

**FLUENCY6**

with information technology

SKILLS, CONCEPTS, & CAPABILITIES



LAWRENCE SNYDER

# Chapter 4

## *A Hypertext Markup Language Primer*

# 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

# Web Pages

- Web pages are created, stored, and sent in encoded form
- A **browser** converts them to what we see on the screen
- Hypertext Markup Language (HTML) is the main language used to define how a Web page should look
- We use HTML version 5

# Formatting with Tags

- Formatting instructions are given by tags, which the browser obeys.
- Tags are words or abbreviations enclosed in angle brackets, < and >
- Many tags come in pairs
- The second of the pair comes with a slash: **<title> Fluency </title>**
- The tag pair surrounds the text to be formatted like parentheses

# Formatting with Tags

`<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!***

# Tags for Bold and Italic

- 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 />
- Examples: <hr /> produces a horizontal line  
<br /> 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)

# Required Tags

- HTML tags that are required for every Web page:

**<!doctype html>**

**<html>**

**<head>**

**</head>**

**<body>**

**</body>**

**</html>**

# Required Tags

```
<head>  
  <meta charset="UTF-8" />  
  <title>Title</title>  
  additional material  
  discussed later  
</head>
```

- The head section contains the beginning material like the title and other information that applies to the whole page

# Required Tags

**<body>**  
the main content of  
the page goes here  
**</body>**

- The body section contains the content of the page.
- This <html> <head> <body> form must *always* be followed
- All of these tags are required

# Configure Your Computer for Writing HTML

- Check that two programs are installed:
  - A browser (check for Firefox)
  - A text editor (Notepad++ for Windows or Text Wrangler for Macs)
- Both programs are free
- These programs are preferred for technical and educational reasons

# Firefox

- 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)

# Text Editor

- A text editor is a basic way to process text
- Our word processors are called “what-you-see-is-what-you-get” or (WYSIWYG)
- Word processors include many application-specific information in their files
- This information confuses browsers

# Text Editor

- You must use a text editor to write HTML
- Text editors do not include this extra information
- Browsers want Web pages written in ASCII characters only
- Think of ASCII as the normal keyboard characters with “nothing strange”



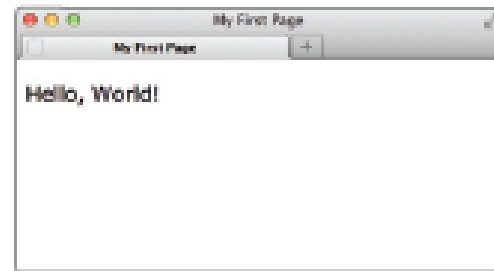
# Text Editor

- 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 simple text editors installed.
  - TextEdit can be found on the Mac
  - Notepad comes with Windows
- TextWrangler and Notepad++ are better choices

# Hello, World!

- To produce your first HTML page, follow these instructions:
  1. In your text editor, open a *New document instance*
  2. Carefully type in your text (see next slide)
  3. Save the file as starterPage.html
  4. Open the file with the Firefox browser

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



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

# 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

# Headings in HTML

- Documents can have headings and subheadings
- HTML provides several levels of *heading tags*:
  - **<h1> and </h1> level one**
  - **<h2> and </h2> level two**
  - ...
  - **<h6> and </h6> 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.

# HTML Format Versus Display Format

- HTML source code tells the browser how to produce the formatted page based on the *meanings of the tags*
- HTML is written in a structured format to make it easier for people to understand
- Indenting is frequently used to emphasize the tags' meanings

# White Space

- 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



# White 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

<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

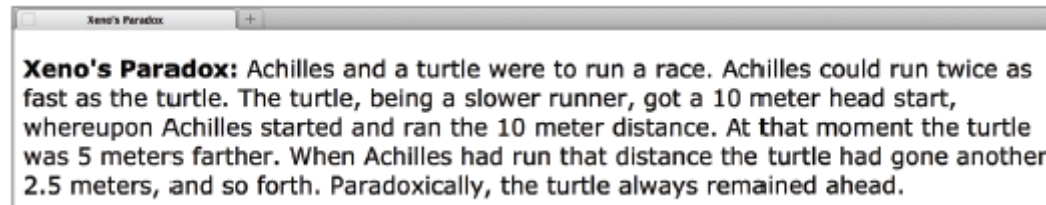
moment the turtle was 5 meters farther.

