

# Chapter 4

## *A Hypertext Markup Language Primer*

# Table of Contents

- Part 1: Becoming Skilled at Computing
  - Chapter 1: Defining Information Technology
  - Chapter 2: Exploring the Human-Computer Interface
  - Chapter 3: The Basics of Networking
  - Chapter 4: A Hypertext Markup Language Primer
  - Chapter 5: Locating Information on the WWW
  - Chapter 6: An Introduction to Debugging
- Part 2: Algorithms and Digitizing Information
- Part 3: Data and Information
- Part 4: Problem Solving

# Learning Objectives

- Know the meaning of and use [hypertext terms](#)
- Use [HTML tags](#) to structure a document
- Use [HTML tag attributes](#)
- Use [Cascading Style Sheets](#) to style a Web page
- Use HTML tags to [link to other files](#)
- Explain the differences between [absolute and relative pathnames](#)
- Use HTML to encode [lists and tables](#) to structure a Web page



## 서울대학교 컴퓨터공학부

Seoul National University  
Dept. of Computer Science and Engineering

검색

소개 구성원 연구 입학 학사 및 교과 온라인 서비스 새소식 & 행사

### 컴퓨터공학부 복수전공학생 간담회 개최



컴퓨터공학을 복수전공하는 학생 4명과  
이건우 공대 학장님, 박근수 학부장님이  
간담회를 가졌습니다.

더 보기

### 공과대학으로 몰려가는 문과 대학생들



공과대학에서 소프트웨어(SW)와 컴퓨터  
공학 등을 복수전공하는 문과 계열 대학생  
이 크게 늘고 있습니다.

더 보기

### 신공학과 3층 해동학술정보 실 이용 안내



학부행정실 맞은편 해동학술정보실에 대  
한 학생 여러분의 많은 관심과 이용 바랍  
니다(자료실 운영: 월~금, 9:00 ~ 18:00).

더 보기

### 장병탁 교수 연구진, Imagination Machine 개발



만화영화를 보고 스스로 지식을 습득하는  
인공지능이 국내 연구진에 의해 개발되었  
습니다.

더 보기

### 2015년 2학기 수강신청 변경 안내

- 변경 기간: 9.1(화) ~ 9.7(월)(토·일 제외)  
- 웹 수강신청: <http://sugang.snu.ac.kr>

### 새소식

8/24 (월) 2015년 8월 우수학위논문상 수상자 안내  
6/8 (월) 2015 ACM SIGMOD 프로그래밍 콘테스트 3위 ...  
5/26 (화) 이광근 교수, [컴퓨터과학이 여는 세계] 도서 ...  
4/27 (월) 컴퓨터공학부 복수전공학생 간담회 개최  
4/17 (금) 공과대학으로 몰려가는 문과 대학생들  
4/16 (목) 빅데이터, 데이터마이닝 분야 신입교수님을 ...



더 보기

### 공지사항

8/27 (목) 2015학년도 2학기 '컴퓨터구조' 강의실 변경 ...  
8/27 (목) 2016년 2월 학부 졸업예정자 졸업(수료)신청 ...  
8/27 (목) 2016년 2월 학부 졸업예정자 졸업논문 지도 ...  
8/27 (목) 제10회 한국반도체산업협회 장학생 추천 의 ...  
8/27 (목) 2015학년도 2학기 전주식 교수님 강좌 개설 ...  
8/27 (목) '후기 학위수여식'행사일 자가용 운행 자제 ...



더 보기

```

1 <!DOCTYPE html>
2 <html lang="ko" dir="ltr">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <link rel="shortcut icon" href="http://cse.snu.ac.kr/sites/all/themes/dcse/favicon.ico" type="image/vnd.microsoft.icon" />
6 <link rel="alternate" type="application/rss+xml" title="공지사항" href="http://cse.snu.ac.kr/departement-notices.xml" />
7 <link rel="alternate" type="application/rss+xml" title="새소식" href="http://cse.snu.ac.kr/news.xml" />
8 <meta name="Generator" content="Drupal 7 (http://drupal.org)" />
9 <title>서울대학교 컴퓨터공학부</title>
10 <link type="text/css" rel="stylesheet" href="http://cse.snu.ac.kr/sites/default/files/css/css_fv_Aqxc-TFIMJfHYNW_3F7bTydhFUHTU2FA0YKgVQ6o.css" media="all" />
11 <link type="text/css" rel="stylesheet" href="http://cse.snu.ac.kr/sites/default/files/css/css_aPIVzOTIqeTpwJwee_6zfAIHJnVVCt2yrpnsRMKcOM.css" media="all" />
12 <link type="text/css" rel="stylesheet" href="http://cse.snu.ac.kr/sites/default/files/css/css_d2IMouZ708cp7ttPnfc9VKEEzyI5M66c1QzKhQNBQ.css" media="all" />
13 <link type="text/css" rel="stylesheet" href="http://cse.snu.ac.kr/sites/default/files/css/css_tK11-CCEkI16MT3trUnXLD2LJb9PLotY0skEgRQTfwQ.css" media="all" />
14 <link type="text/css" rel="stylesheet" href="http://cse.snu.ac.kr/sites/default/files/css/css_dh8VFLuxReMiP3WgqXQJDHyvUuwTp9Lu09YUG3nLbSY.css" media="print" />
15
16 <!--[if lte IE 8]>
17 <link type="text/css" rel="stylesheet" href="http://cse.snu.ac.kr/sites/default/files/css/css_RjdLjrH38JQ9GkB8D7hBv7QmLKjAZUuyNjD7CwdeyF8.css" media="all" />
18 <![endif]-->
19 <meta name="google-translate-customization" content="6a6d90d7527df5b2-d0c482b2e806f2cb-g4c306c95986efc1f-e"></meta>
20 </head>
21 <body class="html front not-logged-in no-sidebars page-home i18n-ko">
22 <div id="skip-link">
23 <a href="#navigation" class="element-invisible element-focusable">내비게이션으로 건너뛰기</a>
24 </div><!-- // -->
25 <div id="page-wrapper"><div id="page">
26
27 <div id="header">
28 <div id="header-top"><div class="section">
29 <div id="menu-wrapper">
30 <div id="login-block" class="block"><a href="/user/login">로그인</a></div>
31 <div class="region region-secondary">
32 <div id="block-locale-language" class="block block-locale first last odd">
33
34
35 <div class="content">
36 <ul class="language-switcher-locale-url"><li class="ko first active"><a href="/" class="language-link active">한국어</a></li>
37 <li class="en last"><a href="/en" class="language-link">English</a></li>
38 </ul></div>

```

# Web Pages

- **Hypertext Markup Language (HTML)** is the main language used to define how a Web page should look
- Web pages are created, stored, and sent in **HTML encoded form**
- **A browser** converts **HTML codes** to what we see on the screen
- Features like **background color, font, and layout** are specified in HTML
  - **HTML Tags** are for basic skeletons of documents
  - **CSS rule sets** are for styling and embellishing
- We use **HTML 5** which is the newest and best WWW standard language
- Embellish: 꾸미다, 장식하다

## 2 Free SWs for HTML Programming: Browser

- Firefox is a free open source browser
- Open source means that the program code is publicly available, and any programmer can contribute improvements to it
- Firefox is the browser referenced throughout this book
- It is available at [www.mozilla.com/en-US/firefox/all.html](http://www.mozilla.com/en-US/firefox/all.html)

## 2 Free SWs for HTML Programming: Text Editor

- Browsers want [Web pages written in ASCII characters only](#)
  - Think of ASCII as the normal keyboard characters with “nothing strange”
- [Modern-day word processors](#)
  - “what-you-see-is-what-you-get” or (WYSIWYG)
  - include many application-specific information in their files
  - This information confuses browsers
- We better use [a text editor to write HTML as text](#)
  - Text editors do not include the extra information
  - Text editors figure out what language you are writing in and add color to your code to make it easier to read
- Operating systems come with text editors installed
  - TextEdit can be found on the Mac
  - Notepad (메모장) comes with Windows
  - [TextWrangler](#) and [Notepad++](#) (by Don Ho) or [Notepad2](#) (by Florian Balmer) are better choices



# Required Tags

- HTML tags that are required for every Web page:

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8"/>
    <title>Required Tags</title>
  </head>
  <body>
    <p>Content</p>
  </body>
</html>
```

**Figure 4.1** Required tags for HTML5.

<head>

other information that applies to the whole page goes here...that will come later

</head>

<body>

the main content of the page goes here

</body>

starterPage.html

# Head Tag (문서에 적용되는 정보를 정의)

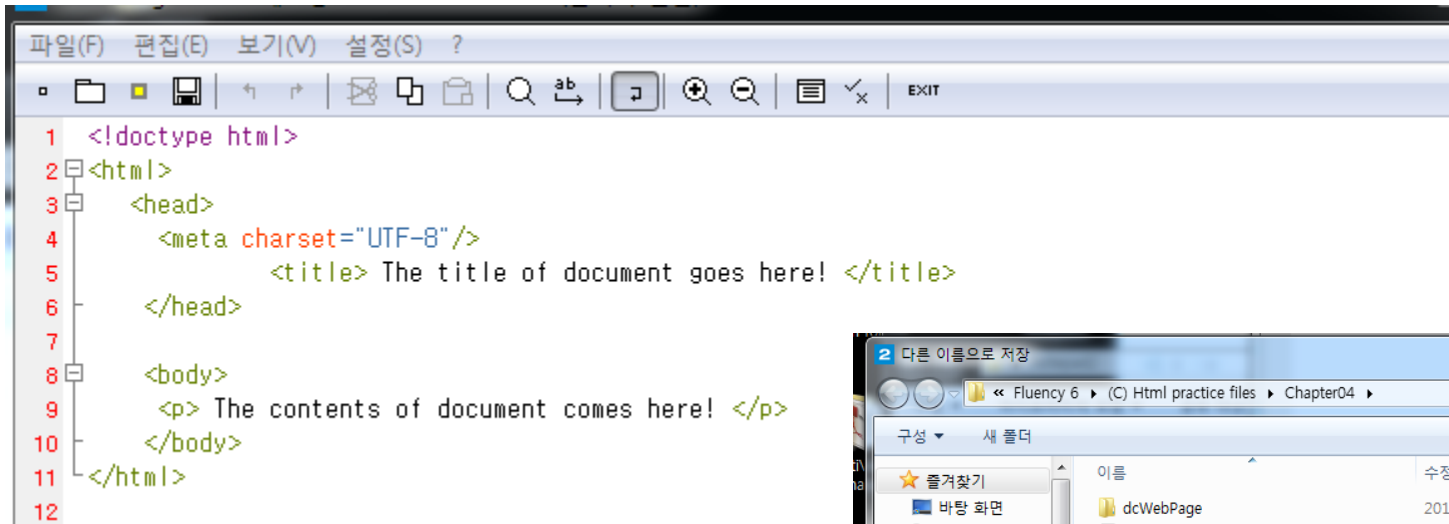
- Head Tag 내부에 포함되는 Tag들

Tag	Description
<title>	문서의 제목 을 정의
<script>	클라이언트측 스크립트를 정의
<style>	문서의 스타일 정보를 정의
<meta>	HTML 문서 에 대한 메타 데이터를 정의

- 그외에 <base> <link> <noscript> ....
- Tags can be looked up: [www.w3schools.com/tags/default.asp](http://www.w3schools.com/tags/default.asp)
- Fluency book: Appendix A HTML5 reference

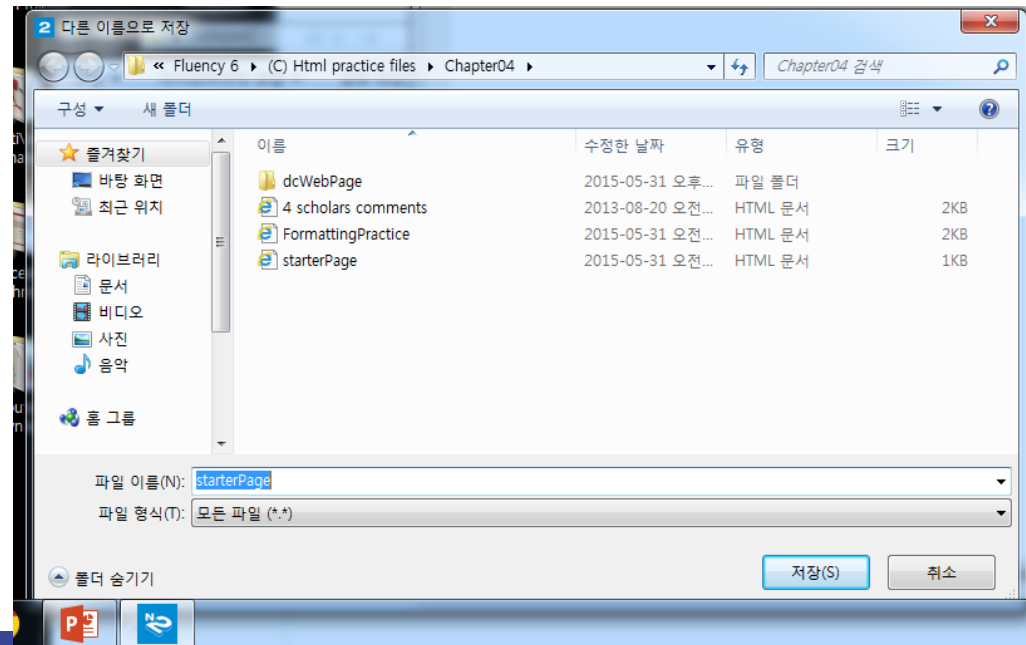
# Handling Html files [1/2]

## • Notepad2



The screenshot shows the Notepad2 application window. The menu bar includes '파일(F)', '편집(E)', '보기(V)', and '설정(S)'. The toolbar contains icons for file operations and editing. The text area displays the following HTML code:

```
1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="UTF-8"/>
5     <title> The title of document goes here! </title>
6   </head>
7
8   <body>
9     <p> The contents of document comes here! </p>
10  </body>
11 </html>
12
```



# Handling HTML files [2/2]

## •MS Explorer

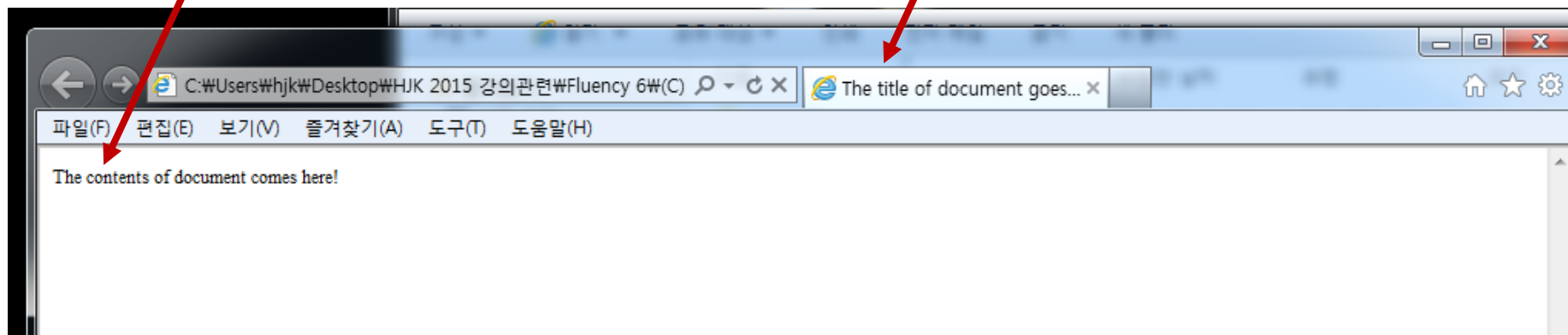


starterPage.html

Drag & Drop

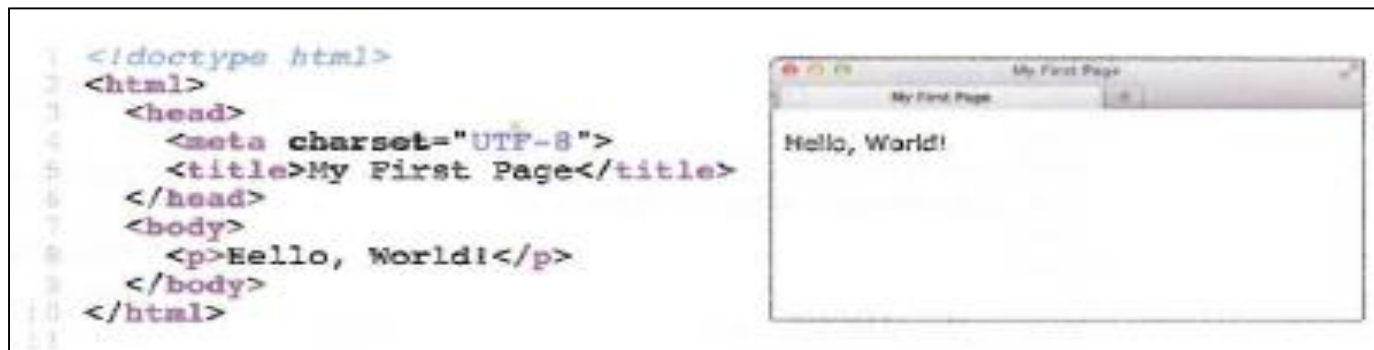
Title bar

Body of HTML document



# Hello, World!

- To produce your first HTML page, follow these instructions:
  - In your text editor, open a [New](#) document instance
  - Carefully type in your text
    - Change the title to something personal, like [My First Page](#)
    - Replace the paragraph contents with: `<p>Hello, World!</p>`
  - Save the file as [starterPage.html](#)
  - Open the file with the Firefox browser



**Figure 4.2** Text for a simple HTML5 Web page displayed as Page Source, and its display in Firefox.

# UTF-8

- Unicode Transformation Format (UTF)
- **UTF-8** is a character encoding capable of encoding all possible characters in Unicode
  - variable-length (→ uses 1 to 4 units of 8-bit code)
  - UTF-8의 1 unit는 ASCII 문자
- The dominant character encoding for the World Wide Web, accounting for 84.6% of all Web pages

# Open with Double-Click

- As HTML is written, files must be opened in two applications:
  - the text editor, to make changes
  - the browser, to see the changes made
- [Double-click](#) on the file to open it with the default application (your browser)

## Save This Page

- All HTML files have the same structure as the [starterPage.html](#) file you just created
- Use it as [a template](#) for future HTML coding
- Set up a new folder to keep your HTML files in
- Using your new page as a template ensures that all pages will have the correct form

# Formatting with Tags

- Tags in HTML must be **all HTML-reserved tags**: **No User-defined HTML Tag**
- Tags are words or abbreviations enclosed in **angle brackets**, **<** and **>**
- Many tags come in pairs and the second of the pair comes with a slash:

**<title>** Fluency 6 **</title>**

- HTML tags must be **lowercase**: **<TITLE>**, **<Title>**, and **<tITle>** are illegal
- The tag pair surrounds the text (**content**)
- **<title>** Serena Williams **</title>**
  - These tags can be read as “**this is where the title starts**” and “**this is where the title ends**”
  - **<title>** is referred to as **the start** or **open tag**
  - **</title>** is **the end** or **close tag**
  - The title appears on **the title bar** of the browser



# Tags for Bold and Italic

- HTML has tags:
  - for bold text, `<b>` and `</b>`
  - for italic text, `<i>` and `</i>`
  - for paragraphs, `<p>` and `</p>`
- More than one kind of formatting can be used at a time:  
`<p><b><i> Veni, Vidi, Vici! </i></b></p>` produces *Veni, Vidi, Vici!*
- It doesn't matter in which order you start the tags: italic follows bold, or bold follows italic ➔ You get the same result
- The rule is to make sure the tags “nest” correctly...they should mirror each other
- All the tags between a starting tag and its ending tag should be matched

# Singleton Tags

- A few tags are not paired
- They do not have a matching ending tag
- For those tags, the closing angle bracket > of the singleton tag is replaced by />
  - The horizontal rule <hr /> tage produces a horizontal line after a paragraph
  - The break <br /> tag continues the text on the next line

## More Formatting Tags

- Documents can be formatted in many ways
- Each formatting feature requires its own tag
- Programmers and Web designers need to remember a few common tags
- Uncommon tags can be looked up: [www.w3schools.com/tags/default.asp](http://www.w3schools.com/tags/default.asp)

# HTML Tags

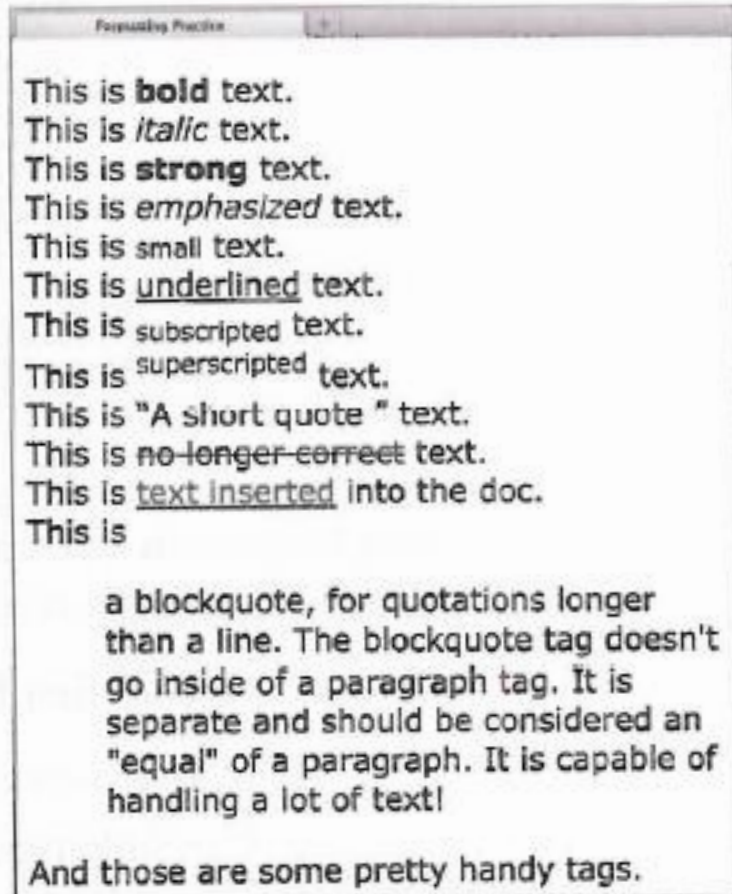
<!-->	<!DOCTYPE>	<a>	<abbr>	<acronym>	<address>
<applet>	<area>	<article>	<aside>	<audio>	<b>
<base>	<basefont>	<bdi>	<bdo>	<big>	<blockquote>
<body>	 	<button>	<canvas>	<caption>	<center>
<cite>	<code>	<col>	<colgroup>	<datalist>	<dd>
<del>	<details>	<dfn>	<dialog>	<dir>	<div>
<dl>	<dt>	<em>	<embed>	<fieldset>	<figcaption>
...					

- Over 118 tags! (Including HTML5)
  - Each tag has 0 to 30 attributes
  - <http://www.w3schools.com/html/default.asp>

```

<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Formatting Practice</title>
</head>
<body>
  <p>
    This is <b>bold</b> text.      <br>
    This is <i>italic</i> text.    <br>
    This is <strong>strong</strong> text. <br>
    This is <em>emphasized</em> text. <br>
    This is <small>small</small> text. <br>
    This is <u>underlined</u> text. <br>
    This is <sub>subscripted</sub> text. <br>
    This is <sup>superscripted</sup> text. <br>
    This is <q>A short quote </q> text. <br>
    This is <s>no longer correct</s> text. <br>
    This is <ins>text inserted</ins> into the doc. <br>
    This is
  </p>
  <blockquote>
    a blockquote, for quotations longer than a line.
    The blockquote tag doesn't go inside of a
    paragraph tag. It is separate and should be
    considered an "equal" of a paragraph. It is
    capable of handling a lot of text!
  </blockquote>
  <p>And those are some pretty handy tags.</p>
</body>
</html>

```



**Figure 4.3** A practice Web page for various formatting tags, and its display. Experiment with resizing the browser window to see how the `<blockquote>` text changes.

# HTML Time Line

- **October 1991:** [HTML Tags](#) an informal CERN document listing 18 HTML tags, was first mentioned in public
- **June 1993:** [Hypertext Markup Language \(HTML 1.0\)](#) was published by the IIR Working Group as an Internet-Draft. Selt was 1 months later, it was replaced by [HTML 2.0](#)
- **April 1995:** [HTML 3.0](#) was proposed as a standard to the IETF, but the proposal expired five months later (28 September 1995) without further action
- **December 1997:** [HTML 4.0](#) published as a W3C Recommendation
- **January 2000:** [XHTML 1.0](#) is published, as a W3C Recommendation, [XHTML](#) is a separate language that began as a reformulation of HTML 4.0 using XML 1.0. It is no longer being developed as a separate standard.
- **January 2008:** [HTML5](#) was published as working draft by the W3C



# WHAT IS ? [1/3]

- 초창기의 HTML (HTML 1.0, 1993)

- 인터넷을 통해 문서 형태의 정적인 정보를 공유하는 것이 주된 목적
- 서로 연결된 단순한 웹 문서들의 집합체 수준
- 네트워크와 서버속도의 제약으로 인해 텍스트와 이미지 정도가 표현 가능

- HTML2 ➔ HTML3 ➔ HTML4 (1997)

- 텍스트와 이미지 외에 비디오와 음성의 표현
- 미려한 차트, 메시징, 지리 정보, 푸시 등 확장된 서비스
- 플랫폼 다양성에 대한 지원과 멀티미디어·보안 등 확장된 서비스 요구에 대응어려움
- 비표준 기술인 Active X, Flash, Silverlight, Java FX 등과 이에 기반한 3rd 파티 플러그인
- 수많은 플러그인벤더 종속성 심화, 브라우저 간 호환성 제약, 보안 취약성, 성능, 서비스 접근성 보장 및 사용자 편의성 등에서 많은 문제

# WHAT IS ? [2/3]

## • 차세대 웹 표준인 HTML5 (2008년)

- ‘웹 콘텐츠’보다는 ‘웹 애플리케이션’을 주된 목표 대상으로 하는 기술
- 최신 멀티미디어 콘텐츠와 복잡한 기능을 포함하는 “표준화된 웹 프로그래밍 언어”
- 웹 문서의 구조를 정의하는 ‘HTML5 마크업’ 영역 → 전체 문서의 구조와 의미(Semantics)를 명확히 하여 디자인과 기능 영역의 독립성을 확보
- 문서의 디자인과 스타일을 표현하기 위한 ‘CSS3’ 영역 → 서로 다른 브라우저 상에서도 일관된 표현이 가능하며, 효과적이고 편리하게 표현 방식을 변경
- 다양한 동적·상호 작용 기능을 제공하기 위한 ‘JavaScript API 확장’ 영역 → 과거 외부 플러그인 확장들을 통해서만 제공이 가능하던 오디오, 비디오 처리 등의 기능을 제약 없이 웹 브라우저가 표준화된 방식으로 제공

- 웹 애플리케이션 장점
  - ‘배포와 설치가 필요 없어 유지 관리가 용이하고, 새로운 시도를 하는 것이 간단함’
  - ‘Open API를 통한 매시업 기반의 재사용이 가능함’
- 웹이 가지는 태생적인 약점
  - ‘브라우저에 좌우되는 성능’
  - ‘대용량 처리의 한계’
  - ‘디바이스 H/W에 대한 직접적인 제어 불가’
  - ‘로컬 저장 정보의 활용 불가’
- HTML5는 웹 상에서 제공하는 디바이스에 대한 직접 제어, 로컬 저장 정보의 활용, 멀티스레딩 및 오프라인 동작 지원과 같은 핵심적인 기능



# Headings in HTML

- Documents can have **headings and subheadings**
- HTML provides several levels of heading tags:

`<h1>` and `</h1>` is for level one

`<h2>` and `</h2>` is for level two

...

`<h6>` and `</h6>` is for level six

- Headings display content on a new line
- Headings are bold and get less “strong”

(smaller and perhaps not so bold) as the level number increases.

**Country: USA**

**State: Hawaii**

**County: Hawai'i**

**City: Hilo**

**Neighborhood: Waiakea**

**Street: Ululani**

Standard text size

# HTML Format vs. Display Format

- HTML source code tells the browser how to produce the formatted page based on **the meanings of the tags, not on how the source instructions look**
- Even so, HTML is written in a structured format to make it easier for people to understand
- **Indenting** is frequently used to emphasize the tags' meanings

