

This week's returned panel member is **Brian Tufts**, a master infographics reading from TV news reports to corporate info-graphics, he's got problems. **Information Overload is Not the Problem** is the title of his article. The main problem with these illustrations is that it's too hard to figure them out. They include information we don't need that distracts from the real story. It's chart junk.

Edward Tufte coined chart junk in his 1983 book *The Visual Display of Quantitative Information*. In the infographics project that got added to a chart in a graph which add no value, distracting viewers with information that isn't vital to communicate. Chart junk happens when the chart obscures the point of the information more important than the information the chart is trying to communicate.

As you create charts and graphs, are you adding a story? Are you presenting your data and message in a design that fits the story? In this discussion, we'll go through an infographics narrative, see if the data alone (through the chart) always provides. Does people who have certain biases see the logic of chart design have better performance on chart junk tests? Miguel Fornaris book entitled, *Design is Good* (in *Designing Charts and Diagrams*, makes me realize just looking at the chart.

But there might be a new side to this story. A newly published paper entitled, "Visualizing the Effects of Information Overload on Cognitive Function and Memory" suggests that chart junk is as bad as it seems. In fact, it can even be valuable in certain circumstances. If you are trying to maximize recall and general trends, then having cluttered chart junk tends to build better associations and help people remember a better story.

One of the examples used in the report was a chart about the price of diamonds in the last ten and even five years. Rather than just using a regular line graph, the researchers used a series of graphs to highlight the information. In the wrong hands this can be misleading, but with the right tools, it's a great way to present data.

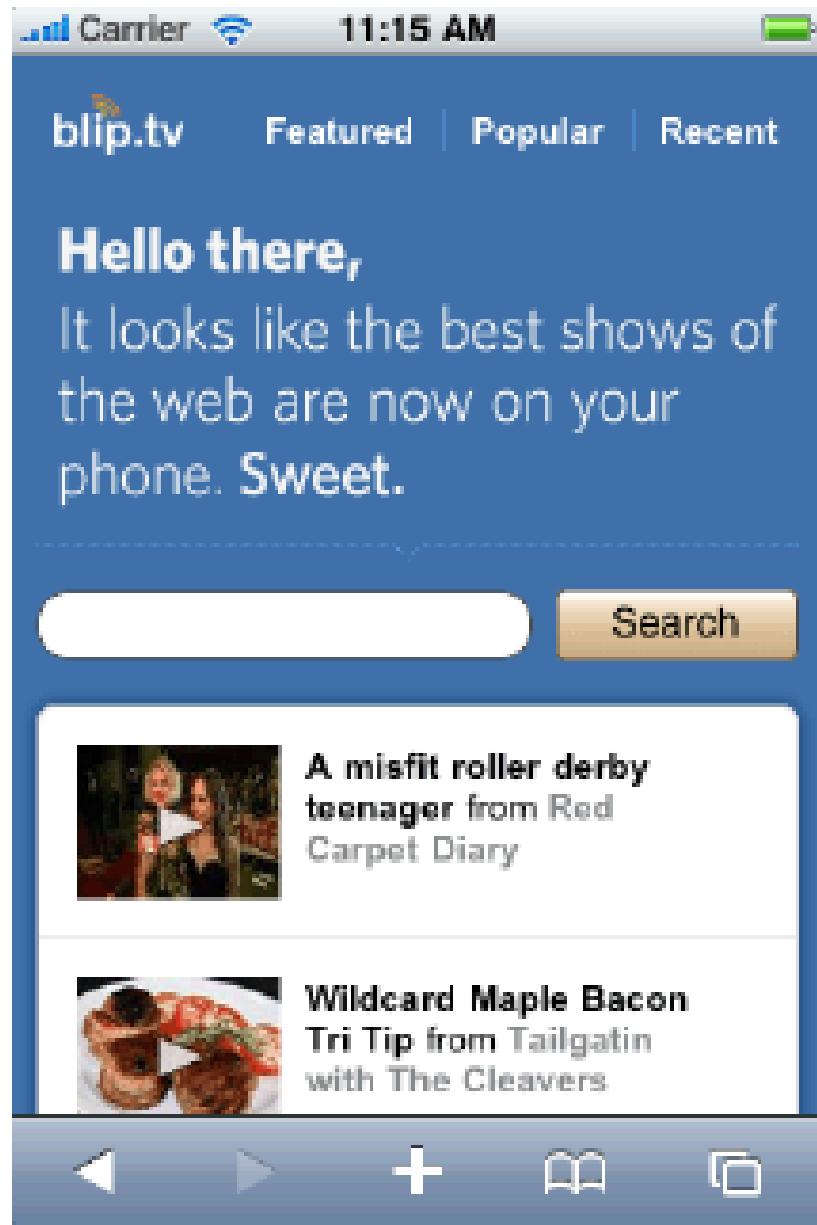
These results suggest that information overload doesn't always bad. You don't always have to show data and only data when presenting. In certain circumstances, more data and with memory and recall through necessary infographics.

In other cases, such as when you are displaying user values for user experiences, then minimizing ink and maximizing chart junk is preferred.

You always need to remember that your goals over medium goals and the data may not always align and you need to choose the right tool for the job. Chart junk junk. In some cases, might just be a suitable tool.

# Resizing: Fixed Width vs. Liquid Width

	Pros	Cons
Fixed width	<ul style="list-style-type: none"><li>■ gives designer more control over how an image floated within the content will look</li><li>■ allows for planned whitespace</li><li>■ improves readability with narrower text blocks</li></ul>	<ul style="list-style-type: none"><li>■ can appear dwarfed in large browser windows</li><li>■ takes control away from the user</li></ul>
Liquid width	<ul style="list-style-type: none"><li>■ adapts to most screen resolutions and devices</li><li>■ reduces user scrolling</li></ul>	<ul style="list-style-type: none"><li>■ challenging to read when text is spanning a wide distance</li><li>■ harder to execute successfully</li><li>■ limits or imposes on whitespace</li></ul>



<a href="#"></a>	<a href="#">CONTACT</a>	<a href="#">PRINTABLE COUPONS</a>	<a href="#">GETTING STARTED GUIDES</a>	<a href="#">GROCERY STORE COUPONS</a>	<a href="#">COUPONING WORKSHOPS</a>	<a href="#">TUTORIALS</a>			
 <h1>Southern Savers</h1> <p>Southern Deals and Frugal Steals</p>									
<a href="#">B-I-LO</a>	<a href="#">CVS</a>	<a href="#">HARRIS TEETER</a>	<a href="#">KROGER</a>	<a href="#">OTHER GROCERY STORES</a>	<a href="#">PUBLIX</a>	<a href="#">RITE AID</a>	<a href="#">TARGET</a>	<a href="#">WALGREENS</a>	<a href="#">WEEKLY INSERTS</a>

## Walgreens Weekly Ad 8/22-8/28

By Jenny · Comments 0

These deals start Sunday 8/22 at Walgreens. It's a long ad but only a few things will probably really jump out at you. If you see any other deals leave a comment below and I'll add it to the list.

Be on the lookout for a lot of random clearance items. Many stores are currently running a clearance sale.

To see the full weekly ad, [go here](#).

If you are new to Walgreens read the [getting started guide](#).

### Register Rewards

- \$5 RR wyb (2) Zytect or Benadryl products including:  
Zytect 25-30 ct \$18.99 ea  
-\$2 off Zytect any RP \$10, My Allergy Guide booklet, [printable](#) or [printable](#)
- -\$4 off Zytect Liquid Gels, 25 ct + [printable](#) (if included)
- Benadryl Allergy Relief 20-34 ct. starting \$4.99
- -\$1 off Benadryl product RP \$10 or My Allergy Guide Booklet (makes it \$1.48 ea)
- -\$1.50 off Benadryl product [printable](#) (limit reached)
- \$5 RR wyb (3) Papel 12 pk \$4 ea (makes it \$2.34 ea)
- \$2 RR wyb (2) Kellogg's Cereal or Pop-Tarts 8 pk \$2.99 ea  
Rice Krispies, Frosted Flakes and Raisin Bran pictured  
-\$1.25 Kellogg's Rice Krispies and/or Cocoa Krispies Cereals, RP \$1.55 and [printable](#)  
-\$1.25 Kellogg's Fruit Loops, Apple Jacks, and Frosted Flakes [printable](#)

New to Southern Savers? Start Here

[Google Custom Search](#)

[Search](#)



#### DEALS BY STORE

[Select Category](#)

#### RECENT POSTS

- Walgreens Weekly Ad 8/22-8/28
- Friday Finals 8/20
- \$500 Shopping Spree @ Linen N Things + Free Shipping :)
- Reminder: Back to School Bach Giveaways Galore
- Recycle Bank: Get Rewarded for Being Responsible
- Publix Menu Plans 8/18
- Rite Aid Weekly Ad

#### PRINTABLE COUPONS





Deals in Your Inbox!  
Subscribe to our email newsletter

[Printables](#) | [Learn to Coupon](#) | [About Me](#) | [Workshops](#) | [Coupons](#)

## Grocery Store Deals

B&O	HARRIS TEETER	KROGER	PUBLIX	TARGET	OTHER
-----	---------------	--------	--------	--------	-------

## Drug Store Deals

CVS	RITE AID	WALGREENS
-----	----------	-----------

Sunday Ad Inserts

Best Deals This Week

## Confused? I can help!

## Coupon Abbreviations

SS = Smart Source

RP = Red Plt

PG = Proctor and Gamble

zzzal

## Couponing Terms

wyb = "when you buy"

b1g1 = buy one get one free

.75/1 = 75 cents off one product

Don't show me this again 

## How to Start Couponing

## Store Coupon Policies

## Where to Get Coupons

## Videos

Collapse  
to a tab  
when  
closed

## Most Recent Blog Entry Title

4/22/10 | Jeong | 294 Comments

## Frozen Foods

- Buy 3 Mystic Pizzas, 16.7-18 oz, up to \$9.99 and get
  - Pint Ice Cream, 36 oz, FREE
  - Southern Home Potato Skins, 7 oz, FREE
  - Coke, 2 liter, FREE
- Bonanza Beef Chuck Roast, 2.5 lb, \$2.99 lb
  - Baby Cut Carrots, 1 lb, FREE
  - Whole Mushrooms, 8 oz, FREE