When Achilles had run

that distance the turtle had gone another 2.5 meters,

and so forth. Paradoxically, the turtle always remained

ahead. </p>



# Attributes

- Some tags have *attributes*
- Given as name=value pairs inside the opening tag
- The meta tag takes a charset attribute:  
`<meta charset="UTF-8" />`
  - Different tags take different attributes
  - Some are optional and some are required

Table 4.1 Basic tags for HTML

Start Tag	End Tag	Meaning	Required
<!doctype html>		First tag in an HTML5 file	✓
<html>	</html>	Tag enclosing all HTML text	✓
<title>	</title>	Title bar text; describes page	✓
<head>	</head>	Preliminary material; e.g., title at start of page	✓
<body>	</body>	The main, content part of the page	✓
<p>	</p>	Paragraph	
<hr/>		Line (horizontal rule)	
<h1>...<h6>	</h1>...</h6>	Headings, six levels	
<b>	</b>	Bold	
<i>	</i>	Italic	
<pre>	</pre>	Preformatted text in which white space matters	
<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 a .jpg, png, or .gif file	
 		Break, continue text on the next line	

# 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

`&lt;` displays as `<`  
`&gt;` displays as `>`  
`&amp;` displays as `&`

# 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

Table 4.2 Specifying accent marks for Western European languages

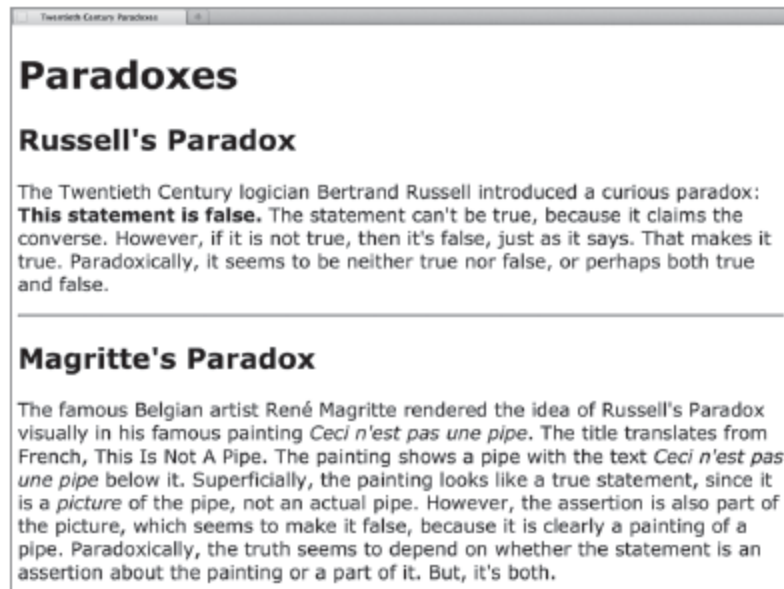
Symbol	Text	Number	Symbol	Text	Number	Symbol	Text	Number
à	&agrave;	&#224;	ê	&ecirc;	&#234;	ô	&ocirc;	&#244;
ä	&auml;	&#228;	î	&icirc;	&#238;	ù	&ugrave;	&#249;
é	&eacute;	&#233;	ó	&oacute;	&#243;	ã	&atilde;	&#227;
í	&iacute;	&#237;	ø	&oslash;	&#248;	è	&egrave;	&#232;
ò	&ograve;	&#242;	ü	&uuml;	&#252;	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;
â	&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).

```

<!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 <img alt="A pipe" data-bbox="340 570 380 600"/> 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).



# HTML

Notice these features in the previous slide:

- Title appears on the browser's frame
- The h1 and h2 text is shown in bold
- The separating line extends the full browser window width
- An acute accent appears in Margritte's first name
- Text in italics as directed by the `<i>` tag.

# Compose and Check

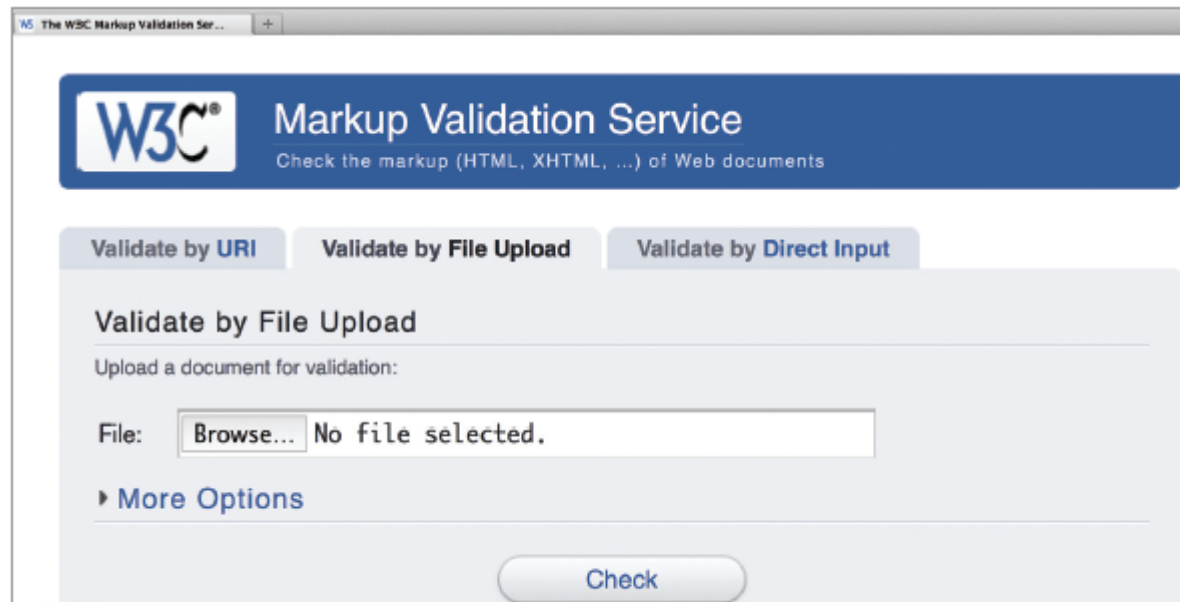