```
<h1>Country: USA</h1>  
  <h2>State: Hawaii</h2>  
    <h3>County: Hawai'i</h3>  
      .
```

Let's make a simple HTML for the previous document in the page 23!

# White Space [1/2]

- Spaces that have been inserted for readability are called white space
- White space is created with spaces, tabs, and new lines (return or enter)
- HTML ignores white space
- The browser turns a sequence of white space characters into a single space
- Preformatted information contained within `<pre>` and `</pre>` is displayed exactly as it appears (white space and all)
- The width of a line of text is determined by the width of the browser window
  - A narrower or wider browser window makes the lines break in different places

# White Space [2/2]

<p> <b> Xeno's Paradox: </b>

Achilles and a turtle were to run a race. Achilles could run twice as fast as the turtle. The turtle, being a slower runner, got a 10 meter head start, whereupon Achilles started and ran the 10 meter distance. At that moment the turtle was 5 meters farther. When Achilles had run that distance the turtle had gone another 2.5 meters, and so forth. Paradoxically, the turtle always remained ahead. </p>

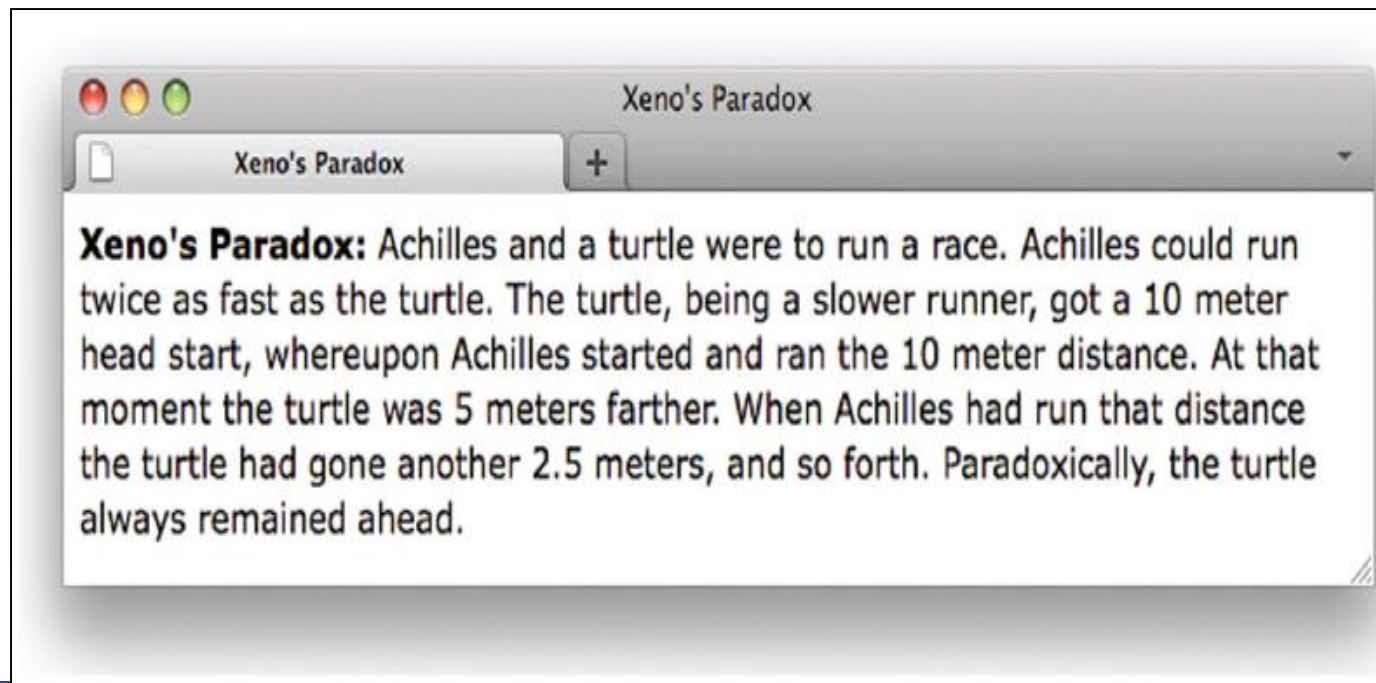



Table 4.1 Basic tags for HTML

Start Tag	End Tag	Meaning	Required
<!doctype html>	</html>	HTML document; first and last tags	✓
<title>	</title>	Title bar text; describes page	✓
<head>	</head>	Preliminary material; e.g., title at start of page	✓
<body>	< /body>	The main part of the page	✓
<p>	</p>	Paragraph	
<hr/>		Line (horizontal rule)	
<h1>...<h6>	</h1>...</h6>	Headings, six levels	
<b>	</b>	Bold	
<i>	< /i>	Italic	
<a href="fn">	</a>	Anchor reference, <i>fn</i> must be a pathname to an HTML file	
		Image source reference, <i>fn</i> must be a pathname to .jpg or .gif file	
 		Break, continue text on a new line	

위의 table에 있는 Tag들을 적어도 1번이상 사용하여 Web Page를 Programming 하기!

# The Escape Symbol (&)

- What if the Web page had to show a math relationship:  $0 < p > r$
- The browser might misinterpret `<p>` as a paragraph tag
- Using angle brackets as text is prohibited
- To show angle brackets, use an **escape symbol (&)**, followed by an **abbreviation**, followed by a **semicolon**



This is not an angle bracket

<p><code>&amp;lt;</code> displays as <code>&lt;</code> <code>&amp;gt;</code> displays as <code>&gt;</code> <code>&amp;amp;</code> displays as <code>&amp;</code></p>
--

- $0 \&lt; p \&gt; r$

# Accent Marks in HTML

- Letters with accent marks also use the escape symbol (&)
- General form is: `&ampersand`, followed by the letter, followed by the name of the accent mark, followed by a semicolon
- The case of the letter is important!
  - `&eacute`; displays as é
  - `&Egrave`; displays as È

&eacute

**Table 4.2** Specifying accent marks for Western European language

Symbol	Text	Number	Symbol	Text	Number	Symbol	Text	Number
à	&agrave;	&#224;	ê	&ecirc;	&#234;	ô	&ocirc;	&#244;
ä	&auml;	&#228;	î	&icirc;	&#238;	ù	&ugrave;	&#249;
é	&eacute;	&#233;	ó	&oacute;	&#243;	ã	&atilde;	&#227;
í	&iacute;	&#237;	ø	&oslash;	&#248;	è	&egrave;	&#232;
ò	&ograve;	&#242;	ü	&uuml;	&#252;	ì	&igrave;	&#236;
ö	&ouml;	&#246;	â	&acirc;	&#226;	ñ	&ntilde;	&#241;
û	&ucirc;	&#251;	ç	&ccedil;	&#231;	õ	&otilde;	&#245;
á	&aacute;	&#225;	ë	&euml;	&#235;	ú	&uacute;	&#250;
å	&aring;	&#229;	ï	&iuml;	&#239;			

*Note:* For an accent mark on an uppercase letter, make the letter following the & uppercase or find the number for the uppercase letter at [www.w3schools.com/tags/ref\\_entities.asp](http://www.w3schools.com/tags/ref_entities.asp).



# Sample HTML [1/2]

Notice the following:

- The title is shown on **the title bar** of the browser window
- The statement of Russell's Paradox is in bold
- The HTML source paragraphs are indented more than the <h2> heading lines to make them more readable
- The line between the two paragraphs crosses the width of the browser window
- An **acute accent** is used in Magritte's first name
- The **French phrase** from the painting is in italics
- The word **picture** is in italics for emphasis

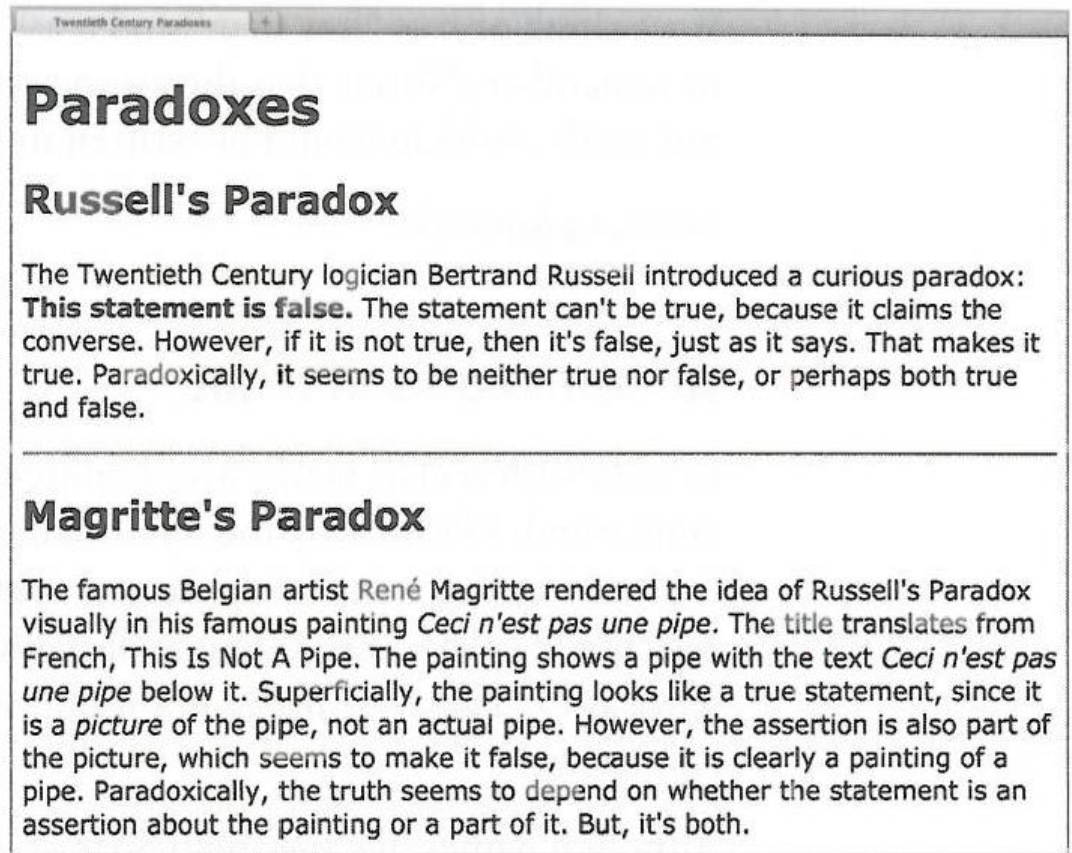
```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title> Twentieth Century Paradoxes </title>
  </head>
  <body>
    <h1>Paradoxes</h1>
    <h2>Russell's Paradox</h2>
    <p>The Twentieth Century logician Bertrand Russell
    introduced a curious paradox: <b>This statement is
    false.</b> The statement can't be true, because it
    claims the converse. However, if it is not true, then it's
    false, just as it says. That makes it true. Paradoxically,
    it seems to be neither true nor false, or perhaps both
    true and false.</p>
    <hr/>
    <h2>Magritte's Paradox</h2>
    <p> The famous Belgian artist Ren<b>eacute</b> Magritte
    rendered the idea of Russell's Paradox visually in his
    famous painting <i>Ceci n'est pas une pipe</i>. The
    title translates from French, This Is Not A Pipe. The
    painting shows a pipe with the text <i>Ceci n'est pas
    une pipe</i> below it. Superficially, the painting looks
    like a true statement, since it is a <i>picture</i> of
    the pipe, not an actual pipe. However, the assertion is
    also part of the picture, which seems to make it false,
    because it is clearly a painting of a pipe. Paradoxically,
    the truth seems to depend on whether the statement is
    an assertion about the painting or a part of it. But, it's
    both. </p>
  </body>
</html>
```

21CenturyParadox.html



# Sample HTML [2/2]

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title> Twentieth Century Paradoxes </title>
</head>
<body>
  <h1>Paradoxes </h1>
  <h2>Russell's Paradox</h2>
  <p>The Twentieth Century logician Bertrand Russell introduced a curious paradox: <b>This statement is false.</b> The statement can't be true, because it claims the converse. However, if it is not true, then it's false, just as it says. That makes it true. Paradoxically, it seems to be neither true nor false, or perhaps both true and false.</p>
  <hr/>
  <h2>Magritte's Paradox</h2>
  <p>The famous Belgian artist René Magritte rendered the idea of Russell's Paradox visually in his famous painting <i>Ceci n'est pas une pipe</i>. The title translates from French, This Is Not A Pipe. The painting shows a pipe with the text <i>Ceci n'est pas une pipe</i> below it. Superficially, the painting looks like a true statement, since it is a <i>picture</i> of the pipe, not an actual pipe. However, the assertion is also part of the picture, which seems to make it false, because it is clearly a painting of a pipe. Paradoxically, the truth seems to depend on whether the statement is an assertion about the painting or a part of it. But, it's both.</p>
</body>
</html>
```