## Buy One Get One

- Bonanza Pork Loin Half, 3-5 lb, at \$4.39 lb
- Southern Home Medium Shrimp, 16 oz, Price Not Listed
- Hebrew National Beef Franks, 12 oz, at \$4.00 (\$2.49)
  - \$.75/1 Hebrew National Beef Franks, SS 3/07
  - \$1 off Hebrew National Beef Franks, SS 3/07 (makes it \$1.49)
- Southern Home Tray Pack Chicken Strips, Nuggets or Patties, 32 oz, at \$7.99 (\$3.99)
- Southern Home Original Franks, 16 oz, at \$2.89 (\$1.44)
- Carolina Pride Ham or Turkey, 16 oz, at \$3.69 (\$1.84)

## Latest Posts

- ← New Freebies and Coupons →
- [Marshalls \\$25 Gift Card Giveaway and Contest](#)
  - [H-E-B Weekly Ad 4/28-5/4](#)
  - [CVS 2 Free Picture DVDs Available Again](#)
  - [Hanney's & Food Lion Coupons 4/28](#)
  - [H-E-B Weekly Ad 4/28-5/4](#)
  - [CVS 2 Free Picture DVDs Available Again](#)
  - [Hanney's & Food Lion Coupons 4/28](#)
  - [H-E-B Weekly Ad 4/28-5/4](#)
  - [CVS 2 Free Picture DVDs Available Again](#)

[Latest Posts](#)[Categories](#)[Videos](#)

## Print Some Coupons!



See our full list of printable coupon sources

## 2.1 Introduction

- ▶ HTML5 (HyperText Markup Language 5)
  - markup language that specifies the *structure* and *content* of documents that are displayed in web browsers
- ▶ We introduce some basics, then cover more sophisticated HTML5 techniques such as:
  - tables, which are particularly useful for structuring information from databases (i.e., software that stores structured sets of data)
  - forms for collecting information from web-page visitors
  - internal linking for easier page navigation
  - meta elements for specifying information about a document

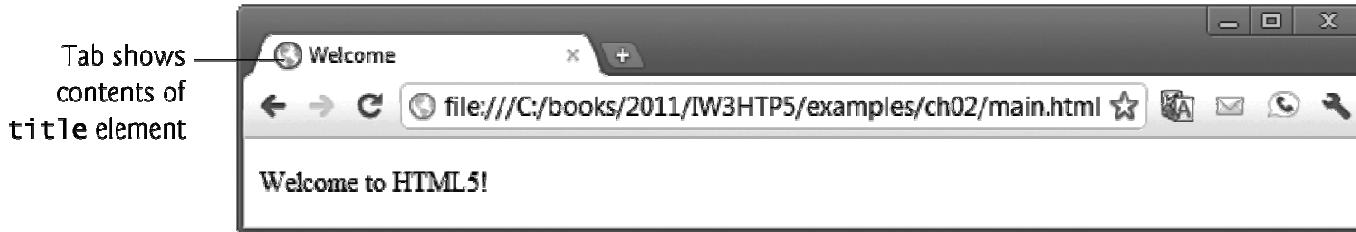
## 2.2 Editing HTML5

- ▶ We'll create HTML5 documents by typing HTML5 markup text in a text editor (such as Notepad,TextEdit, vi, emacs) and saving it with the .html or .htm filename extension.
- ▶ Computers called web servers store HTML5 documents.
- ▶ Clients (such as web browsers running on your local computer or smartphone) request specific resources such as HTML5 documents from web servers.

## 2.3 First HTML5 Example

- ▶ Figure 2.1 is an HTML5 document named `main.html`, which is stored in the `examples/ch02` folder.
- ▶ This first example displays the message `Welcome to HTML5!` in the browser.

```
1 <!DOCTYPE html>
2
3 <!-- Fig. 2.1: main.html -->
4 <!-- First HTML5 example. -->
5 <html>
6   <head>
7     <meta charset = "utf-8">
8     <title>Welcome</title>
9   </head>
10
11  <body>
12    <p>Welcome to HTML5!</p>
13  </body>
14 </html>
```



**Fig. 2.1 |** First HTML5 example.

## 2.3 First HTML5 Example

### *Document Type Declaration*

- ▶ The document type declaration (DOCTYPE) is required in HTML5 documents so that browsers render the page in standards mode.
- ▶ Some browsers operate in quirks mode to maintain backward compatibility with web pages that are not up-to-date with the latest standards.

## 2.3 First HTML5 Example

### *Comments*

- ▶ Insert comments in your HTML5 markup to improve readability and describe the content of a document.
- ▶ The browser ignores comments when your document is rendered.
- ▶ Comments start with `<!--` and end with `-->`.

## 2.3 First HTML5 Example

### *html, head and body Elements*

- ▶ HTML5 markup contains text (and images, graphics, animations, audios and videos) that represents the content of a document and elements that specify a document's *structure and meaning*.
  - The `html` element *encloses* the head section (represented by the `head` element) and the body section (represented by the `body` element).
  - The **head section** contains information about the HTML5 document, such as the character set (UTF-8, the most popular character-encoding scheme for the web) that the page use—which helps the browser determine how to render the content—and the **title**.
  - The head section also can contain special document-formatting instructions called **CSS3 style sheets** and client-side programs called **scripts** for creating dynamic web pages.
  - The **body section** contains the page's content, which the browser displays when the user visits the web page.

## 2.3 First HTML5 Example (cont.)

### *Start Tags and End Tags*

- ▶ HTML5 documents *delimit* most elements with a start tag and end tag.
- ▶ A **start tag** consists of the element name in angle brackets
  - For example, <html>
- ▶ An **end tag** consists of the element name preceded by a forward slash (/) in angle brackets
  - For example, </html>
- ▶ There are several so-called “void elements” that do not have end tags.
- ▶ Many start tags have attributes that provide additional information about an element, which browsers use to determine how to process the element.
- ▶ Each attribute has a **name** and a **value** separated by an equals sign (=).



## Good Programming Practice 2.1

---

Although HTML5 element and attribute names are case insensitive (you can use uppercase and lowercase letters), it's a good practice to use only lowercase letters.

## 2.3 First HTML5 Example (Cont.)

### *Title Element*

- ▶ The title element is called a nested element, because it's enclosed in the head element's start and end tags.
- ▶ The head element is also a nested element, because it's enclosed in the html element's start and end tags.
- ▶ The title element describes the web page.
  - Titles usually appear in the title bar at the top of the browser window, in the browser tab on which the page is displayed, and also as the text identifying a page when users add the page to their list of Favorites or Bookmarks, enabling them to return to their favorite sites.
  - Search engines use the title for indexing purposes and when displaying results



## Good Programming Practice 2.2

---

Indenting nested elements emphasizes a document's structure and promotes readability. We use three spaces for each level of indentation.

## 2.3 First HTML5 Example (Cont.)

*Paragraph Element (<p> . . . </p>)*

- ▶ All text placed between the `<p>` and `</p>` tags forms one paragraph.

## 2.4 W3C HTML5 Validation Service

- ▶ HTML5 documents that are syntactically correct are guaranteed to render properly
- ▶ HTML5 documents that contain syntax errors may not display properly
- ▶ Validation services (e.g., [validator.w3.org/#validate-by-upload](https://validator.w3.org/#validate-by-upload)) ensure that an HTML5 document is syntactically correct



### Error-Prevention Tip 2.1

---

Most browsers attempt to render HTML5 documents even if they're invalid. This can lead to unexpected and undesirable results. Use a validation service, such as the W3C *MarkUp Validation Service*, to confirm that an HTML5 document is syntactically correct.

## 2.5 Headings

- ▶ HTML5 provides six heading elements (h1 through h6) for specifying the *relative importance* of information
  - Heading element h1 is considered the most significant heading and is rendered in the largest font.
  - Each successive heading element (i.e., h2, h3, etc.) is rendered in a progressively smaller font.