- A productive way to work is to keep two windows open:
  - your text editor
  - your browser
- After writing a few HTML formatting tags, save the file
- Check the result in the browser by a Reload or Refresh of the source
- Repeat

# Markup Validation Service

- Another way to limit the mistakes you make is to have it automatically validated
- 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

# Check My Work

- Normally, validation doesn't happen until the HTML page is finished and stable
- During “compose and check,” the validation occurs at a “stopping place”
- To validate go to the W3C Markup Validation Service  
at:[validator.w3.org/#validate\\_by\\_upload](https://validator.w3.org/#validate_by_upload)



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

# Check My Work

- 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.
- It's common to have a lot of errors at the start

# Cascading Style Sheets (CSS)

- ***Cascading Style Sheets (CSS)*** can be used to control the layout and styling of Web pages

# Setting Global Style

- Style specifications can be placed in a global location inside the **<head>**
- They then apply to the whole page
- Place the style information inside a pair of **<style>** and **</style>** tags



# Styling Background and Paragraph

- for each tag, you can give properties

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

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

# Global Styling

- Give the name of the tag you want to control
- But do not use the brackets around the name
- After the element name and inside curly braces ({} ) is the list of property-value pairs separated by semicolons

# 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;">text</b>**

- this will result in text turning red
- the value of the style attribute is a list of property-pairs with the same format as the contents of { } in the <style> section.

# Styling The Paradox

Inside the <head>:

```
<style>
```

```
  body {background-color: darkslategray}
```

```
  p {color: lightyellow}
```

```
  h1 {color: gold; text-align: center}
```

```
  h2 {color: darkorange}
```

```
</style>
```

# Styling the Paradox

The tag which makes “The statement is false” bold will also make it red:

```
<b style="color: red">
```

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.

The image shows a classic oil painting of a smoking pipe, rendered in a realistic style with soft lighting and shadows. Below the pipe, the French phrase "Ceci n'est pas une pipe." is written in a cursive, handwritten script. The background is a plain, light-colored wall.

Figure 4.7 Styled Paradoxes page.

# Making Links

- 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***
- the address is specified in the anchor tag using the href attribute
- the highlighted text is contained between the opening and closing anchor tag

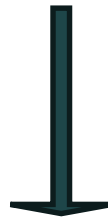
# Marking Links with Anchor Tags

- Both parts of the hyperlink are specified using 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="URL"`**, 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](http://www.biyo.com/bios/sci/russell.html)

normal text                      hyperlink reference                      anchor



**Bertrand Russell**

# Structure of the Image Tag

- An image tag specifies a file that contains an image: **``**
- **src** is the abbreviation for “source”
- **filename** is the URL of the image file
- **alt** value specifies an alternative form for the image, usually a textual description

# Structure of the Image Tag

- The **alt** tag was introduced to assist persons who are visually impaired
- 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 unavailable or loads slowly, browsers show the **alt** information

# Image File Types

- The file name needs to use the correct file extension
- .gif, .png, and .jpg are frequently used extensions


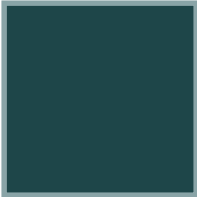
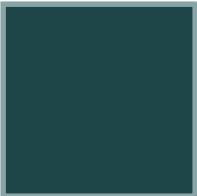
# Attributes for Image Tags

- The img tag has attributes width and height which take values in pixels
- Use one or the other to resize the image
- Using both can stretch the image

```

```

# Styling Position for Images

- Images are inserted in the page at the point in the text where the  tag is specified in  the HTML, and 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

# Styling Position for Images

- You can either position the image on the left with the text to its right, or vice versa
- To make the text flow *around* the image, use the *style attribute* in the image tag with the value "float:left" or "float:right"
- To display an image without any text around it, enclose it in paragraph tags

# Making an Image Link

- We do not have to use 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>`
- When the page displays, the usual highlighting that links receive will be used to mark the .gif as a link



# Referring to Pages and Images

- The value of the href or src attribute is a URL
- Sometimes a full URL is used.  
`<a href =  
"http://www.example.com/page.html">`
- Shorter forms are used to refer to files which are closer

# Absolute References (URLs)

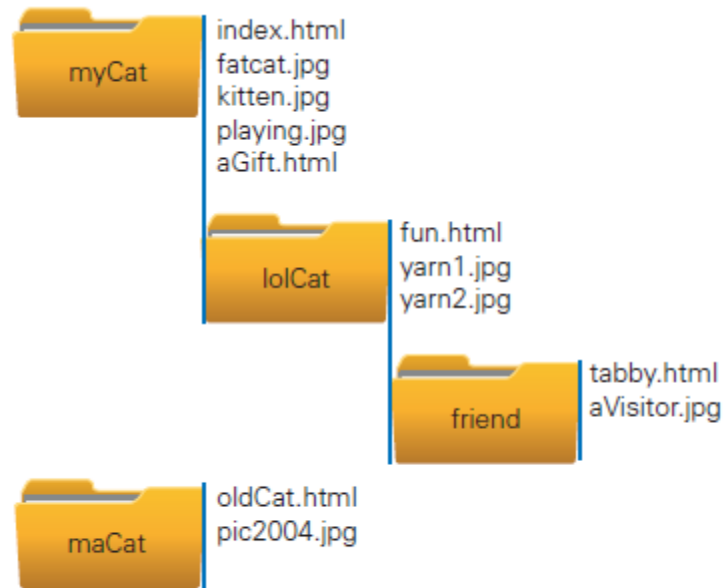
- In these anchor tags, the hyperlink reference is an ***entire*** URL
  - The Web browser needs to know how/where to find the page
- A complete URL is made from:
  - a protocol specification, ***http://***
  - a domain or IP address, ***www.biyo.com***
  - a path to the file, ***/bios/sci/russell.html***

# Relative References

- Often links refer to other Web pages on the *same site*
- These pages will be stored in the same or nearby folders
- These anchor tags use *relative references*
- A relative reference describes how to find the referenced file *relative* to the file in which the anchor or img 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
- When the referenced file is “deeper” in the directory, we simply give the path from the current folder down to the file



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

# Going Down

From index.html to

- aGift.html, use  
href="aGift.html"
- fun.html, use  
href="lolCat/fun.html"
- tabby.html, use  
href="lolCat/friend/tabby.html"

# Going “Higher” in a Hierarchy

- To go up to a folder, add `../` to the beginning of the relative URL
- Each `../` moves up one level higher in the hierarchy
- This convention originated in the Unix operating system

# Going Up

- From tabby.html to fun.html, use `href=" ../fun.html"`
- From tabby.html to pic2004.jpg `src=" ../../../maCat/pic2004.jpg"`



# Span

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

# Span