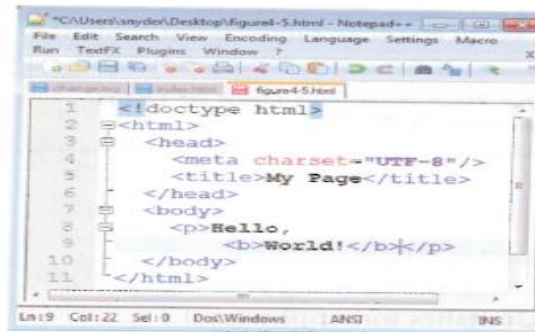
**Figure 4.4** The HTML for a simple page and the result (inset).

# Compose and Check

- A productive way to work is to keep two windows open:

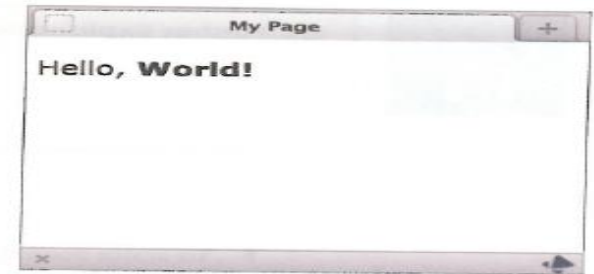
your text editor  $\longleftrightarrow$  your browser

- After writing a few HTML formatting tags, save the file
- Check the result in the browser by a **Reload or Refresh of the source**
- Repeat

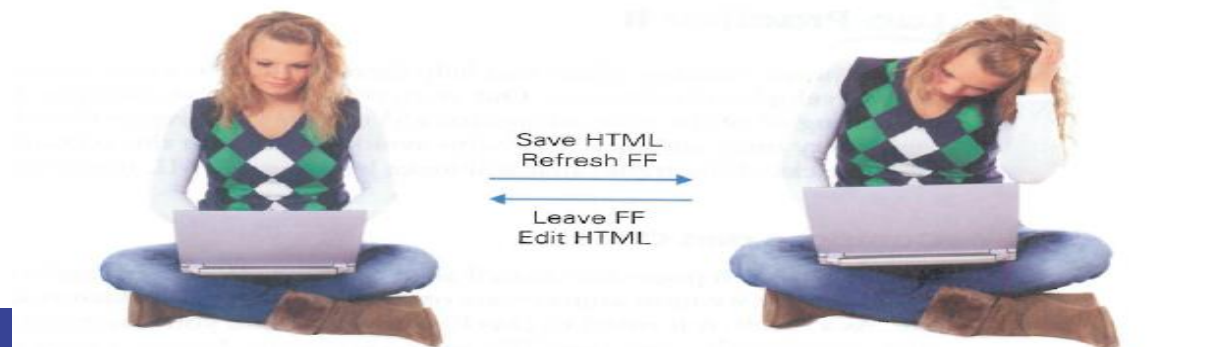


```
1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="UTF-8"/>
5     <title>My Page</title>
6   </head>
7   <body>
8     <p>Hello,
9       <b>world!</b></p>
10  </body>
11 </html>
```

COMPOSE

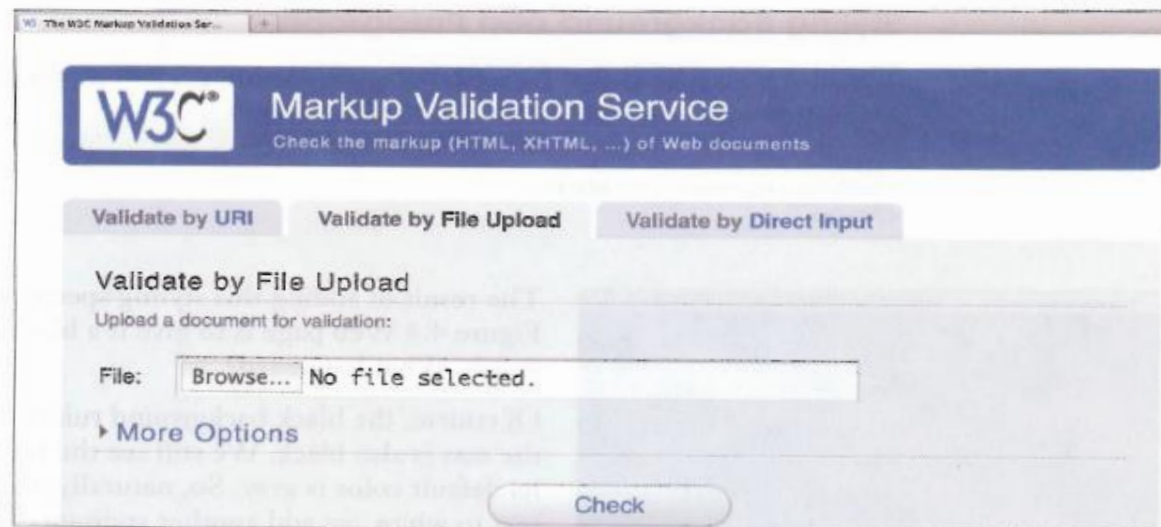


CHECK



# Markup Validation Service

- Another way to limit the mistakes you make is to have it automatically validated: [validator.w3.org/#validate\\_by\\_load](https://validator.w3.org/#validate_by_load)
- This service checks to make sure [your HTML is correct](#)
- If it is wrong, the service tells you [where the mistakes are](#) and [what's not proper](#)
- If [a green banner](#) comes back: the HTML checks out
- If [a red banner](#) comes back, it will have a list of errors with it and an explanation of what's wrong.

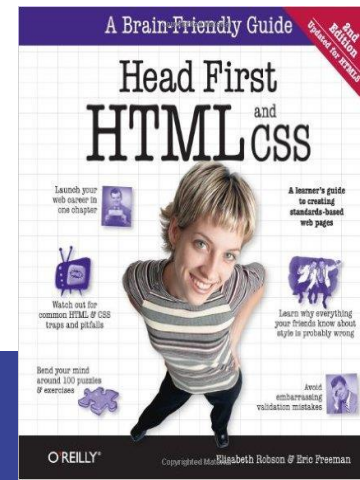


**Figure 4.6** The W3C Markup Validation Service page; browse for your HTML file and then click *Check*.

# Cascading Style Sheets (CSS)

- The paradox page in [Figure 4.4](#) is informative, but so plain and uninviting.
- We want “rockin” Web pages in [Figure 4.7](#) with help of [design styling](#).
- CSS are responsible for much of the design of Web pages
- [CSS1](#) (Level 1): W3C recommendation in [December 1996](#).
  - introduce a simple visual formatting model for all the HTML tags.
- [CSS2](#) (Level 2): W3C recommendation in [May 1998](#) and [builds on CSS1](#)
  - support for media-specific style sheets e.g. printers and aural devices, downloadable fonts
- [CSS3](#) (Level 3): The earliest CSS 3 drafts were published in [June 1999](#)

Rockin → Awesome, 잘하고있는, 잘된





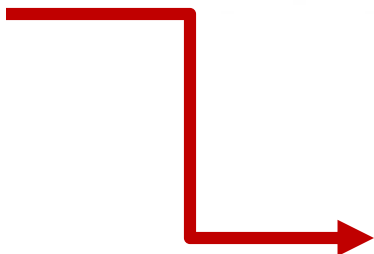
# Paradoxes

## Russell's Paradox

The Twentieth Century logician Bertrand Russell introduced a curious paradox: **This statement is false.** The statement can't be true, because it claims the converse. However, if it is not true, then it's false, just as it is true. Paradoxically, it seems to be neither true nor false, or and false.

## Magritte's Paradox

The famous Belgian artist René Magritte rendered the idea visually in his famous painting *Ceci n'est pas une pipe*. The French, This Is Not A Pipe. The painting shows a pipe with the text *une pipe* below it. Superficially, the painting looks like a true statement, since it is a *picture* of the pipe, not an actual pipe. However, the assertion is also part of the picture, which seems to make it false, because it is clearly a painting of a pipe. Paradoxically, the truth seems to depend on whether the assertion about the painting or a part of it. But, it's both.



Twentieth Century Paradoxes


## Paradoxes

### Russell's Paradox

The Twentieth Century logician Bertrand **Russell** introduced a curious paradox: **This statement is false.** The statement can't be true, because it claims the converse. However, if it is not true, then it's false, just as it says. That makes it true. Paradoxically, it seems to be neither true nor false, or perhaps both true and false.

### Magritte's Paradox

The famous Belgian artist René **Magritte** rendered the idea of Russell's Paradox visually in his famous painting *Ceci n'est pas une pipe*. The title translates from French, This Is Not A Pipe. The painting shows a pipe with the text *Ceci n'est pas une pipe* below it. Superficially, the painting looks like a true statement, since it is a *picture* of the pipe, not an actual pipe. However, the assertion is also part of the picture, which seems to make it false, because it is clearly a painting of a pipe. Paradoxically, the truth seems to depend on whether the statement is an assertion about the painting or a part of it. But, it's both.



# CSS Syntax

A CSS rule set consists of a selector and a declaration block:



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a property name and a value, separated by a colon.

## CSS Example

A CSS declaration always ends with a semicolon, and declaration groups are surrounded by curly braces:

```
p {color:red;text-align:center;}
```

To make the CSS code more readable, you can put one declaration on each line.

In the following example all <p> elements will be center-aligned, with a red text color:

### Example

```
p {  
  color: red;  
  text-align: center;  
}
```

# CSS Properties

- Like in HTML, there are numerous elements in selecting CSS Properties

Element	Properties
Text and Fonts	font, font-family, font-size, font-weight, font-style ...
Positioning	position, top, right, bottom, left, clip, z-index, float ...
List	list-style, list-style-type, list-style-image, list-style-position
Table	table-layout, border-collapse, border-spacing, empty-cells ...
Box Model	padding, border, outline, margin, width, height ...
Paged Media	page-break-before, page-break-after, orphans, widows ...
Whole Structure	background, background-attachment, background-color ...

- <http://htmldog.com/reference/cssproperties>

# Properties of CSS Rule Sets

HTML

CSS

JAVASCRIPT

[CSS HOME](#)[CSS Introduction](#)[CSS Syntax](#)[CSS Selectors](#)[CSS How To](#)[CSS Backgrounds](#)[CSS Text](#)[CSS Fonts](#)[CSS Links](#)[CSS Lists](#)[CSS Tables](#)[CSS Box Model](#)[CSS Border](#)[CSS Outline](#)[CSS Margin](#)[CSS Padding](#)[CSS Dimension](#)[CSS Display](#)[CSS Max-width](#)[CSS Position](#)[CSS Float](#)[CSS Inline-block](#)[CSS Align](#)[CSS Combinators](#)[CSS Pseudo-class](#)[CSS Pseudo-element](#)[CSS Navigation Bar](#)[CSS Image Gallery](#)[CSS Image Opacity](#)[CSS Image Sprites](#)[CSS Media Types](#)[CSS Attr Selectors](#)



## CSS Tutorial

CSS HOME  
CSS Introduction  
CSS Syntax  
CSS Selectors  
CSS How To  
CSS Backgrounds  
CSS Text  
CSS Fonts  
CSS Links  
CSS Lists  
CSS Tables  
CSS Box Model  
CSS Border  
CSS Outline  
CSS Margin  
CSS Padding  
CSS Dimension  
CSS Display  
CSS Max-width  
CSS Position  
CSS Float  
CSS Inline-block  
CSS Align  
CSS Combinators  
CSS Pseudo-class  
CSS Pseudo-element  
CSS Navigation Bar  
CSS Image Gallery  
CSS Image Opacity  
CSS Image Sprites

## File APIs for Programmers

DOC, XLS, PPT, PDF, MSG and more .NET, Java, SSRS &amp; SharePoint



## CSS Background

[« Previous](#)

CSS background properties are used to define the background effects of an element.