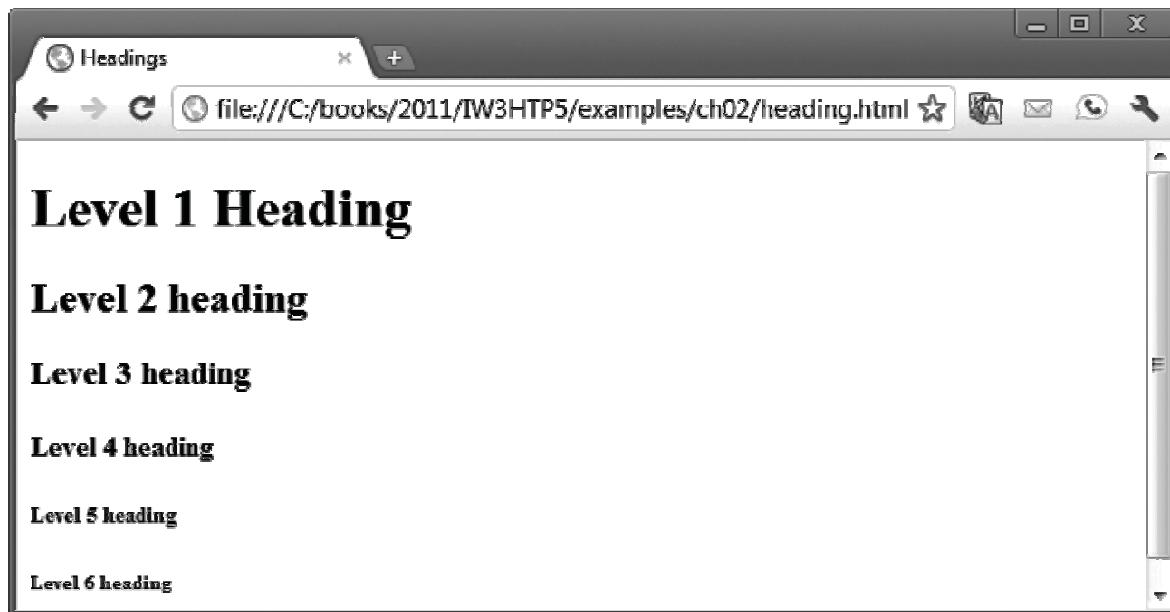
### **Portability Tip 2.1**

---

The text size used to display each heading element can vary between browsers. In Chapter 4, we use CSS to control the text size and other text properties.

```
1 <!DOCTYPE html>
2
3 <!-- Fig. 2.2: heading.html -->
4 <!-- Heading elements h1 through h6. -->
5 <html>
6   <head>
7     <meta charset = "utf-8">
8     <title>Headings</title>
9   </head>
10
11  <body>
12    <h1>Level 1 Heading</h1>
13    <h2>Level 2 heading</h2>
14    <h3>Level 3 heading</h3>
15    <h4>Level 4 heading</h4>
16    <h5>Level 5 heading</h5>
17    <h6>Level 6 heading</h6>
18  </body>
19 </html>
```

**Fig. 2.2** | Heading elements h1 through h6. (Part 1 of 2.)



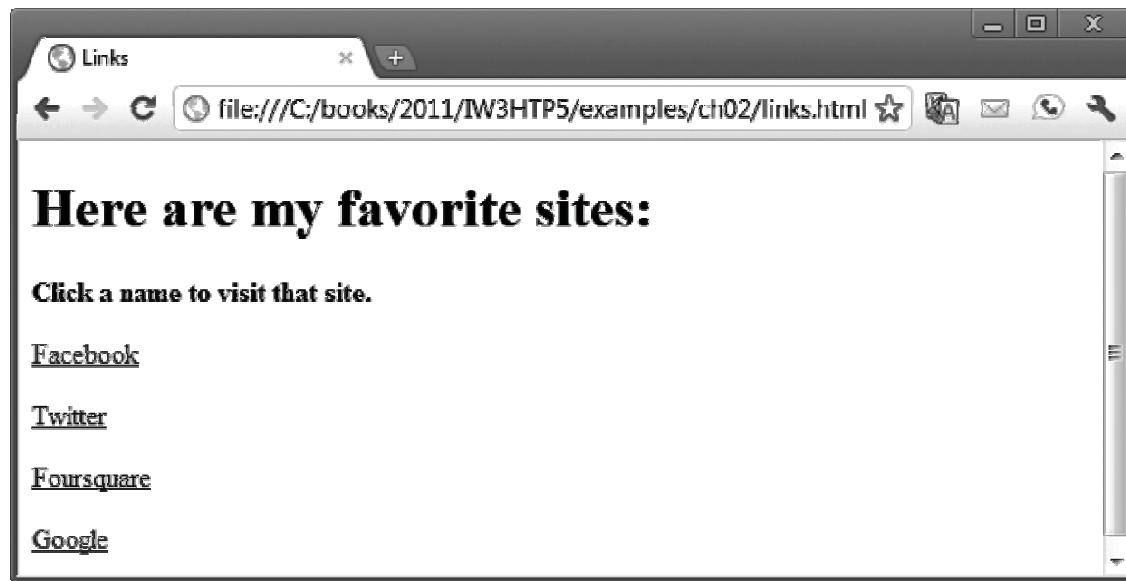
**Fig. 2.2 |** Heading elements h1 through h6. (Part 2 of 2.)

## 2.6 Linking

- ▶ A hyperlink references or links to other resources, such as HTML5 documents and images.
- ▶ Web browsers typically *underline* text hyperlinks and color them *blue* by default.

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.3: links.html -->
4  <!-- Linking to other web pages. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Links</title>
9      </head>
10
11     <body>
12         <h1>Here are my favorite sites:</h1>
13         <p><strong>Click a name to visit that site.</strong></p>
14
15         <!-- create four text hyperlinks -->
16         <p><a href = "http://www.facebook.com">Facebook</a></p>
17         <p><a href = "http://www.twitter.com">Twitter</a></p>
18         <p><a href = "http://www.foursquare.com">Foursquare</a></p>
19         <p><a href = "http://www.google.com">Google</a></p>
20
21     </body>
22 </html>
```

**Fig. 2.3 |** Linking to other web pages. (Part 1 of 2.)



**Fig. 2.3 |** Linking to other web pages. (Part 2 of 2.)

## 2.6 Linking (Cont.)

- ▶ The **strong** element indicates that the content has high importance. Browsers typically render such text in a bold font.
- ▶ Links are created using the **a (anchor)** element.
- ▶ Attribute **href (hypertext reference)** specifies a resource's location, such as
  - a web page or location within a web page
  - a file
  - an e-mail address
- ▶ When a URL does not indicate a specific document on the website, the web server returns a default web page. This page is often called **index.html**, but most web servers can be configured to use any file as the default web page for the site.
- ▶ If the web server cannot locate a requested document, it returns an error indication to the web browser (known as a 404 error), and the browser displays a web page containing an error message.

## 2.6 Linking (Cont.)

### *Hyperlinking to an E-Mail Address*

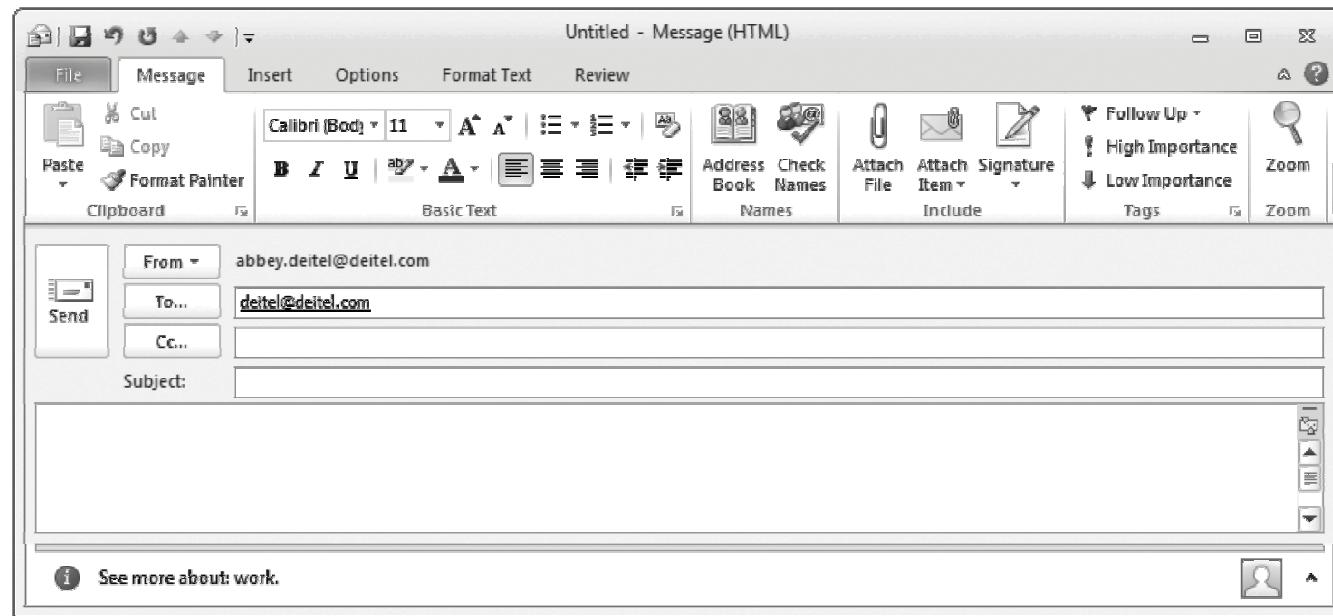
- Anchors can link to an e-mail address using a mailto: URL
  - When a user clicks this type of anchored link, most browsers launch the default e-mail program (e.g., Mozilla Thunderbird, Microsoft Outlook or Apple Mail) to enable the user to write an e-mail message to the linked address.