```
<span style="color:darkturquoise">Eeny</span>  
<span style="color:blue">Meeny</span>  
<span style="color:violet">Miny</span>  
<span style="color:darkmagenta">Mo</span>
```



# Lists Tags

- 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>`

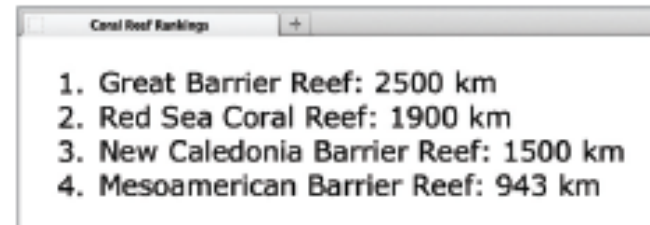
```
<ul>  
  <li>Luxo Jr.</li>  
  <li>Toy Story</li>  
  <li style="font-family:courier">  
    Monsters Inc.</li>  
  <li>Wall•E</li>  
</ul>
```



# Lists Tags

- Another list is the ordered list
- It uses the tags `<ol>` and `</ol>`
- Bullets are replaced with numbers
- The ordered list behaves just like the unnumbered list

```
<ol>  
  <li>Great Barrier Reef: 2500 km</li>  
  <li>Red Sea Coral Reef: 1900 km</li>  
  <li>New Caledonia Barrier Reef: 1500 km</li>  
  <li>Mesoamerican Barrier Reef: 943 km</li>  
</ol>
```



# Lists Tags

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

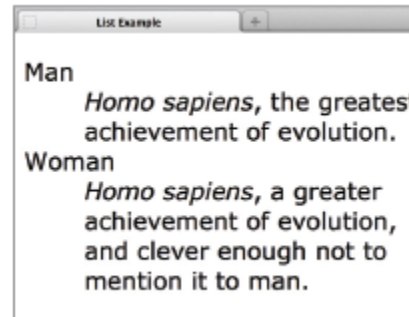
```
<ul>  
  <li>Speedo</li>  
  <li>Bikini  
    <ul>  
      <li>Top</li>  
      <li>Bottom</li>  
    </ul>  
  </li>  
  <li>Boardies</li>  
</ul>
```



# Lists Tags

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

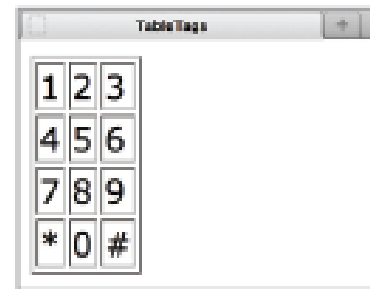
```
<dl>
  <dt> Man </dt>
  <dd><i>Homo sapiens</i>, the greatest
    achievement of evolution. </dd>
  <dt> Woman </dt>
  <dd><i>Homo sapiens</i>, a greater
    achievement of evolution, and clever
    enough not to mention it to man. </dd>
</dl>
```



# Handling Tables

- 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 browser aligns cells to form columns

```
<table border="1">  
  <tr><td>1</td><td>2</td><td>3</td></tr>  
  <tr><td>4</td><td>5</td><td>6</td></tr>  
  <tr><td>7</td><td>8</td><td>9</td></tr>  
  <tr><td>*</td><td>0</td><td>#</td></tr>  
</table>
```

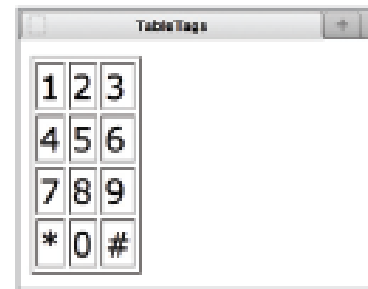


1	2	3
4	5	6
7	8	9
*	0	#

# Handling Tables

- 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>1</td><td>2</td><td>3</td></tr>  
  <tr><td>4</td><td>5</td><td>6</td></tr>  
  <tr><td>7</td><td>8</td><td>9</td></tr>  
  <tr><td>*</td><td>0</td><td>#</td></tr>  
</table>
```