CSS properties used for background effects:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

## Background Color

The `background-color` property specifies the background color of an element.

The background color of a page is set like this:

## Example

```
body {  
  background-color: #b0c4de;  
}
```

# CSS Box Model

[<< Previous](#)

## The CSS Box Model

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around HTML elements, and it consists of: margins, borders, padding, and the actual content.

The box model allows us to add a border around elements, and to define space between elements.

The image below illustrates the box model:



Explanation of the different parts:

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

div tag element에 대해서  
이런 box를 만들어 보여라

### Example

```
div {  
  width: 300px;  
  padding: 25px;  
  border: 25px solid navy;  
  margin: 25px;  
}
```

# Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External style sheet
- Internal style sheet
- Inline style

## External Style Sheet

With an external style sheet, you can change the look of an entire website by changing just one file!

Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the head section:

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a .css extension. An example of a style sheet file called "myStyle.css", is shown below:

```
body {
    background-color: lightblue;
}

h1 {
    color: navy;
    margin-left: 20px;
}
```

# CSS Style Setting

- The basic idea: For a given tag element, set up some style values
- The text between the tag's angle brackets, known as the tag element, is given by the element name
- After the element name and inside curly braces ({} ) is the list of property —value pairs separated by semicolons

```
elem_name { prop1 : val1 ; . . . ; propN : valN }
```

```
<style>  
  body { background-color:black; color:white }  
</style>
```

- [www.w3schools.com/css/default.asp](http://www.w3schools.com/css/default.asp) for list of all the styling possibilities

# CSS Global Style

- Can be placed in **one global location** inside the `<head>` tag
- It then applies to **the whole page**
- Place the style information inside a pair of `<style>` and `</style>` tags

```
<title>Twentieth Century Paradoxes</title>  
<style>  
    CSS style specifications go here  
</style>
```

- Styling Background and Paragraph

```
<style>  
    body {background-color:black; color:white}  
</style>
```

- This results in the background being black, and all text on the page being white

```

<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title> Twentieth Century Paradoxes </title>
  </head>
  <body>
    <h1>Paradoxes</h1>
    <h2>Russell's Paradox</h2>
    <p>The Twentieth Century logician Bertrand Russell
    introduced a curious paradox: <b>This statement is
    false.</b> The statement can't be true, because it
    claims the converse. However, if it is not true, then it's
    false, just as it says. That makes it true. Paradoxically,
    it seems to be neither true nor false, or perhaps both
    true and false.</p>
    <hr/>
    <h2>Magritte's Paradox</h2>
    <p> The famous Belgian artist Ren&eacute; Magritte
    rendered the idea of Russell's Paradox visually in his
    famous painting <i>Ceci n'est pas une pipe</i>. The
    title translates from French, This Is Not A Pipe. The
    painting shows a pipe with the text <i>Ceci n'est pas
    une pipe</i> below it. Superficially, the painting looks
    like a true statement, since it is a <i>picture</i> of
    the pipe, not an actual pipe. However, the assertion is
    also part of the picture, which seems to make it false,
    because it is clearly a painting of a pipe. Paradoxically,
    the truth seems to depend on whether the statement is
    an assertion about the painting or a part of it. But, it's
    both. </p>
  </body>
</html>

```

Figure 4.4 The HTML for a simple pa

## Paradoxes

### Russell's Paradox

The Twentieth Century logician Bertrand Russell introduced a curious paradox: **This statement is false.** The statement can't be true, because it claims the converse. However, if it is not true, then it's false, just as it says. That makes it true. Paradoxically, it seems to be neither true nor false, or perhaps both true and false.

### Magritte's Paradox

The famous Belgian artist René Magritte rendered the idea of Russell's Paradox visually in his famous painting *Ceci n'est pas une pipe*. The title translates from French, This Is Not A Pipe. The painting shows a pipe with the text *Ceci n'est pas une pipe* below it. Superficially, the painting looks like a true statement, since it is a *picture* of the pipe, not an actual pipe. However, the assertion is also part of the picture, which seems to make it false, because it is clearly a painting of a pipe. Paradoxically, the truth seems to depend on whether the statement is an assertion about the painting or a part of it. But, it's both.

```

<style>
  body {background-color:black; color:white}
</style>

```

## Paradoxes

### Russell's Paradox

The Twentieth Century logician Bertrand Russell introduced a curious paradox: **This statement is false.** The statement can't be true, because it claims the converse. However, if it is not true, then it's false, just as it says. That makes it true. Paradoxically, it seems to be neither true nor false, or perhaps both true and false.

### Magritte's Paradox

The famous Belgian artist René Magritte rendered the idea of Russell's Paradox visually in his famous painting *Ceci n'est pas une pipe*. The title translates from French, This Is Not A Pipe. The painting shows a pipe with the text *Ceci n'est pas une pipe* below it. Superficially, the painting looks like a true statement, since it is a *picture* of the pipe, not an actual pipe. However, the assertion is also part of the picture, which seems to make it false, because it is clearly a painting of a pipe. Paradoxically, the truth seems to depend on whether the statement is an assertion about the painting or a part of it. But, it's both.



# Paradoxes

## Russell's Paradox

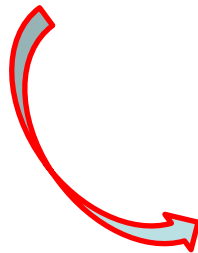
The Twentieth Century logician Bertrand Russell introduced a curious paradox: **This statement is false.** The statement can't be true, because it claims the converse. However, if it is not true, then it's false, just as it says. That makes it true. Paradoxically, it seems to be neither true nor false, or perhaps both true and false.

## Magritte's Paradox

The famous Belgian artist René Magritte rendered the idea of Russell's Paradox visually in his famous painting *Ceci n'est pas une pipe*. The title translates from French, This Is Not A Pipe. The painting shows a pipe with the text *Ceci n'est pas une pipe* below it. Superficially, the painting looks like a true statement, since it is a *picture* of the pipe, not an actual pipe. However, the assertion is also part of the picture, which seems to make it false, because it is clearly a painting of a pipe. Paradoxically, the truth seems to depend on whether the statement is an assertion about the painting or a part of it. But, it's both.

Remember that body, p, h1, h2 are all tags!

```
<style>
body {background-color:darkslategray}
p {color:lightyellow}
h1 {color:gold}
h2 {color:darkorange}
</style>
```



# Paradoxes

## Russell's Paradox

The Twentieth Century logician Bertrand Russell introduced a curious paradox: **This statement is false.** The statement can't be true, because it claims the converse. However, if it is not true, then it's false, just as it says. That makes it true. Paradoxically, it seems to be neither true nor false, or perhaps both true and false.

## Magritte's Paradox

The famous Belgian artist René Magritte rendered the idea of Russell's Paradox visually in his famous painting *Ceci n'est pas une pipe*. The title translates from French, This Is Not A Pipe. The painting shows a pipe with the text *Ceci n'est pas une pipe* below it. Superficially, the painting looks like a true statement, since it is a *picture* of the pipe, not an actual pipe. However, the assertion is also part of the picture, which seems to make it false, because it is clearly a painting of a pipe. Paradoxically, the truth seems to depend on whether the statement is an assertion about the painting or a part of it. But, it's both.

# CSS Inline Styling

- To make a styling specification that applies to a tag in only one place, use [the style attribute](#) inside [that one tag](#)

`<b style = "color: red;" > This statement is false </b>`

- This will result in text turning red

## Paradoxes

### Russell's Paradox

The Twentieth Century logician Bertrand Russell introduced a curious paradox: **This statement is false.** The statement can't be true, because it claims the converse. However, if it is not true, then it's false, just as it says. That makes it true. Paradoxically, it seems to be neither true nor false, or perhaps both true and false.

### Magritte's Paradox

The famous Belgian artist René Magritte rendered the idea of Russell's Paradox visually in his famous painting *Ceci n'est pas une pipe*. The title translates from French, This Is Not A Pipe. The painting shows a pipe with the text *Ceci n'est pas une pipe* below it. Superficially, the painting looks like a true statement, since it is a *picture* of the pipe, not an actual pipe. However, the assertion is also part of the picture, which seems to make it false, because it is clearly a painting of a pipe. Paradoxically, the truth seems to depend on whether the statement is an assertion about the painting or a part of it. But, it's both.

## Paradoxes

### Russell's Paradox

The Twentieth Century logician Bertrand Russell introduced a curious paradox: **This statement is false.** The statement can't be true, because it claims the converse. However, if it is not true, then it's false, just as it says. That makes it true. Paradoxically, it seems to be neither true nor false, or perhaps both true and false.

### Magritte's Paradox

The famous Belgian artist René Magritte rendered the idea of Russell's Paradox visually in his famous painting *Ceci n'est pas une pipe*. The title translates from French, This Is Not A Pipe. The painting shows a pipe with the text *Ceci n'est pas une pipe* below it. Superficially, the painting looks like a true statement, since it is a *picture* of the pipe, not an actual pipe. However, the assertion is also part of the picture, which seems to make it false, because it is clearly a painting of a pipe. Paradoxically, the truth seems to depend on whether the statement is an assertion about the painting or a part of it. But, it's both.



# Marking Links with Anchor Tags

- Two parts to a hyperlink
  - The highlighted text in the current document, which is called **the anchor text**
  - The address of the other Web page, called **the hyperlink reference**
- Both parts of the hyperlink are specified in **the anchor tag**:
  - Begin with `<a` and make sure there's a space after the a but not before it. **a** is for anchor
  - Give the hyperlink reference using the href attribute `href="filename"`, making sure to include the double quotes
  - Close the anchor tag with the `>` symbol
  - Give the anchor text, which will be highlighted when it is displayed by the browser
  - End the hyperlink with the `</a>` tag

Bertrand Russell



Bertrand	<code>&lt;a href="http://www.bioz.com/bios/sci/russell.html"&gt;</code>	Russell	<code>&lt;/a&gt;</code>
normal text	hyperlink reference	anchor	

# Structure of the Image Tag

- An image tag specifies a file that contains an image:

```

```

- `src` is the abbreviation for “source”
  - `filename` uses the same rules for absolute and relative pathnames as anchor tags
  - `alt` value specifies an alternative form for the image, usually a textual description
  - `src` and `alt` are tag attributes belonging to `img` tag
- Screen readers don’t know what the image is, but they can read the description of the alt tag
  - HTML requires `alt` tags
  - When an image is not available or loads slow, browsers display the alt information

# Tag Attributes

## HTML Attributes

- HTML elements 는 attributes 를 가질 수 있다.
- attributes 는 element 에 대한 추가적인 정보를 제공한다.
- attribute 는 항상 start tag 안에 명시되어 진다.
- attribute name="value" 과 같은 형태이다.

## Attribute Example:

HTML 링크들은 <a> tag 로 정의되어 진다.  
링크주소는 attribute 로서 제공되어 진다.

```
<a href="http://www.w3schools.com">This is a link</a>
```

## Always Quote Attribute Values

Attribute 값은 상항 따옴표로 둘러 싸여야 한다.  
쌍 따옴표가 일반적이지만, 홑 따옴표도 허락된다.








## 어떤 HTML Tag가 어떤 attribute들을 가지는가?

Tags and attributes can be looked up: [www.w3schools.com/tags/default.asp](http://www.w3schools.com/tags/default.asp)

# HTML <button> Tag

A clickable button is marked up as follows:

```
<button type="button">Click Me!</button>
```

Attribute	Value	Description
<u>autofocus</u>	 autofocus	Specifies that a button should automatically get focus when the page loads
<u>disabled</u>	disabled	Specifies that a button should be disabled
<u>form</u>	 form_id	Specifies one or more forms the button belongs to
<u>formaction</u>	 URL	Specifies where to send the form-data when a form is submitted. Only for type="submit"
<u>formenctype</u>	 application/x-www-form-urlencoded multipart/form-data text/plain	Specifies how form-data should be encoded before sending it to a server. Only for type="submit"
<u>formmethod</u>	 get post	Specifies how to send the form-data (which HTTP method to use). Only for type="submit"
<u>formnovalidate</u>	 formnovalidate	Specifies that the form-data should not be validated on submission. Only for type="submit"
<u>formtarget</u>	 _blank _self _parent _top framename	Specifies where to display the response after submitting the form. Only for type="submit"
<u>name</u>	name	Specifies a name for the button
<u>type</u>	button reset submit	Specifies the type of button
<u>value</u>	text	Specifies an initial value for the button

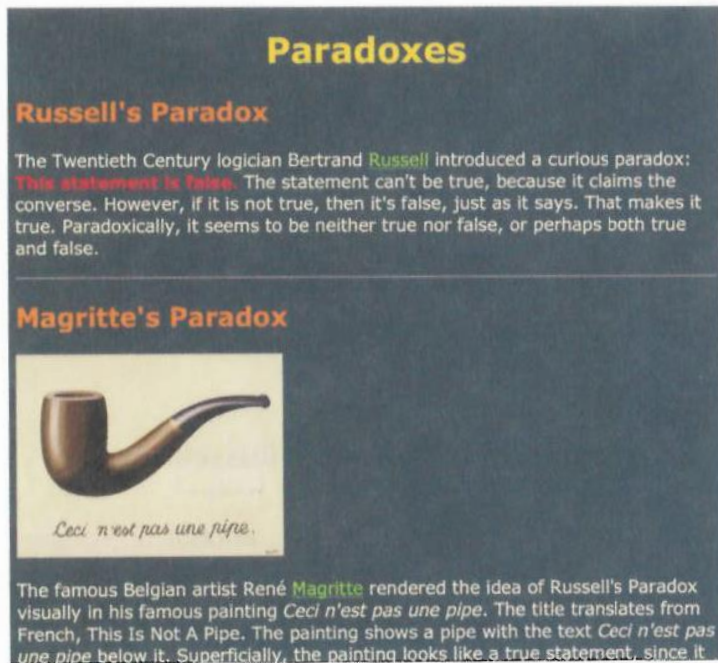
# Attributes for Image Tags

- Image File Types: `.gif`, `.png`, and `.jpg` are frequently used extensions

```




```

- Specifying only the `width` or the `height` of an image is enough



- Homework: Update The HTML code in Figure 4.4 for this display!