```
1 <!DOCTYPE html>
2
3 <!-- Fig. 2.4: contact.html -->
4 <!-- Linking to an e-mail address. -->
5 <html>
6   <head>
7     <meta charset = "utf-8">
8     <title>Contact Page</title>
9   </head>
10
11  <body>
12    <p>
13      To write to <a href = "mailto:deitel@deitel.com">
14        Deitel & Associates, Inc.</a>, click the link and your default
15        email client will open an email message and address it to us.
16    </p>
17  </body>
18 </html>
```

**Fig. 2.4** | Linking to an e-mail address. (Part I of 3.)



**Fig. 2.4** | Linking to an e-mail address. (Part 2 of 3.)



**Fig. 2.4** | Linking to an e-mail address. (Part 3 of 3.)

## 2.7 Images

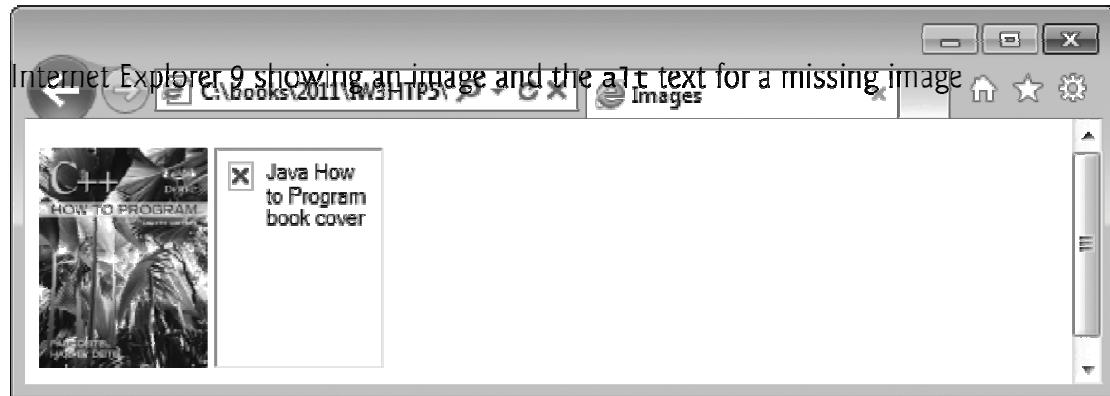
- ▶ The most popular image formats used by web developers today are PNG (Portable Network Graphics) and JPEG (Joint Photographic Experts Group).
- ▶ Users can create images using specialized software, such as Adobe Photoshop Express ([www.photoshop.com](http://www.photoshop.com)), G.I.M.P. ([www.gimp.org](http://www.gimp.org)), Inkscape ([www.inkscape.org](http://www.inkscape.org)) and many more.
- ▶ Images may also be acquired from various websites, many of which offer royalty-free images.

Image-sharing site	URL
Flickr®	<a href="http://www.flickr.com">www.flickr.com</a>
Photobucket	<a href="http://photobucket.com">photobucket.com</a>
Fotki™	<a href="http://www.fotki.com">www.fotki.com</a>
deviantART	<a href="http://www.deviantart.com">www.deviantart.com</a>
Picasa™	<a href="http://picasa.google.com">picasa.google.com</a>
TinyPic®	<a href="http://tinypic.com">tinypic.com</a>
ImageShack	<a href="http://www.imageshack.us">www.imageshack.us</a>
FreeDigitalPhotos.net	<a href="http://www.freedigitalphotos.net">www.freedigitalphotos.net</a>
Open Stock Photography	<a href="http://www.openstockphotography.org">www.openstockphotography.org</a>
Open Clip Art Library	<a href="http://www.openclipart.org">www.openclipart.org</a>

**Fig. 2.5 | Popular image-sharing sites.**

```
1 <!DOCTYPE html>
2
3 <!-- Fig. 2.6: picture.html -->
4 <!-- Including images in HTML5 files. -->
5 <html>
6   <head>
7     <meta charset = "utf-8">
8     <title>Images</title>
9   </head>
10
11  <body>
12    <p>
13      <img src = "cpphttp.png" width = "92" height = "120"
14          alt = "C++ How to Program book cover">
15      <img src = "jhttp.png" width = "92" height = "120"
16          alt = "Java How to Program book cover">
17    </p>
18  </body>
19 </html>
```

**Fig. 2.6** | Including images in HTML5 files. (Part 1 of 2.)



**Fig. 2.6 | Including images in HTML5 files. (Part 2 of 2.)**

## 2.7 Images (cont.)

- ▶ The `img` element's `src` attribute specifies an image's location
- ▶ Every `img` element must have an `alt` attribute, which contains text that is displayed if the client cannot render the image
  - The `alt` attribute makes web pages more accessible to users with disabilities, especially vision impairments
  - `width` and `height` are optional attributes
    - If omitted, the browser uses the image's actual width and height
    - Images are measured in pixels



### **Performance Tip 2.1**

Always include the **width** and the **height** of an image in the `<img>` tag so that when the browser loads the HTML5 file, it will know how much screen space to provide and can lay out the page properly, even before it downloads the image. Including the **width** and **height** attributes in an `<img>` tag can help the browser load and render pages faster.



## **Look-and-Feel Observation 2.2**

---

Entering new dimensions for an image that change its width-to-height ratio distorts the appearance of the image. To avoid distortion, if your image is 200 pixels wide and 100 pixels high, for example, any new dimensions should maintain the 2:1 width-to-height ratio.

## 2.7 Images (Cont.)

### *alt Attribute*

- ▶ A browser may not be able to render an image.
- ▶ Every `img` element in an HTML5 document must have an `alt` attribute.
- ▶ If a browser cannot render an image, the browser displays the `alt` attribute's value.
- ▶ The `alt` attribute is also important for accessibility—speech synthesizer software can speak the `alt` attribute's value so that a visually impaired user can understand what the browser is displaying. For this reason, the `alt` attribute should describe the image's contents.

## 2.7 Images (Cont.)

### *Void Elements*

- ▶ Some HTML5 elements (called **void elements**) contain only attributes and do not mark up text (i.e., text is not placed between a start and an end tag).
- ▶ You can terminate void elements (such as the `img` element) by using the forward slash character (/) inside the closing right angle bracket (>) of the start tag.
- ▶ For example, lines 15–16 of Fig. 2.6 could be written as follows:

```
<img src = "jhttp.png" width = "92" height = "120"  
      alt = "Java How to Program book cover" />
```

## 2.7 Images (Cont.)

### *Using Images as Hyperlinks*

- ▶ By using images as hyperlinks, you can create graphical web pages that link to other resources.
- ▶ In Fig. 2.7, we create five different image hyperlinks.
- ▶ Clicking an image in this example takes the user to a corresponding web page—one of the other examples in this chapter.

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.7: nav.html -->
4  <!-- Images as link anchors. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Navigation Bar</title>
9      </head>
10
11     <body>
12         <p>
13             <a href = "links.html">
14                 <img src = "buttons/links.jpg" width = "65"
15                     height = "50" alt = "Links">
16             </a>
17
18             <a href = "list.html">
19                 <img src = "buttons/list.jpg" width = "65"
20                     height = "50" alt = "List of Features">
21             </a>
22
```

**Fig. 2.7 |** Images as link anchors. (Part 1 of 3.)

```
23      <a href = "contact.html">
24          <img src = "buttons/contact.jpg" width = "65"
25              height = "50" alt = "Contact Me">
26      </a>
27
28      <a href = "table1.html">
29          <img src = "buttons/table.jpg" width = "65"
30              height = "50" alt = "Tables Page">
31      </a>
32
33      <a href = "form.html">
34          <img src = "buttons/form.jpg" width = "65"
35              height = "50" alt = "Feedback Form">
36      </a>
37      </p>
38  </body>
39 </html>
```

**Fig. 2.7 |** Images as link anchors. (Part 2 of 3.)



**Fig. 2.7 |** Images as link anchors. (Part 3 of 3.)

## 2.8 Special Characters and Horizontal Rules

- ▶ HTML5 provides character entity references (in the form &*code*;) for representing special characters that cannot be rendered otherwise
- ▶ The code can be:
  - Word abbreviations
  - Numbers
    - Decimal
    - Hexadecimal

Symbol	Description	Character entity reference
HTML5 character entities		
&	ampersand	&amp;
,	apostrophe	&ampapos
>	greater-than	&gt;
<	less-than	&lt;
"	quote	&quot;
Other common character entities		
non-breaking space		&nbsp;
©	copyright	&copy;
—	em dash	&mdash;
–	en dash	&ndash;
¼	fraction 1/4	&frac14;
½	fraction 1/2	&frac12;

**Fig. 2.8 | Some common HTML character entity references.**

Symbol	Description	Character entity reference
¾	fraction 3/4	&frac34;
...	horizontal ellipsis	&hellip;
®	registered trademark	&reg;
§	section	&sect;
™	trademark	&trade;

**Fig. 2.8 | Some common HTML character entity references.**

## 2.8 Special Characters and Horizontal Rules (Cont.)

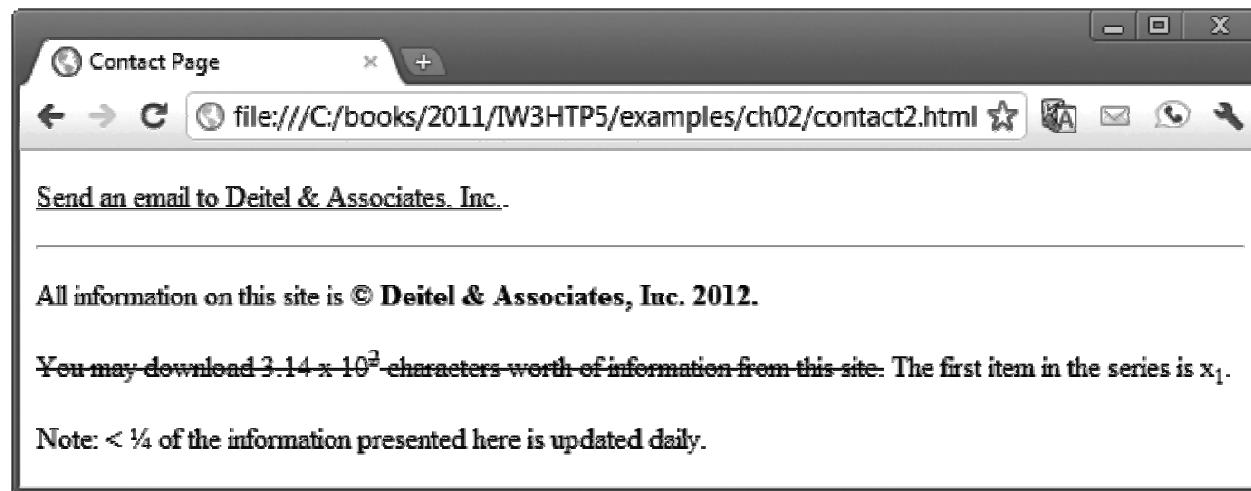
- ▶ Figure 2.9 demonstrates how to use special characters in an HTML5 document.
- ▶ For an extensive list of character entities, see

[www.w3.org/TR/REC-html40/sgml/entities.html](http://www.w3.org/TR/REC-html40/sgml/entities.html)

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.9: contact2.html -->
4  <!-- Inserting special characters. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Contact Page</title>
9      </head>
10
11     <body>
12         <p>
13             <a href = "mailto:deitel@deitel.com">Send an email to
14                 Deitel & Associates, Inc.</a>.
15         </p>
16
17         <hr> <!-- inserts a horizontal rule -->
18
19         <!-- special characters are entered -->
20         <!-- using the form &code; -->
21         <p>All information on this site is <strong>&copy;
22             Deitel & Associates, Inc. 2012.</strong> </p>
23
24         <!-- to strike through text use <del> element -->
```

**Fig. 2.9 |** Inserting special characters. (Part 1 of 2.)