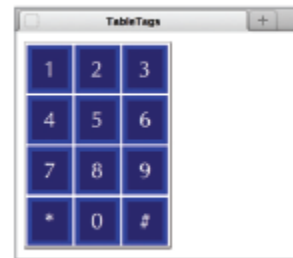
1	2	3
4	5	6
7	8	9
*	0	#



# Handling Tables

- The td element has a number of style attributes.
- Place a td specification in the <style> section, and it will control every cell.

```
td {  
    border-style:solid;  
    border-width:4px;  
    border-color:mediumblue;  
    padding:5px;  
    width:20px;  
    text-align:center;  
    font-family:optima;  
    font-size:large;  
    background-color:midnightblue;  
    color:lavender;  
}
```



1	2	3
4	5	6
7	8	9
*	0	#

# Embellishing Tables

- You can give tables captions and column headings:
- Place caption tags within the table tags around the table's caption
  - caption tags are `<caption>` and `</caption>`
- Captions are centered at the top of the table

# Embellishing Tables

- Column headings are the first row of the table
- In the “heading” row, replace the table data tags with table heading tags `<th>` and `</th>`
- Column headings display in bold
- The `colspan` attribute allows a cell to expand into multiple columns

# Embellishing Tables

```
<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>
```

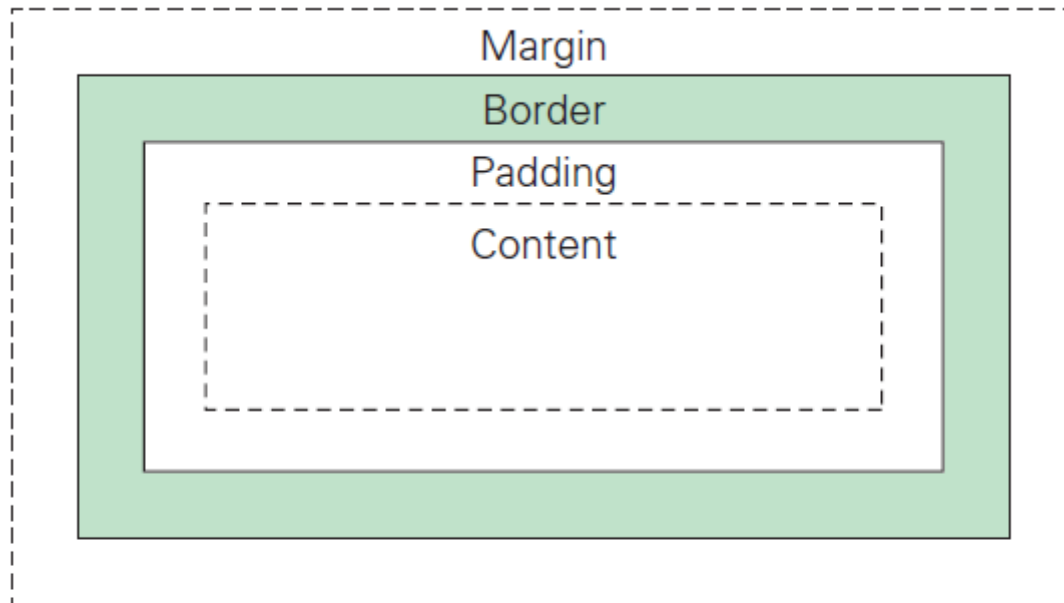


The screenshot shows a browser window titled "TableTags" with a "+" button in the top right corner. The table content is as follows:

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

# 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.

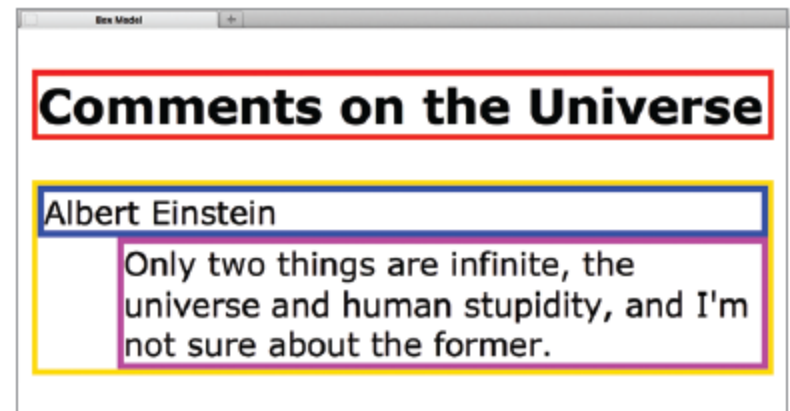
# Controlling The Box

- The margin and padding are blank space
- Content and border areas may contain visible material
- CSS can be used to set the sizes and widths
- Different sides can be set separately

```

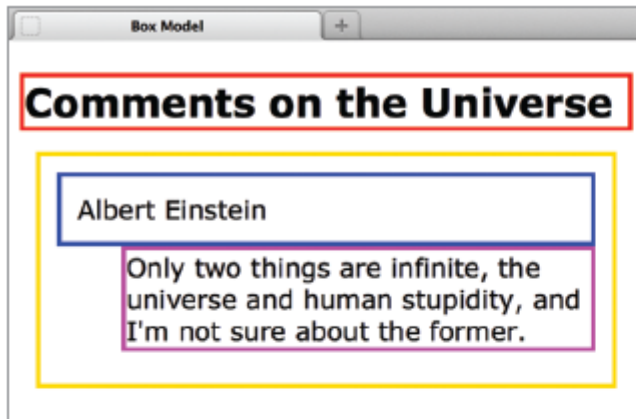
<style>
  h2 {border-style:solid; border-color:red}
  dl {border-style:solid; border-color:gold}
  dt {border-style:solid; border-color:blue}
  dd {border-style:solid; border-color:magenta}
</style>
</head>
<body>
  <h2>Comments on the Universe</h2>
  <dl>
    <dt>Albert Einstein </dt>
    <dd>Only two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

```

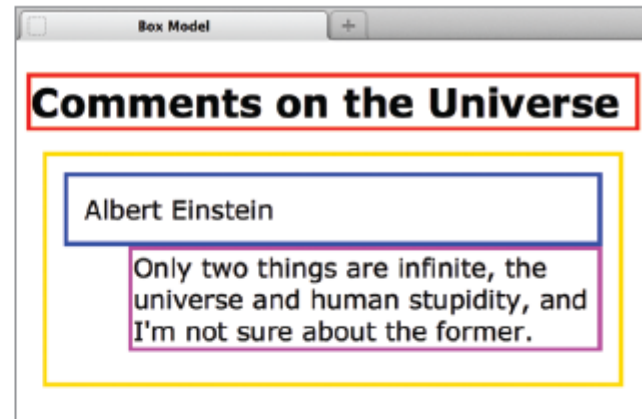


**Figure 4.10** The boxes around definitional list elements with 0-width padding and margin.

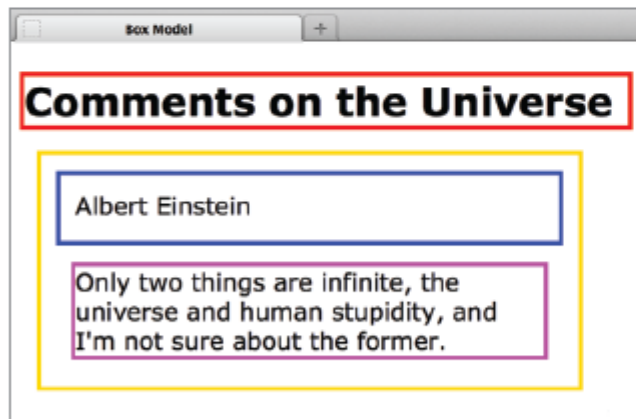




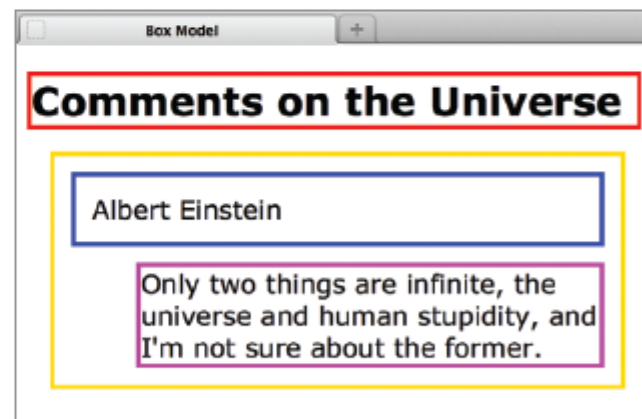
(a)



(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.