# Styling Position for Images

- If the image is the same size or smaller than the surrounding text, it is placed in line just  like a letter.
- If the image is larger than the letters , images are inserted in a way that the text lines up with the bottom of the image
- Messy? Hard to read?
-  A better way to place images in text is to flow the text around them
- ``
- You can either by positioning the image on the left with the text to its right, or vice versa
  - Use the `style attribute` in the image tag with the value `"float:left"` or `"float:right"`

# Making an Image Link

- We do not have to use only text as the anchor
- Images can be used as well
- Combine an anchor tag with an image tag:

```
<a href="history_red_square.html">  </a>
```

hyperlink reference

anchor

- When the page displays, the usual highlighting that links receive will be used to mark the .gif as a link

# Absolute Pathnames vs Relative Pathnames

- Absolute Pathnames (URLs)

- In these anchor tags, the hyperlink reference is an entire URL
  - The Web browser needs to know how/where to find the page
  - A URL <http://www.bioz.com/bios/sci/russell.html>
  - a protocol specification, <http://>
  - a domain or IP address, [www.bioz.com](http://www.bioz.com)
  - a path to the file, [/bios/sci/russell.html](http://www.bioz.com/bios/sci/russell.html)

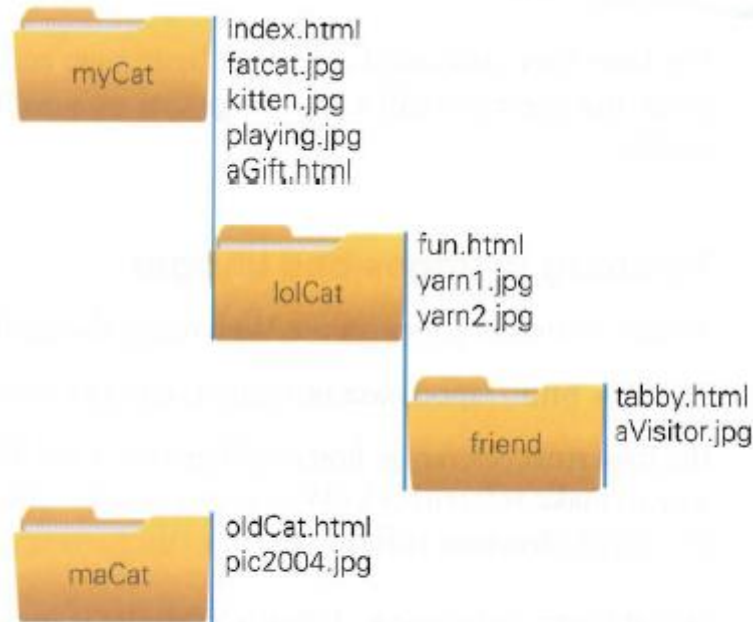
- Relative Pathnames

- Often links refer to [other Web pages on the same site](#)
- These pages are all be stored [in the same or nearby folders](#)
- These anchor tags use relative pathnames
- A relative pathname describes how to find the referenced file relative to the file in which [the anchor tag appears](#)



# Going “Deeper” in a Folder

- When the file containing the anchor and the referenced file are in the same folder — we just give the file name `<a href="russellbio.html"> Russell </a>`
- When the referenced file is “deeper” in the directory, we simply give the path from the current folder down to the file  
`<a href="biographies/russellbio.html"> Russell </a>`
- Navigating up to another folder: add `../` to the beginning of the href attribute  
`<a href="../biographies/russellbio.htm "> Russell </a>`
  - Each `../` moves up one level higher in the hierarchy which is a UNIX operating system technique



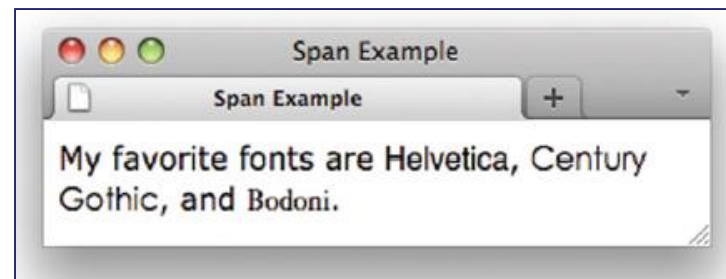
**Figure 4.8** Sample directory (folder) structure for a Web site.

# Span tag

- What if there is no tag to add a style attribute?
- Tag 없는곳에 style을 심고 싶다면 → Use the `<span>` tag
- `<span>`'s only job is to surround text, and provide a place for style and other attribute information

My favorite fonts are

```
<span style="font-family : helvetica"> Helvetica </span>,  
<span style="font-family : century gothic"> Century Gothic </span>, and  
<span style="font-family : bodoni"> Bodoni </span>.
```



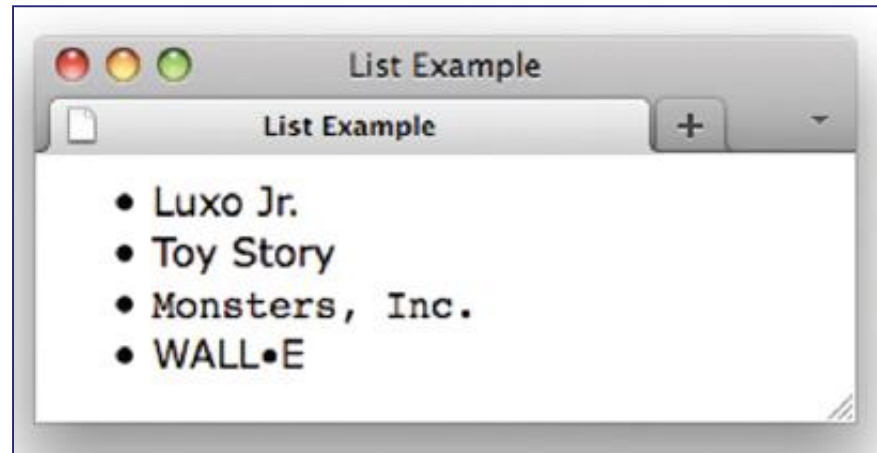
# For the Rest of the Chapter

- List Tags
  - Unordered List
  - Ordered List
  - Nested List
  - Definition List
- Table Tag
  - Table Captions
  - Column Headings
- Box Model of CSS
- Class attribute for Style
- Styling for HyperText Link using Pseudo Classes

# Lists Tags: Unordered List

- The easiest type of list is the unordered list
- **Unordered list** tags `<ul>` and `</ul>` surround the items of the list
- The items are enclosed in **list item** tags, `<li>` and `</li>`

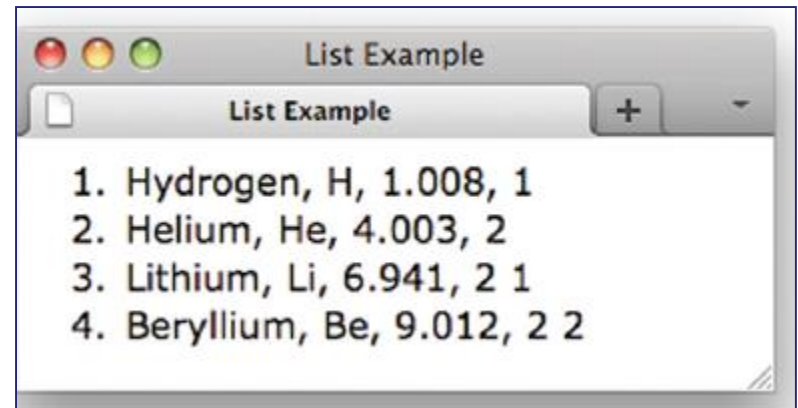
```
<ul>
  <li>Luxo Jr.</li>
  <li>Toy Story</li>
  <li><span style="font-family:courier">
    Monsters, Inc.</span></li>
  <li>WALL•E</li>
</ul>
```



# Lists Tags: Ordered List

- Another list is [the ordered list](#): the tags `<ol>` and `</ol>`
- Bullets in the unordered list are replaced with numbers in the ordered list
- The ordered list behaves just like the unnumbered list

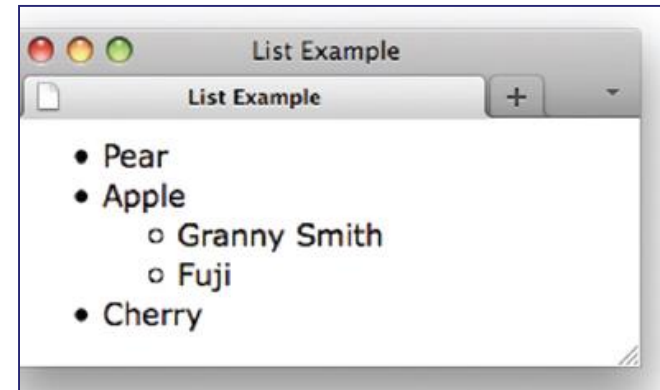
```
<ol>  
  <li> Hydrogen, H, 1.008, 1 </li>  
  <li> Helium, He, 4.003, 2 </li>  
  <li> Lithium, Li, 6.941, 2 1 </li>  
  <li> Beryllium, Be, 9.012, 2 2 </li>  
</ol>
```



# Lists Tags: Nested List

- You can also have a list within a list
- Make a **sublist** within the main list
- Notice that sublists use a **different bullet symbol**

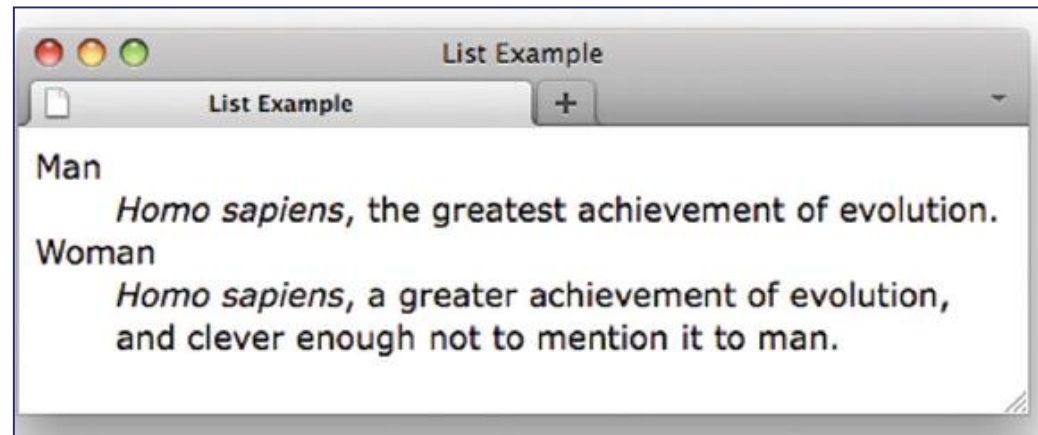
```
<ul>  
  <li>Pear</li>  
  <li>Apple</li>  
    <ul>  
      <li>Granny Smith</li>  
      <li>Fuji</li>  
    </ul>  
  <li>Cherry</li>  
</ul>
```



# Lists Tags: Definition List

- The definitional list is indicated by the `<dl>` and `</dl>` tags
- Definitional lists are made of **definitional terms** surrounded by `<dt>` and `</dt>` And **definitional data** surrounded by `<dd>` and `</dd>`

```
<dl>
  <dt> Man </dt>
  <dd> Homo sapiens, the greatest
    achievement of evolution. </dd>
  <dt> Woman </dt>
  <dd> Homo sapiens, a greater
    achievement of evolution, and clever
    enough not to mention it to man.
</dd>
</dl>
```





# Handling Table Tags

- A table is a good way to present information: It's like defining a list of lists
- The main list items, called rows, has one or more items, called cells
- The table is enclosed in `<table>` and `</table>`
- The table can have a border, if you use the border attribute
- Each row is enclosed in `<tr>` and `</tr>`
- Cells are surrounded by table data tags, `<td>` and `</td>`

```
<table border="1">
  <tr>
    <td>Canada</td>
    <td>Ottawa</td>
    <td>English/French</td>
  </tr>
  <tr>
    <td>Iceland</td>
    <td>Reykjavik</td>
    <td>Icelandic</td>
  </tr>
</table>
```



Canada	Ottawa	English/French
Iceland	Reykjavik	Icelandic

# Handling Tables: Tables Captions and Column Headings

- Caption tags are `<caption>` and `</caption>`
- Captions are centered at the top of the table
- Column headings are the first row of the table (i.e., `table heading`)
- In the “heading” row, column headings have table heading tags `<th>` and `</th>`
- Column headings display in bold