```
25      <!-- to subscript text use <sub> element -->
26      <!-- to superscript text use <sup> element -->
27      <!-- these elements are nested inside other elements -->
28      <p><del>You may download  $3.14 \times 10^2$ </sup>
29          characters worth of information from this site.</del>
30          The first item in the series is  $x<sub>1</sub>$ .</p>
31      <p>Note: &lt;  $\frac{1}{4}$ ; of the information
32          presented here is updated daily.</p>
33  </body>
34 </html>
```



**Fig. 2.9 | Inserting special characters. (Part 2 of 2.)**

## 2.8 Special Characters and Horizontal Rules (Cont.)

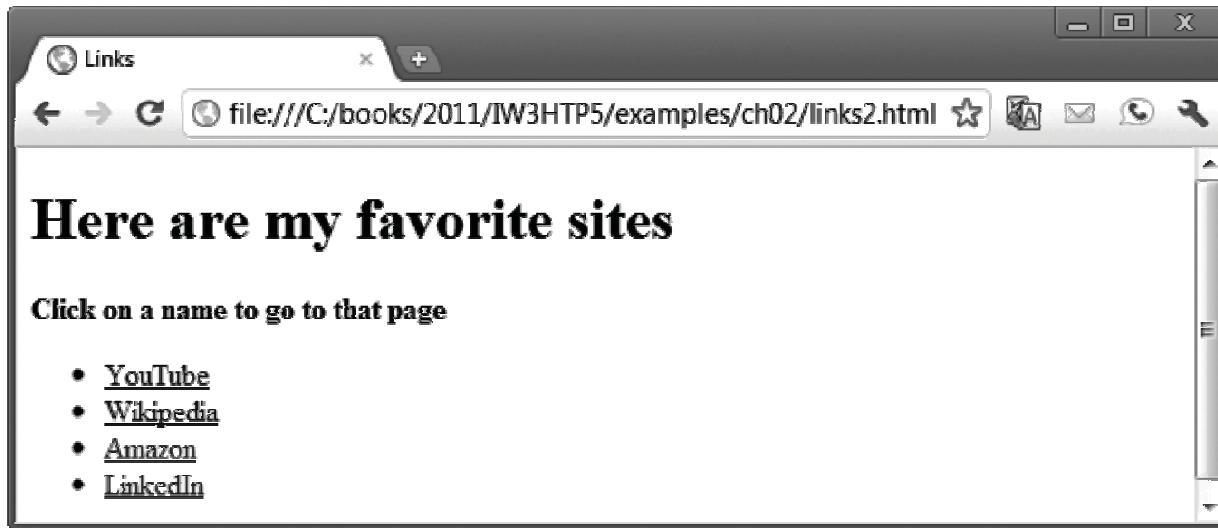
- ▶ A horizontal rule, indicated by the `<hr>` tag renders a horizontal line with extra space above and below it in most browsers.
- ▶ The horizontal rule element should be considered a legacy element and you should avoid using it.
- ▶ CSS can be used to add horizontal rules and other formatting to documents.
- ▶ Special characters can also be represented as **numeric character references**—decimal or hexadecimal (hex) values representing special characters.
  - For example, the & character is represented in decimal and hexadecimal notation as `&#38;` and `&x26;`, respectively.
- ▶ Hexadecimal numbers are discussed in Appendix E, Number Systems, which is available online at [www.deitel.com/books/iw3htp5/](http://www.deitel.com/books/iw3htp5/).

## 2.9 Lists

- ▶ Unordered list element `ul`
  - creates a list in which each item in the list begins with a bullet symbol (typically a disc)
  - Each entry is an `li` (list item) element. Most web browsers render these elements with a line break and a bullet symbol at the beginning of the line.

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.10: links2.html -->
4  <!-- Unordered list containing hyperlinks. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Links</title>
9      </head>
10
11     <body>
12         <h1>Here are my favorite sites</h1>
13         <p><strong>Click on a name to go to that page</strong></p>
14
15         <!-- create an unordered list -->
16         <ul>
17             <!-- the list contains four list items -->
18             <li><a href = "http://www.youtube.com">YouTube</a></li>
19             <li><a href = "http://www.wikipedia.org">Wikipedia</a></li>
20             <li><a href = "http://www.amazon.com">Amazon</a></li>
21             <li><a href = "http://www.linkedin.com">LinkedIn</a></li>
22         </ul>
23     </body>
24 </html>
```

**Fig. 2.10 | Unordered list containing hyperlinks. (Part 1 of 2.)**



**Fig. 2.10 | Unordered list containing hyperlinks. (Part 2 of 2.)**

## 2.9 Lists (Cont.)

### *Nested Lists*

- ▶ Lists may be *nested* to represent *hierarchical* relationships, as in a multi-level outline.
- ▶ Figure 2.11 demonstrates nested lists and ordered lists.
- ▶ The ordered-list element `ol` creates a list in which each item begins with a number.

```
1 <!DOCTYPE html>
2
3 <!-- Fig. 2.11: list.html -->
4 <!-- Nested lists and ordered lists. -->
5 <html>
6   <head>
7     <meta charset = "utf-8">
8     <title>Lists</title>
9   </head>
10
11  <body>
12    <h1>The Best Features of the Internet</h1>
13
```

**Fig. 2.11 | Nested lists and ordered lists. (Part 1 of 4.)**

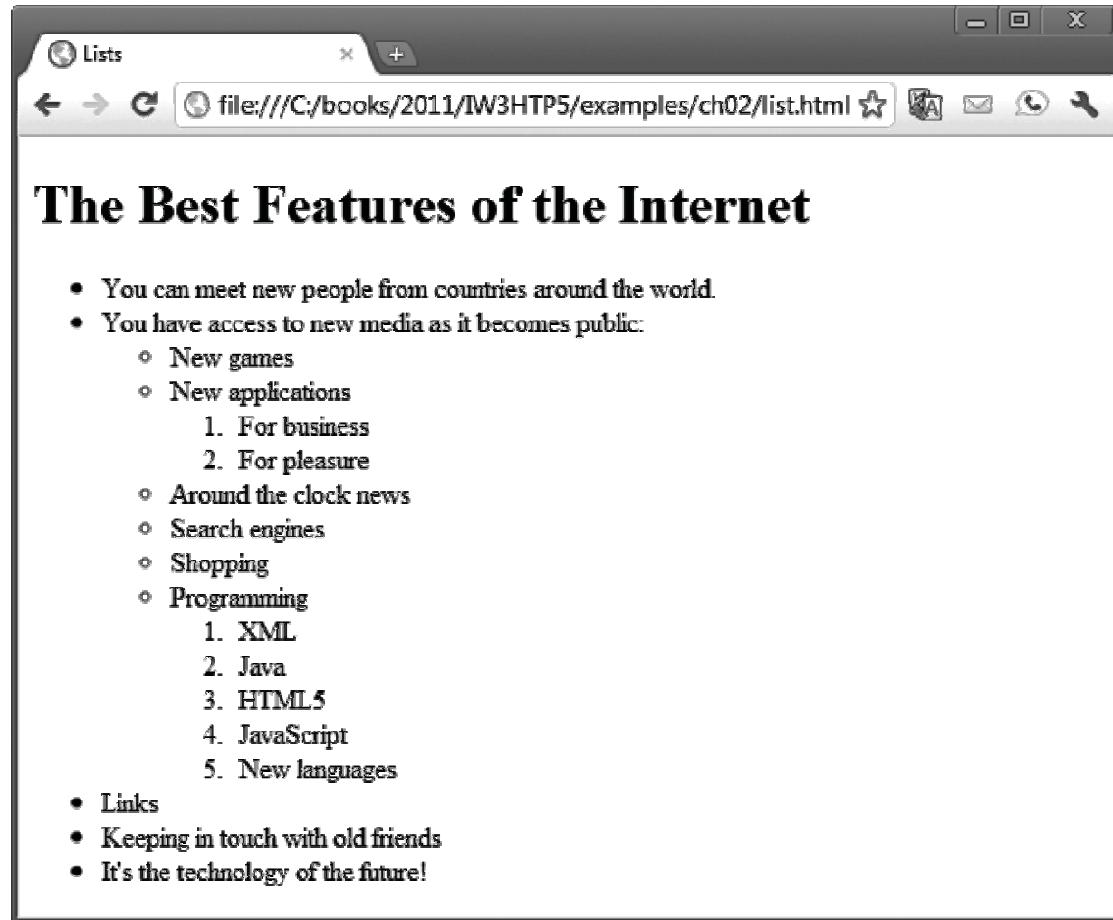
---

```
14      <!-- create an unordered list -->
15      <ul>
16          <li>You can meet new people from countries around
17              the world.</li>
18          <li>
19              You have access to new media as it becomes public:
20
21          <!-- this starts a nested unordered list, which uses a -->
22          <!-- different bullet. The list ends when you -->
23          <!-- close the <ul> tag. -->
24          <ul>
25              <li>New games</li>
26              <li>New applications
27
28                  <!-- nested ordered list -->
29                  <ol>
30                      <li>For business</li>
31                      <li>For pleasure</li>
32                  </ol>
33                  </li> <!-- ends line 27 new applications li-->
34
35                  <li>Around the clock news</li>
36                  <li>Search engines</li>
37                  <li>Shopping</li>
```