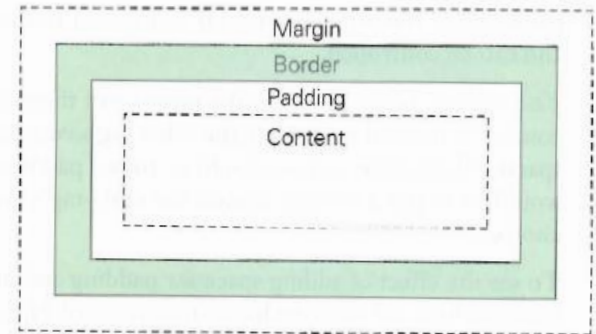
```
<table border="1">
  <caption>Country Data</caption>
  <tr>
    <th>Country</th>
    <th>Capital</th>
    <th>Language(s)</th>
  </tr>
  <tr>
    <td>Canada</td>
    <td>Ottawa</td>
    <td>English/French</td>
  </tr>
  <tr>
    <td>Iceland</td>
    <td>Reykjavik</td>
    <td>Icelandic</td>
  </tr>
  <tr>
    <td>Norway</td>
    <td>Oslo</td>
    <td>Norwegian</td>
  </tr>
</table>
```



Country	Capital	Language(s)
Canada	Ottawa	English/French
Iceland	Reykjavik	Icelandic
Norway	Oslo	Norwegian

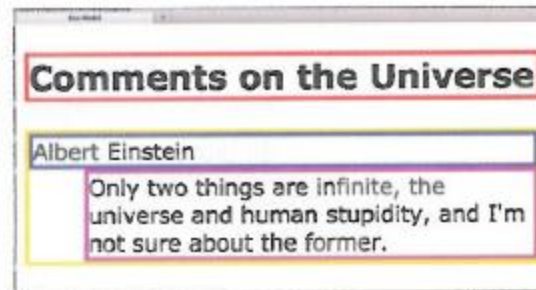
# Box Model of CSS

- CSS considers every HTML5 element to be enclosed in a “box”
  - The boxes will not be visible, unless you make them visible
- Can clearly visualize content, padding, border, and margin

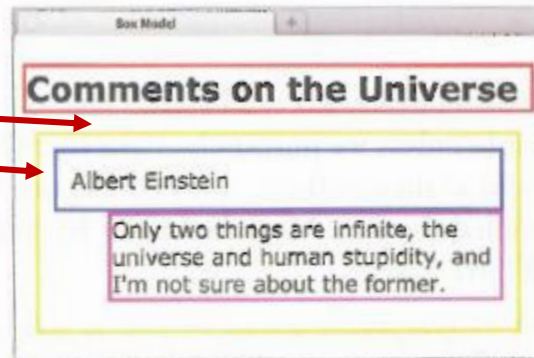


**Figure 4.9** The boxes of the Box Model: Content is enclosed in padding, which is enclosed in a border, which is enclosed by a margin. All of these can be controlled.

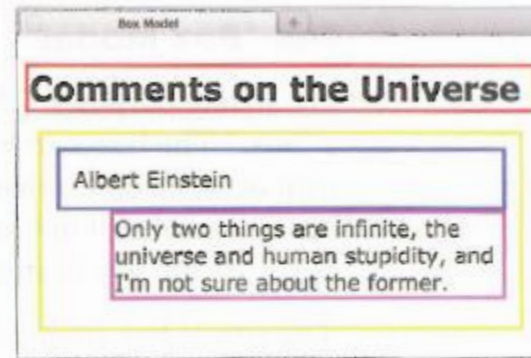
```
<style>
h2 {border-style:solid; border-color:red}
dl {border-style:solid; border-color:gold}
dt {border-style:solid; border-color:blue}
dd {border-style:solid; border-color:magenta}
</style>
</head>
<body>
<h2>Comments on the Universe</h2>
<dl>
<dt>Albert Einstein </dt>
<dd>Only two things are infinite, the
universe and human stupidity, and
I'm not sure about the former.</dd>
</dl>
```



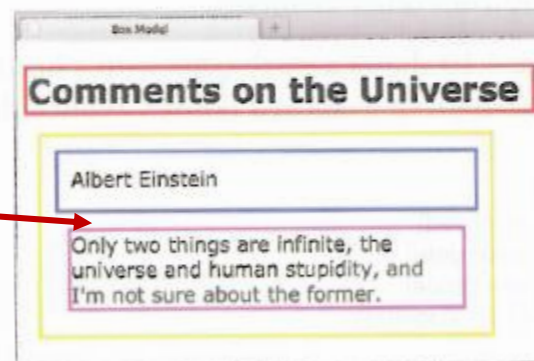
**Figure 4.10** The boxes around definitional list elements with 0-width padding and margin.



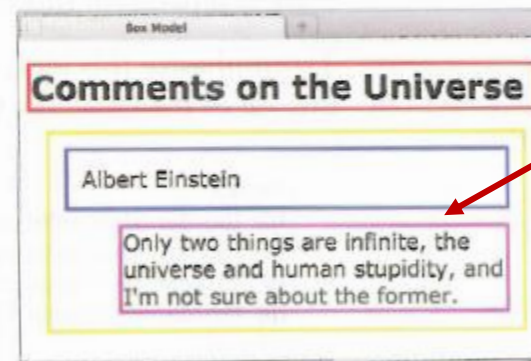
(a)



(b)



(c)



(d)

**Figure 4.11** Padding and margins of 10 pixels (px): (a) yellow, both padding and margin; (b) blue, padding only; (c) purple, margin only; (d) purple, only top margin styled.

# Globally Speaking

- We have placed our style information either [in the head](#) or [directly in the tag](#)
- If we want to easily have the same style information on [multiple pages](#), to have a consistent “look and feel”, we should instead use [a Style File](#)
- Transfer your style section of the page into a new text file
  - do not include the style tags, just [the style elements](#)
- Save the file with a .css extension, you could use [myFaveStyle.css](#)
- Go back to the [original .html](#) page and add a [link to your .css page](#) in the [head](#)

```
<link rel="stylesheet" type="text/css" href="myFaveStyle.css" />
```

- It is okay to use both [a separate file](#), and [internal style tags](#)

# The Cascade of Styling

- Style-Information sources are varied!
  - **default**: comes with the browser
  - **external**: styling in a file (.css file)
  - **style section**: defined inside of <style> tags
  - **inherited**: properties set by enclosing tag
  - **tagged**: defined with the style attribute
- For any property, the closest specification wins
  - **Tagged > Inherited > Style section > External > Default**

# Adding Class Attribute to Style

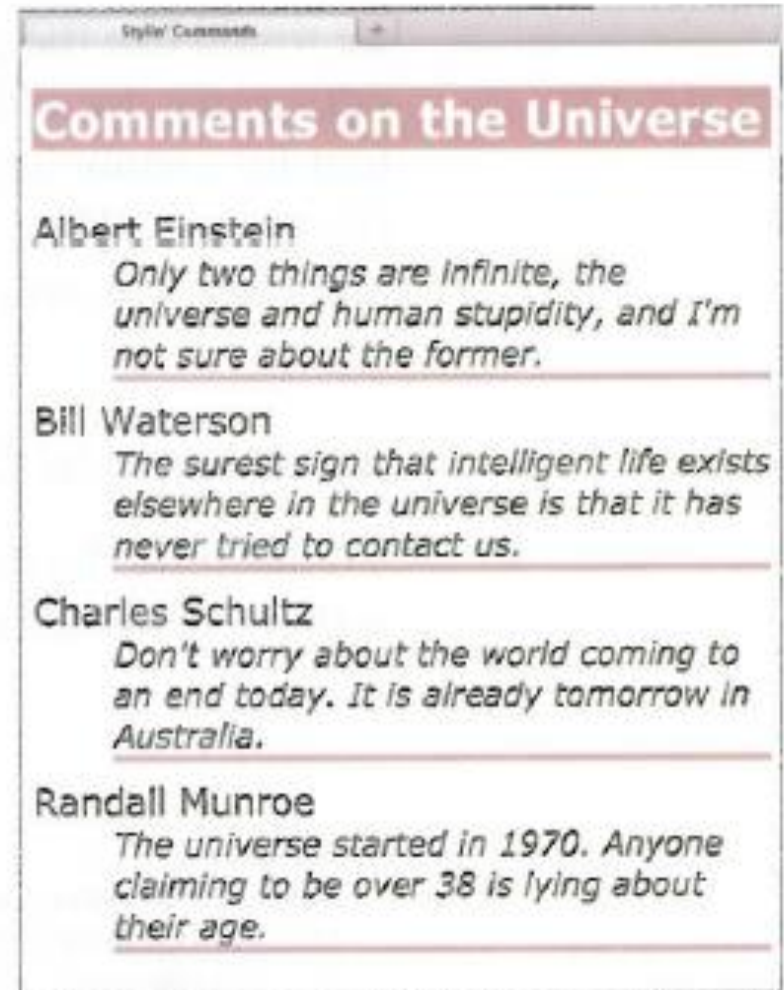
- A class is a family of styling specifications with a common name
- The class is given in 2 places:
  - In the style definition inside the style tags in the <head>
  - At the site in the HTML code when the code is used
- Example: `<h2 class="scientist"> . . . </h2>`
  - For the style definition, we append the class name (scientist) to the tag with a dot:  
`h2.scientist`
  - Plain <h2> tags are styled with those features that apply to all <h2> tags
- Each separate class gives additional styling specifications
- Same tag의 contents가 많을때 contents를 그룹지어서 styling



# Class Attribute Example 1 [1/2]

```
<html>
<head><title>Stylin' Commands</title></head>

<style>
  h2 {background-color : rosybrown; color : white;}
  dt {padding-top:8px; font-size : large;}
  dd {font-style : italic; border-bottom-style:solid;
      border-bottom-color:rosybrown;
      border-bottom-width:3px;}
</style>
</head>
<body><h2>Comments on the Universe</h2>
  <dl><dt>Albert Einstein </dt>
    <dd>Only two things are infinite, the universe
      and human stupidity, and I'm not sure about
      the former.</dd>
    <dt>Bill Waterson </dt>
    <dd>The surest sign that intelligent life exists
      elsewhere in the universe is that it has never
      tried to contact us.</dd>
    <dt>Charles Schultz </dt>
    <dd>Don't worry about the world coming to an end
      today. It is already tomorrow in Australia.</dd>
    <dt>Randall Munroe</dt>
    <dd>The universe started in 1970. Anyone claiming
      to be over 38 is lying about their age.</dd>
  </dl>
```



**Figure 4.12** Styling for a page with a definitional list. (Munroe's joke from 2008 is explained in the Fluency Byte on pages 560 and 561.)

# Class Attribute Example 1 [2/2]

```
<html>
<head><title>Stylin' Commands</title></head>
<style>
h2 {background-color : rosybrown; color : white;}
dt {padding-top:8px; font-size : large;
    text-align:right}
dt.scientist {color:red}
dt.cartoonist {color:royalblue;
    font-family:comic sans MS}
dd {font-style : italic; border-bottom-style:solid;
    border-bottom-color:rosybrown;
    border-bottom-width:3px;}
</style>
</head>
<body><h2>Comments on the Universe</h2>
<dl><dt class="scientist">Albert Einstein </dt>
<dd>Only two things are infinite, the universe
    and human stupidity, and I'm not sure about
    the former.</dd>
<dt class="cartoonist">Bill Waterson </dt>
<dd>The surest sign that intelligent life exists
    elsewhere in the universe is that it has never
    tried to contact us.</dd>
```

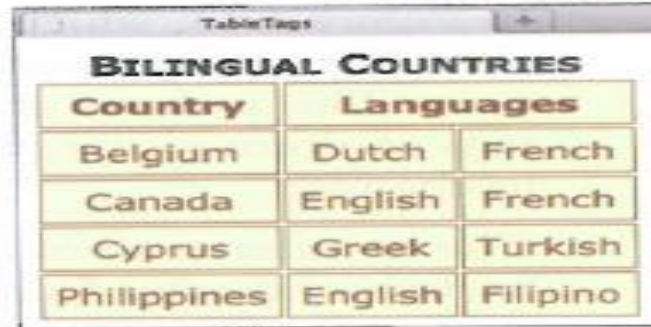


**Figure 4.13** The definitional list exploiting the class feature to style scientists and cartoonists differently.

## Class Attribute Example 2 [1/2]

### Using Class Attribute for Alternating Rows

```
caption {  
    font-size:large;  
    font-variant:small-caps;  
    font-weight:bold;  
td, th {  
    border-style:solid;  
    border-width:1px;  
    padding:5px;  
    text-align:center;  
    background-color:cornsilk;  
    color:saddlebrown;  
}
```



Country	Languages	
Belgium	Dutch	French
Canada	English	French
Cyprus	Greek	Turkish
Philippines	English	Filipino

**Figure 4.14** The Bilingual Countries table, styled to enhance the caption and make the cells easier to read.

```
<table border="1">  
  <caption>Bilingual Countries</caption>  
  <tr><th>Country</th><th colspan="2">Languages</th></tr>  
  <tr><td>Belgium</td><td>Dutch</td><td>French</td></tr>  
  <tr><td>Canada</td><td>English</td><td>French</td></tr>  
  <tr><td>Cyprus</td><td>Greek</td><td>Turkish</td></tr>  
  <tr><td>Philippines</td><td>English</td><td>Filipino</td></tr>  
</table>
```