Fig. 2.11 | Nested lists and ordered lists. (Part 2 of 4.)

```
38      <li>Programming
39
40          <!-- another nested ordered list -->
41          <ol>
42              <li>XML</li>
43              <li>Java</li>
44              <li>HTML5</li>
45              <li>JavaScript</li>
46              <li>New languages</li>
47          </ol>
48          </li> <!-- ends programming li of line 38 -->
49      </ul> <!-- ends the nested list of line 24 -->
50  </li>
51
52      <li>Links</li>
53      <li>Keeping in touch with old friends</li>
54      <li>It's the technology of the future!</li>
55  </ul> <!-- ends the unordered list of line 15 -->
56 </body>
57 </html>
```

**Fig. 2.11 | Nested lists and ordered lists. (Part 3 of 4.)**



**Fig. 2.11 |** Nested lists and ordered lists. (Part 4 of 4.)

## 2.10 Tables

- Tables are frequently used to organize data into *rows* and *columns*.
- The `table` element defines an HTML5 table
- The `summary` attribute summarizes the table's contents and is used by speech devices to make the table more accessible to users with visual impairments.
- The `caption` element specifies a table's title.
- It's good practice to include a general description of a table's information in the `table` element's `summary` attribute—one of the many HTML5 features that make web pages more accessible to users with disabilities.
  - Speech devices use this attribute to make the table more accessible to users with visual impairments.

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.12: table1.html -->
4  <!-- Creating a basic table. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>A simple HTML5 table</title>
9      </head>
10
11     <body>
12         <!-- the <table> tag opens a table -->
13         <table border = "1">
14
15             <!-- the <caption> tag summarizes the table's -->
16             <!-- contents (this helps visually impaired people) -->
17             <caption><strong>Table of Fruits (1st column) and
18                 Their Prices (2nd column)</strong></caption>
19
```

**Fig. 2.12 |** Creating a basic table. (Part 1 of 4.)

```
20      <!-- the <thead> section appears first in the table -->
21      <!-- it formats the table header area -->
22      <thead>
23          <tr> <!-- <tr> inserts a table row -->
24              <th>Fruit</th> <!-- insert a heading cell -->
25              <th>Price</th>
26          </tr>
27      </thead>
28
29      <!-- the <tfoot> section appears last in the table -->
30      <!-- it formats the table footer -->
31      <tfoot>
32          <tr>
33              <th>Total</th>
34              <th>$3.75</th>
35          </tr>
36      </tfoot>
37
38      <!-- all table content is enclosed -->
39      <!-- within the <tbody> -->
40      <tbody>
41          <tr>
42              <td>Apple</td> <!-- insert a data cell -->
43              <td>$0.25</td>
44          </tr>
```

**Fig. 2.12 |** Creating a basic table. (Part 2 of 4.)

```
45      <tr>
46          <td>Orange</td>
47          <td>$0.50</td>
48      </tr>
49      <tr>
50          <td>Banana</td>
51          <td>$1.00</td>
52      </tr>
53      <tr>
54          <td>Pineapple</td>
55          <td>$2.00</td>
56      </tr>
57  </tbody>
58 </table>
59 </body>
60 </html>
```

**Fig. 2.12** | Creating a basic table. (Part 3 of 4.)

The screenshot shows a web browser window with the title "A simple HTML5 table". The page content is a table with the following structure:

Fruit	Price
Apple	\$0.25
Orange	\$0.50
Banana	\$1.00
Pineapple	\$2.00
<b>Total</b>	<b>\$3.75</b>

Annotations with leader lines point to specific parts of the table:

- "Table caption" points to the main title "Table of Fruits" and its subtitle "(1st column) and Their Prices (2nd column)".
- "Table header" points to the first row of the table, which contains the column headers "Fruit" and "Price".
- "Table body" points to the four rows of fruit data.
- "Table footer" points to the final row of the table, which contains the total values.
- "Table border" points to the thick black border surrounding the entire table structure.

**Fig. 2.12 |** Creating a basic table. (Part 4 of 4.)

## 2.10 Tables (Cont.)

- ▶ A table can be split into three distinct sections:
  - Head (`thead` element)
    - Table titles
    - Column headers
  - Body (`tbody` element)
    - Primary table data
  - Table Foot (`tfoot` element)
    - Calculation results
    - Footnotes
    - Above body section in the code, but displays at the bottom in the page

## 2.10 Tables (Cont.)

- ▶ **tr Element**
  - Defines individual table rows
  - Element **th**
    - Defines a header cell
- ▶ **Td Element**
  - Contains table data elements

## 2.10 Tables (Cont.)

### *Using rowspan and colspan with Tables*

- ▶ Figure 2.13 introduces two new attributes that allow you to build more complex tables.
- ▶ You can merge data cells with the rowspan and colspan attributes
  - The values of these attributes specify the number of rows or columns occupied by the cell.
  - Can be placed inside any data cell or table header cell.
- ▶ The br element is rendered as a line break in most browsers—any markup or text following a br element is rendered on the next line.
- ▶ Like the img element, br is an example of a void element.
- ▶ Like the hr element, br is considered a legacy formatting element that you should avoid using—in general, formatting should be specified using CSS.

```
1 <!DOCTYPE html>
2
3 <!-- Fig. 2.13: table2.html -->
4 <!-- Complex HTML5 table. -->
5 <html>
6   <head>
7     <meta charset = "utf-8">
8     <title>Tables</title>
9   </head>
10
11  <body>
12    <h1>Table Example: Spanning Rows and Columns</h1>
13
14    <table border = "1">
15      <caption>A more complex sample table</caption>
16
```

**Fig. 2.13 | Complex HTML5 table. (Part 1 of 4.)**

```
17 <thead>
18     <!-- rowspans and colspans merge the specified -->
19     <!-- number of cells vertically or horizontally -->
20     <tr>
21         <!-- merge two rows -->
22         <th rowspan = "2">
23             <img src = "camel.png" width = "205"
24                 height = "167" alt = "Picture of a camel">
25         </th>
26
27         <!-- merge four columns -->
28         <th colspan = "4">
29             <strong>Camelid comparison</strong><br>
30             Approximate as of 10/2011
31         </th>
32     </tr>
33     <tr>
34         <th># of humps</th>
35         <th>Indigenous region</th>
36         <th>Spits?</th>
37         <th>Produces wool?</th>
38     </tr>
39 </thead>
```

Fig. 2.13 | Complex HTML5 table. (Part 2 of 4.)

```
40      <tbody>
41          <tr>
42              <th>Camels (bactrian)</th>
43              <td>2</td>
44              <td>Africa/Asia</td>
45              <td>Yes</td>
46              <td>Yes</td>
47          </tr>
48          <tr>
49              <th>Llamas</th>
50              <td>1</td>
51              <td>Andes Mountains</td>
52              <td>Yes</td>
53              <td>Yes</td>
54          </tr>
55      </tbody>
56  </table>
57 </body>
58 </html>
```

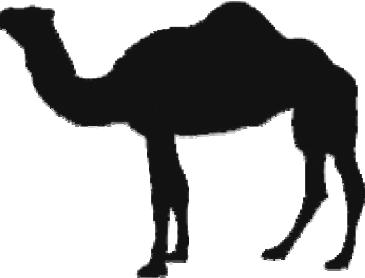
Fig. 2.13 | Complex HTML5 table. (Part 3 of 4.)

Tables

file:///C:/books/2011/IW3HTP5/examples/ch02/table2.html

## Table Example: Spanning Rows and Columns

A more complex sample table



Camelid comparison Approximate as of 6/2011				
	# of humps	Indigenous region	Spits?	Produces wool?
Camels (bactrian)	2	Africa/Asia	Yes	Yes
Llamas	1	Andes Mountains	Yes	Yes

Fig. 2.13 | Complex HTML5 table. (Part 4 of 4.)

## 2.11 Forms

- ▶ HTML5 provides forms for collecting information from users.
- ▶ Figure 2.14 is a simple form that sends data to the web server for processing.

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.14: form.html -->
4  <!-- Form with a text field and hidden fields. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Forms</title>
9      </head>
10
11     <body>
12         <h1>Feedback Form</h1>
13
14         <p>Please fill out this form to help
15             us improve our site.</p>
16
17         <!-- this tag starts the the form, gives the -->
18         <!-- method of sending information and the -->
19         <!-- Location of the form-processing script -->
20         <form method = "post" action = "http://www.deitel.com">
21             <!-- hidden inputs contain non-visual -->
22             <!-- information that will also be submitted -->
23             <input type = "hidden" name = "recipient"
24                 value = "deitel@deitel.com">
```

**Fig. 2.14** | Form with a text field and hidden fields. (Part I of 3.)

```
25      <input type = "hidden" name = "subject"
26          value = "Feedback Form">
27      <input type = "hidden" name = "redirect"
28          value = "main.html">
29
30      <!-- <input type = "text"> inserts a text field -->
31      <p><label>Name:
32          <input name = "name" type = "text" size = "25"
33              maxLength = "30">
34      </label></p>
35
36      <p>
37          <!-- input types "submit" and "reset" insert -->
38          <!-- buttons for submitting and clearing the -->
39          <!-- form's contents, respectively -->
40          <input type = "submit" value = "Submit">
41          <input type = "reset" value = "Clear">
42      </p>
43      </form>
44  </body>
45 </html>
```

**Fig. 2.14 |** Form with a text field and hidden fields. (Part 2 of 3.)

A screenshot of a web browser window titled "Forms". The address bar shows the URL "file:///C:/books/2011/IW3HTP5/examples/ch02/form.html". The main content area displays a "Feedback Form" with the following elements:

- Section Header:**

## Feedback Form
- Text:** Please fill out this form to help us improve our site.
- Text Input:** A text input field labeled "Name:" followed by a blank input box.
- Buttons:** Two buttons at the bottom left: "Submit" and "Clear".

**Fig. 2.14 |** Form with a text field and hidden fields. (Part 3 of 3.)

## 2.11 Forms (Cont.)

### *method Attribute of the form Element*

- ▶ A form is defined by a form element
  - Attribute method specifies how the form's data is sent to the web server.
  - Using method = "post" appends form data to the browser request, which contains the protocol (HTTP) and the requested resource's URL.
  - The other possible value, method = "get", appends the form data directly to the end of the URL of the script, where it's visible in the browser's Address field.
  - The action attribute of the form element specifies the script to which the form data will be sent

## 2.11 Forms (Cont.)

### *action Attribute of the form Element*

- The action attribute of the form element specifies the script to which the form data will be sent.
- Since we haven't introduced server-side programming yet, we set this attribute to `http://www.deitel.com` for now.
- input elements that specify data to provide to the script that processes the form (also called the form handler).
- An input's type is determined by its **type** attribute.

## 2.11 Forms (Cont.)

### *Hidden Inputs*

- ▶ Forms can contain visual and nonvisual components.
- ▶ Visual components include clickable buttons and other graphical user interface components with which users interact.
- ▶ Nonvisual components, called hidden inputs, store any data that you specify, such as e-mail addresses and HTML5 document file names that act as links.

## 2.11 Forms (Cont.)

### *text input Element*

- ▶ The `text input` inserts a text field into the form, which allows the user to input data.
- ▶ The `label` element provides users with information about the input element's purpose
- ▶ The `size` attribute specifies the number of characters visible in the text field.
- ▶ Optional attribute `maxLength` limits the number of characters input into a text field.

## 2.11 Forms (Cont.)

### *submit and reset input Elements*

- ▶ The **submit** input element is a button.
  - When the submit button is pressed, the form's data is sent to the location specified in the form's action attribute.
- ▶ The **value** attribute sets the text displayed on the button.
- ▶ The **reset** input element allows a user to reset all form elements to their default values.

## 2.11 Forms (Cont.)

### *Additional Form Elements*

- ▶ Figure 2.15 contains a form that solicits user feedback about a website.
- ▶ The **textarea** element inserts a *multiline text area* into the form.
- ▶ The number of rows is specified with the **rows** attribute, and the number of columns (i.e., characters per line) with the **cols** attribute.
- ▶ Default text can be specified in other **input** types, such as text fields, by using the **value** attribute.

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.15: form2.html -->
4  <!-- Form using a variety of components. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>More Forms</title>
9      </head>
10
11     <body>
12         <h1>Feedback Form</h1>
13         <p>Please fill out this form to help
14             us improve our site.</p>
15
16         <form method = "post" action = "http://www.deitel.com">
17
18             <input type = "hidden" name = "recipient"
19                 value = "deitel@deitel.com">
20             <input type = "hidden" name = "subject"
21                 value = "Feedback Form">
22             <input type = "hidden" name = "redirect"
23                 value = "main.html">
24
```

**Fig. 2.15 |** Form using a variety of components. (Part 1 of 6.)

```
25      <p><label>Name:<br>
26          <input name = "name" type = "text" size = "25">
27      </label></p>
28
29      <!-- <textarea> creates a multiline textbox -->
30      <p><label>Comments:<br>
31          <textarea name = "comments"
32              rows = "4" cols = "36">Enter comments here.</textarea>
33      </label></p>
34
35      <!-- <input type = "password"> inserts a -->
36      <!-- textbox whose display is masked with -->
37      <!-- asterisk characters -->
38      <p><label>E-mail Address:<br>
39          <input name = "email" type = "password" size = "25">
40      </label></p>
41
42      <p>
43          <strong>Things you liked:</strong><br>
44
45          <label>Site design<br>
46              <input name = "thingsliked" type = "checkbox"
47                  value = "Design"></label>
```

Fig. 2.15 | Form using a variety of components. (Part 2 of 6.)

```

48    <label>Links
49        <input name = "thingsliked" type = "checkbox"
50            value = "Links"></label>
51    <label>Ease of use
52        <input name = "thingsliked" type = "checkbox"
53            value = "Ease"></label>
54    <label>Images
55        <input name = "thingsliked" type = "checkbox"
56            value = "Images"></label>
57    <label>Source code
58        <input name = "thingsliked" type = "checkbox"
59            value = "Code"></label>
60    </p>
61
62    <!-- <input type = "radio"> creates a radio -->
63    <!-- button. The difference between radio buttons -->
64    <!-- and checkboxes is that only one radio button -->
65    <!-- in a group can be selected. -->
66    <p>
67        <strong>How did you get to our site?</strong><br>
68
69    <label>Search engine
70        <input name = "howtosite" type = "radio"
71            value = "search engine" checked></label>

```

**Fig. 2.15 |** Form using a variety of components. (Part 3 of 6.)

```
72      <label>Links from another site
73          <input name = "howtosite" type = "radio"
74              value = "link"></label>
75      <label>Deitel.com Web site
76          <input name = "howtosite" type = "radio"
77              value = "deitel.com"></label>
78      <label>Reference in a book
79          <input name = "howtosite" type = "radio"
80              value = "book"></label>
81      <label>Other
82          <input name = "howtosite" type = "radio"
83              value = "other"></label>
84  </p>
85
86  <p>
87      <label>Rate our site:
88
89          <!-- the <select> tag presents a drop-down -->
90          <!-- list with choices indicated by the -->
91          <!-- <option> tags -->
92      <select name = "rating">
93          <option selected>Amazing</option>
94          <option>10</option>
95          <option>9</option>
96          <option>8</option>
```

Fig. 2.15 | Form using a variety of components. (Part 4 of 6.)

```
97      <option>7</option>
98      <option>6</option>
99      <option>5</option>
100     <option>4</option>
101     <option>3</option>
102     <option>2</option>
103     <option>1</option>
104     <option>Awful</option>
105   </select>
106 </label>
107 </p>
108
109 <p>
110   <input type = "submit" value = "Submit">
111   <input type = "reset" value = "Clear">
112 </p>
113 </form>
114 </body>
115 </html>
```

**Fig. 2.15** | Form using a variety of components. (Part 5 of 6.)

**Feedback Form**

Please fill out this form to help us improve our site.

Name:

Comments:

E-mail Address:

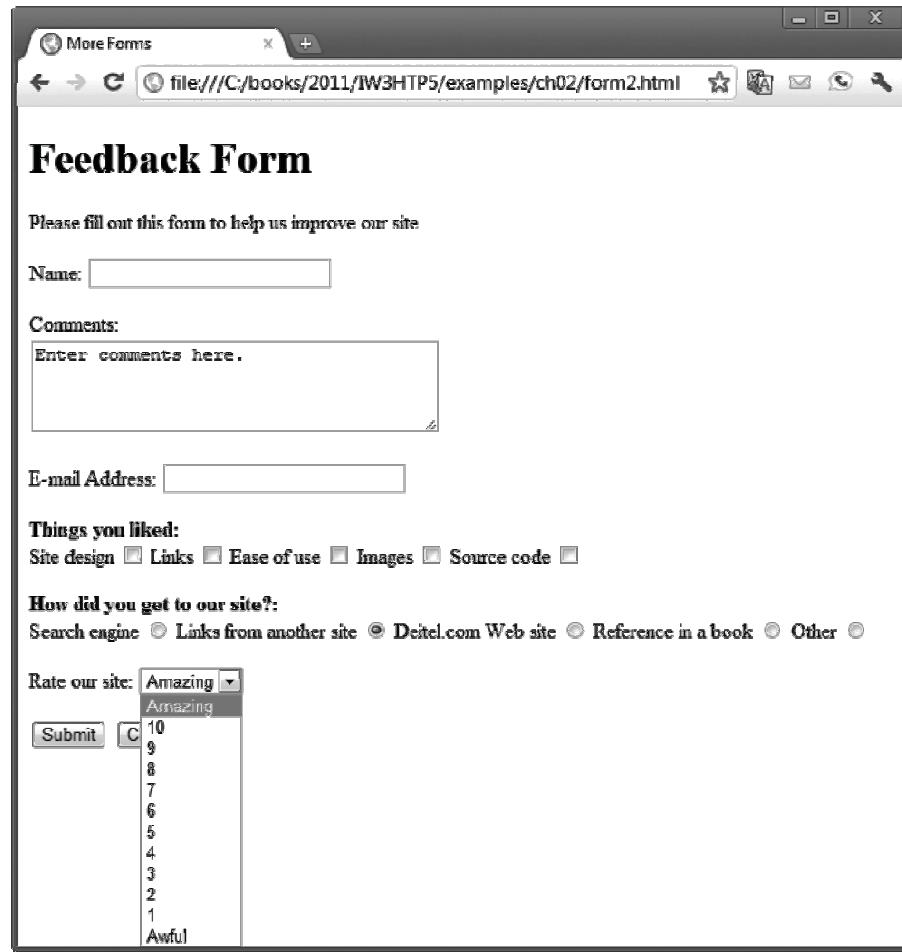
Things you liked:

Site design  Links  Ease of use  Images  Source code

How did you get to our site?:

Search engine  Links from another site  Deitel.com Web site  Reference in a book  Other

Rate our site:    
9  
8  
7  
6  
5  
4  
3  
2  
1  
Awful

A screenshot of a web browser window showing a feedback form titled "Feedback Form". The form includes fields for Name (text input), Comments (text area with placeholder "Enter comments here."), E-mail Address (text input), and a section for rating the site's features (checkboxes for Site design, Links, Ease of use, Images, and Source code). It also asks how the user got to the site (radio buttons for Search engine, Links from another site, Deitel.com Web site, Reference in a book, and Other). A "Rate our site" dropdown menu is open, showing options from "Amazing" down to "Awful" with numerical ratings 10 through 1. A "Submit" button is at the bottom left.

**Fig. 2.15 | Form using a variety of components. (Part 6 of 6.)**

## 2.11 Forms (Cont.)

- ▶ The password input inserts a password box into a form.
  - Allows users to enter sensitive information, such as credit card numbers and passwords, by “masking” the information input with another character, usually asterisks.
  - The actual value input is sent to the web server, not the asterisks that mask the input.

## 2.11 Forms (Cont.)

- ▶ The **checkbox** input element enables users to select and option.
  - When the checkbox is selected, a check mark appears in the checkbox . Otherwise, the checkbox is empty
  - checkboxes can be used individually and in groups. checkboxes that are part of the same group have the same name
- ▶ radio buttons are similar to checkboxes, except that only one radio button in a group can be selected at any time.
  - All radio buttons in a group have the same name attribute but different value attributes.
- ▶ The **select** input provides a drop-down list of items.
  - The name attribute identifies the drop-down list.
  - The **option** element adds items to the drop-down list.

## 2.12 Internal Linking

- ▶ The `a` tag can be used to link to another section of the same document by specifying the element's `id` as the link's `href`.
- ▶ To link internally to an element with its `id` attribute set, use the syntax `#id`.

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.16: internal.html -->
4  <!-- Internal Linking -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Internal Links</title>
9      </head>
10
11     <body>
12         <!-- id attribute creates an internal hyperlink destination -->
13         <h1 id = "features">The Best Features of the Internet</h1>
14
15         <!-- an internal link's address is "#id" -->
16         <p><a href = "#bugs">Go to <em>Favorite Bugs</em></a></p>
17
18         <ul>
19             <li>You can meet people from countries
20                 around the world.</li>
21             <li>You have access to new media as it becomes public:
22                 <ul>
23                     <li>New games</li>
```