## Using Class Attribute for Alternating Rows

Country	Languages	
Belgium	Dutch	French
Canada	English	French
Cyprus	Greek	Turkish
Philippines	English	Filipino

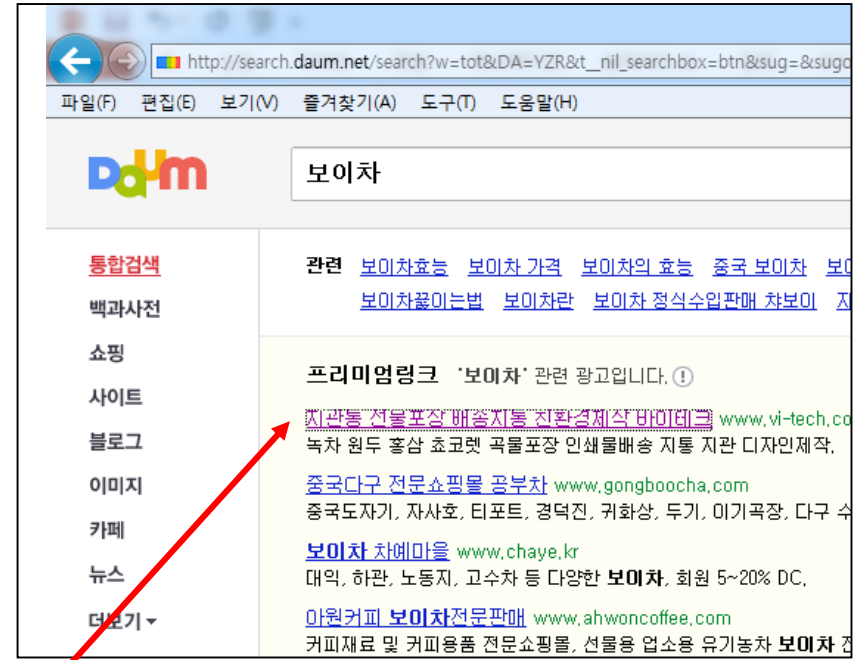
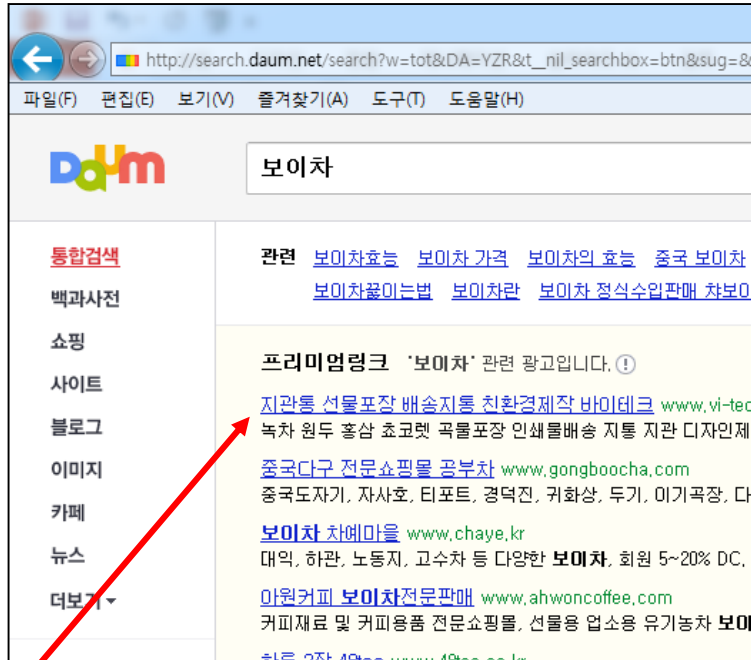
```
td, th {
    border-style:solid;
    border-width:1px;
    padding:5px;
    text-align:center;
    background-color:cornsilk;
    color:saddlebrown;
}
tr.alt td {
    background-color:blanchedalmond;
}
```

```
<table>
  <caption>Bilingual Countries</caption>
  <tr><th>Country</th><th colspan="2">Languages</th></tr>
  <tr><td>Belgium</td><td>Dutch</td><td>French</td></tr>
  <tr class="alt"><td>Canada</td><td>English</td><td>French</td></tr>
  <tr><td>Cyprus</td><td>Greek</td><td>Turkish</td></tr>
  <tr class="alt"><td>Philippines</td><td>English</td><td>Filipino</td></tr>
</table>
```

**Figure 4.15** Table with rows of alternating color, and the styling that achieves it.

# Styling for HyperText Link using Pseudo Classes

## Before and After of Mouse Click



Hypertext link가 심어져 있는 단어, 문장, 그림등이 많은데

- Contents에 click이 되기전에
- Mouse가 contents위에 hovering 할때
- Contents가 click되어 그 내용이 visit되고 있는 도중에
- Contents가 visit된이후에
- 무슨 색으로 무슨 모양으로 보일것인지를 styling

# Styling for HyperText Link: Pseudo Classes [1/2]

- There are properties to change the style of links (<a href=“..”> tag)
- The default style for links is blue and underlined
- Style the anchor tag:

```
<style>  
  a      {text-decoration : none}  
  a:link {color : darkviolet }  
  a:visited {color : gray }  
  a:hover {color : red}  
</style>
```
- These styles are slightly different from the other tags
- The anchor tag has several different states that can be styled separately
- These states are referred to as pseudo classes

The four pseudo class states are

link, styling for an unvisited link (anchor text)  
visited, styling for links that have been visited  
hover, styling when the cursor hovers over a link  
active, styling for links that are in process

# Pseudo Classes [2/2]



(a)

```
ul {list-style-type:none; margin:0; margin-left:140px; padding:0;}
li {text-align:center; font-size:20px;}
li.top {display:inline;}
a.top {text-decoration:none; width:140px; height:40px; background-color:none;
float:left; padding:10px;padding-top:20px; margin-left:3px; color:white;
border-bottom-color:white; border-bottom-width:1px; border-bottom-style:solid}
a.top:hover {background-color:blueviolet;}
```

(b)

```
<ul>
  <li class="top"><a class="top" href="#">HOME</a></li>
  <li class="top"><a class="top" href="#">ABOUT</a></li>
  <li class="top"><a class="top" href="#">TRIPS</a></li>
  <li class="top"><a class="top" href="#">CONTACT</a></li>
</ul><br/><br/>
```

(c)

**Figure 4.16** Horizontal navigation bar: (a) hovering, (b) style specification, and (c) usage in the body.



# Navigation Bars

- Common on the tops of many web pages
  - Can be horizontal or vertical
- Turns off the bullets in an unordered list of links → `list-style-type: none`



(a)

```
<ul style="float:left; margin-right:35px; margin-left:25px">  
  <li class="side">Past Trips</li>  
  <li class="side"><a class="side" href="#">2014</a></li>  
  <li class="side"><a class="side" href="#">2013</a></li>  
  <li class="side"><a class="side" href="#">2012</a></li>  
  <li class="side"><a class="side" href="#">2011</a></li>  
  <li class="side"><a class="side" href="#">2010</a></li>  
  <li class="side"><a class="side" href="#">2009</a></li>  
  <li class="side"><a class="side" href="#">2008</a></li>  
  <li class="side"><a class="side" href="#">2007</a></li>  
  <li class="side"><a class="side" href="#">2006</a></li>  
</ul>
```

(b)

```
ul {list-style-type:none; margin:0; margin-left:140px; padding:0;}  
li {text-align:center; font-size:20px;}  
a.side {text-decoration:none; display:block; width:100px;  
        color:white; background-color:none; }  
a.side:hover {background-color:magenta;}
```

(c)

**Figure 4.17** Vertical navigation bar: (a) hovering, (b) use in the body, and (c) styling specifications.



# Gradient Background

- First, create a gradient image and save it as a `background1.jpg`
  - `narrow, but tall` to cover a long page
  - `url( )` will get the pathname of the `background1.jpg` file
- `background-repeat: repeat-x` will repeat the image `in the x-direction` to fill the space
- `body { background-image: url('background1.jpg');`  
`background-repeat: repeat-x; color: white;`  
`font-family:Helvetica Neue Light;}`

• `background1.jpg`

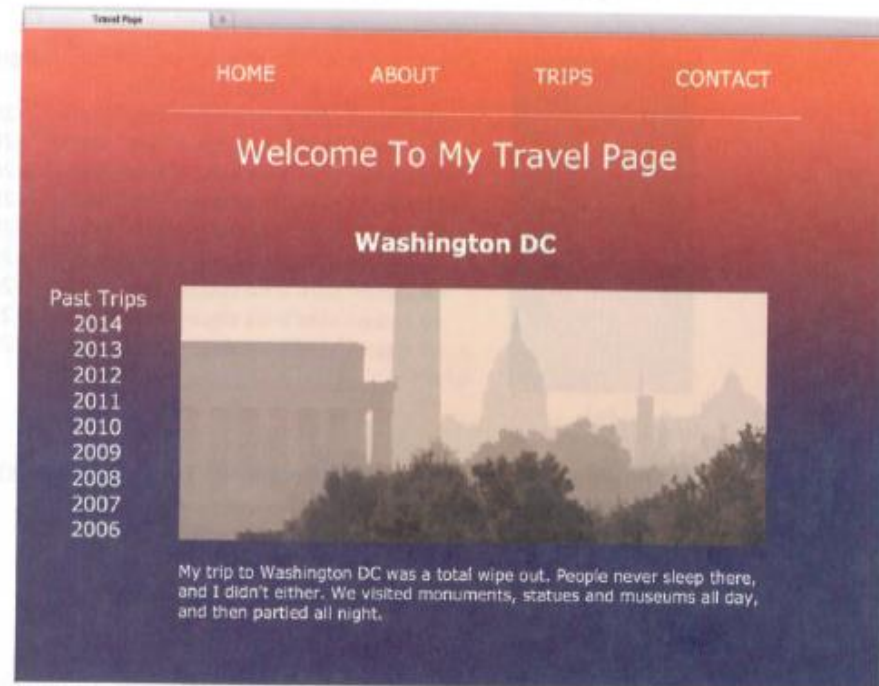


Figure 4.18 A Web page with navigation bars.

# Methods for Publishing your web site

- **Method 1:** Using a web hosting server whose domain name is already registered
  - SNU plaza machine or any ISP machine
    - Suppose server name is “testserver.com” and your log-id is “hjk”
  - Edit your home page with Notepad2 and Name the file as index.html
  - Set up the ftp program & Load the index.html into testserver.com/hjk/public\_html/
- **Method2:** Using your own server in SNU
  - Edit your home page with Notepad2 and Name the file as index.html
  - Install the public WAS (web application server) SW such as Apache Tomcat, Jboss, etc
  - Invent your own server's domain name, say hjk-cs
  - You ask 전산원 to register “hjk-cs.snu.ac.kr” (serve의 domain 등록)
    - Then, 전산원 assigns IP-address to your server's domain name “hjk-cs.snu.ac.kr”
    - Then, 전산원 updates DNS server table

# Methods for Publishing your web site

- **Method3:** Using **your own PC** in your home
  - Edit your home page with Notepad2
  - Install the public **WAS (web application server) SW** such as Apache Tomcat, Jboss, etc
  - Buy **a Public&Fixed IP Address** from your ISP company (KT, U+, cableTV)
  - Buy **a domain name** from 도메인-판매대행업체 (Ex. domain.blueweb.co.kr, www.nameip.co.kr ) on behalf of a domain organization
  - Then, 도메인-판매대행업체 registers **(domain-name, IP-address)** to a correspondent organization (such as 인터넷진흥원 (KISA))

# List of references for the specifications

- Remember that HTML tags are all **reserved** tags!
- Tag attributes are also all **reserved**!
- Properties in CSS style are also all **reserved**!

**Table 4.3** Official W3C Web sites referenced in this chapter

Tags	<a href="http://www.w3schools.com/tags/default.asp">www.w3schools.com/tags/default.asp</a>
Special characters like Ö	<a href="http://www.w3schools.com/tags/ref_entities.asp">www.w3schools.com/tags/ref_entities.asp</a>
Validation	<a href="http://validator.w3.org/#validate_by_upload">validator.w3.org/#validate_by_upload</a>
CSS	<a href="http://www.w3schools.com/css/default.asp">www.w3schools.com/css/default.asp</a>
Color names	<a href="http://www.w3schools.com/cssref/css_colornames.asp">www.w3schools.com/cssref/css_colornames.asp</a>
HTML lists	<a href="http://www.w3schools.com/html/html_lists.asp">www.w3schools.com/html/html_lists.asp</a>
List styles	<a href="http://www.w3schools.com/cssref/pr_list-style-type.asp">www.w3schools.com/cssref/pr_list-style-type.asp</a>
Box Model	<a href="http://www.w3schools.com/css/css_boxmodel.asp">www.w3schools.com/css/css_boxmodel.asp</a>

# Summary

- A working set of HTML tags, giving you the ability to create a Web page
- An explanation of how links are marked with [anchor tags](#)
- [Absolute and relative pathnames](#): Relative pathnames refer to files deeper or higher in the directory hierarchy
- The 2 most popular image formatting schemes and how to place them in a page
- [Cascading Style Sheets](#), a general system for styling Web documents
- Lists and tables
- The precision achievable by using classes