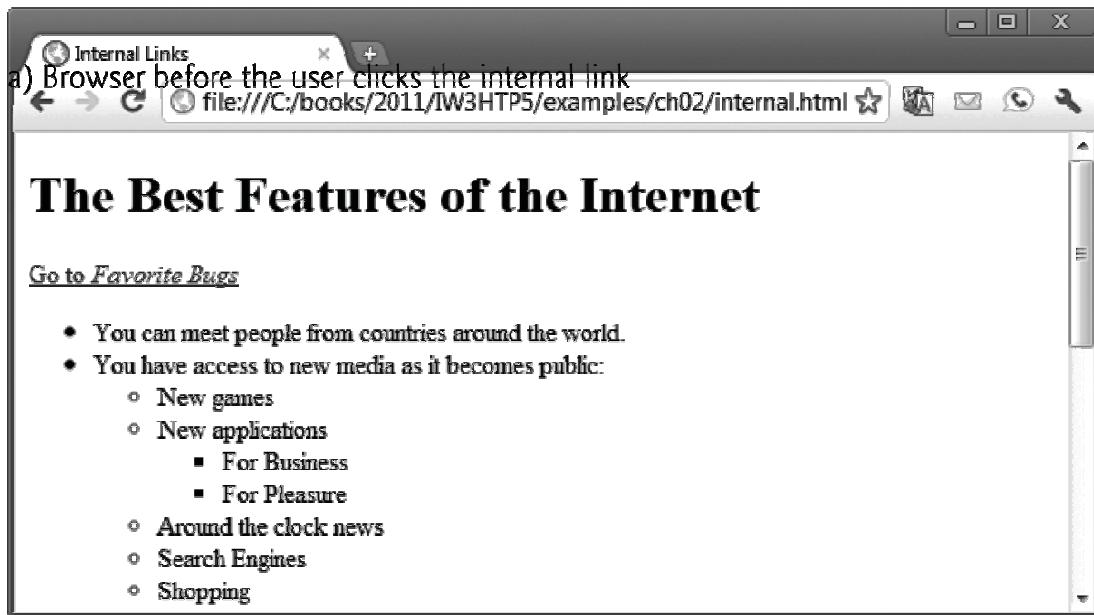
**Fig. 2.16** | Internal hyperlinks to make pages more navigable. (Part I of 5.)

```
24          <li>New applications
25            <ul>
26              <li>For Business</li>
27              <li>For Pleasure</li>
28            </ul>
29          </li>
30
31          <li>Around the clock news</li>
32          <li>Search Engines</li>
33          <li>Shopping</li>
34          <li>Programming
35            <ul>
36              <li>HTML5</li>
37              <li>Java</li>
38              <li>Dynamic HTML</li>
39              <li>Scripts</li>
40              <li>New Languages</li>
41            </ul>
42          </li>
43        </ul>
44      </li>
45
```

**Fig. 2.16 | Internal hyperlinks to make pages more navigable. (Part 2 of 5.)**

```
46      <li>Links</li>
47      <li>Keeping in touch with old friends</li>
48      <li>It is the technology of the future!</li>
49  </ul>
50
51  <!-- id attribute creates an internal hyperlink destination -->
52  <h1 id = "bugs">My 3 Favorite Bugs</h1>
53  <p>
54      <!-- internal hyperlink to features -->
55      <a href = "#features">Go to <em>Favorite Features</em></a>
56  </p>
57  <ol>
58      <li>Fire Fly</li>
59      <li>Gal Ant</li>
60      <li>Roman Tic</li>
61  </ol>
62  </body>
63  </html>
```

**Fig. 2.16 |** Internal hyperlinks to make pages more navigable. (Part 3 of 5.)



**Fig. 2.16 | Internal hyperlinks to make pages more navigable. (Part 4 of 5.)**



**Fig. 2.16 | Internal hyperlinks to make pages more navigable. (Part 5 of 5.)**

## 2.13 meta Elements

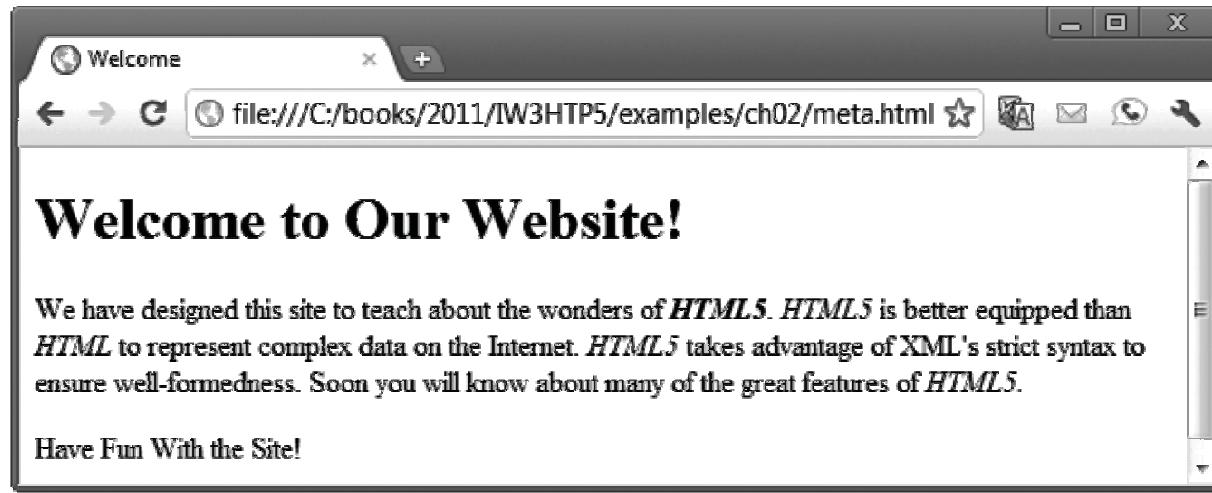
- ▶ One way that search engines catalog pages is by reading the meta element's contents.
  - The name attribute identifies the type of meta element
  - The content attribute
    - Of a keywords meta element: provides search engines with a list of words that describe a page, which are compared with words in search requests
    - Of a description meta element: provides a three-to four-line description of a site in sentence form, used by search engines to catalog your site. This text is sometimes displayed as part of the search result

```
1 <!DOCTYPE html>
2
3 <!-- Fig. 2.17: meta.html -->
4 <!-- meta elements provide keywords and a description of a page. -->
5 <html>
6   <head>
7     <meta charset = "utf-8">
8     <title>Welcome</title>
9
10    <!-- <meta> tags provide search engines with -->
11    <!-- information used to catalog a site      -->
12    <meta name = "keywords" content = "web page, design,
13        HTML5, tutorial, personal, help, index, form,
14        contact, feedback, list, links, deitel">
15    <meta name = "description" content = "This website will
16        help you learn the basics of HTML5 and web page design
17        through the use of interactive examples and
18        instruction.">
19   </head>
```

**Fig. 2.17** | meta elements provide keywords and a description of a page. (Part 1 of 3.)

```
20    <body>
21        <h1>Welcome to Our Website!</h1>
22
23        <p>We have designed this site to teach about the wonders
24        of <strong><em>HTML5</em></strong>. <em>HTML5</em> is
25        better equipped than <em>HTML</em> to represent complex
26        data on the Internet. <em>HTML5</em> takes advantage of
27        XML's strict syntax to ensure well-formedness. Soon you
28        will know about many of the great features of
29        <em>HTML5.</em></p>
30
31        <p>Have Fun With the Site!</p>
32    </body>
33 </html>
```

**Fig. 2.17** | meta elements provide keywords and a description of a page. (Part 2 of 3.)



**Fig. 2.17** | meta elements provide keywords and a description of a page. (Part 3 of 3.)



## **Software Engineering Observation 2.2**

---

meta elements are not visible to users. They must be placed inside the head section of your HTML5 document; otherwise they will not be read by search engines.