Tutorial 1

Developing a Basic Web Page

Objectives

- Learn the history of the Web and HTML
- Describe HTML standards and specifications
- Understand HTML elements and markup tags
- Create the basic structure of an HTML file
- Insert an HTML comment
- Work with block-level elements
- Create ordered, unordered, and definition lists

Objectives

- Work with inline elements
- Understand the div and span elements
- Add attributes to HTML elements
- Format page content using the style attribute
- Mark empty elements with one-sided tags
- Add an inline image to a Web page
- Work with character sets and codes

Exploring the History of the World Wide Web

- A **network** is a structure linking computers together for the purpose of sharing information and services
- Users typically access a network through a computer called a **host** or **node**
- A node that provides information or a service is called a **server**

Exploring the History of the World Wide Web

- A computer or other device that requests services from a server is called a **client**
- One of the most commonly used designs is the **client-server network**
- If the computers that make up a network are close together (within a single department or building), then the network is referred to as a **local area network (LAN)**

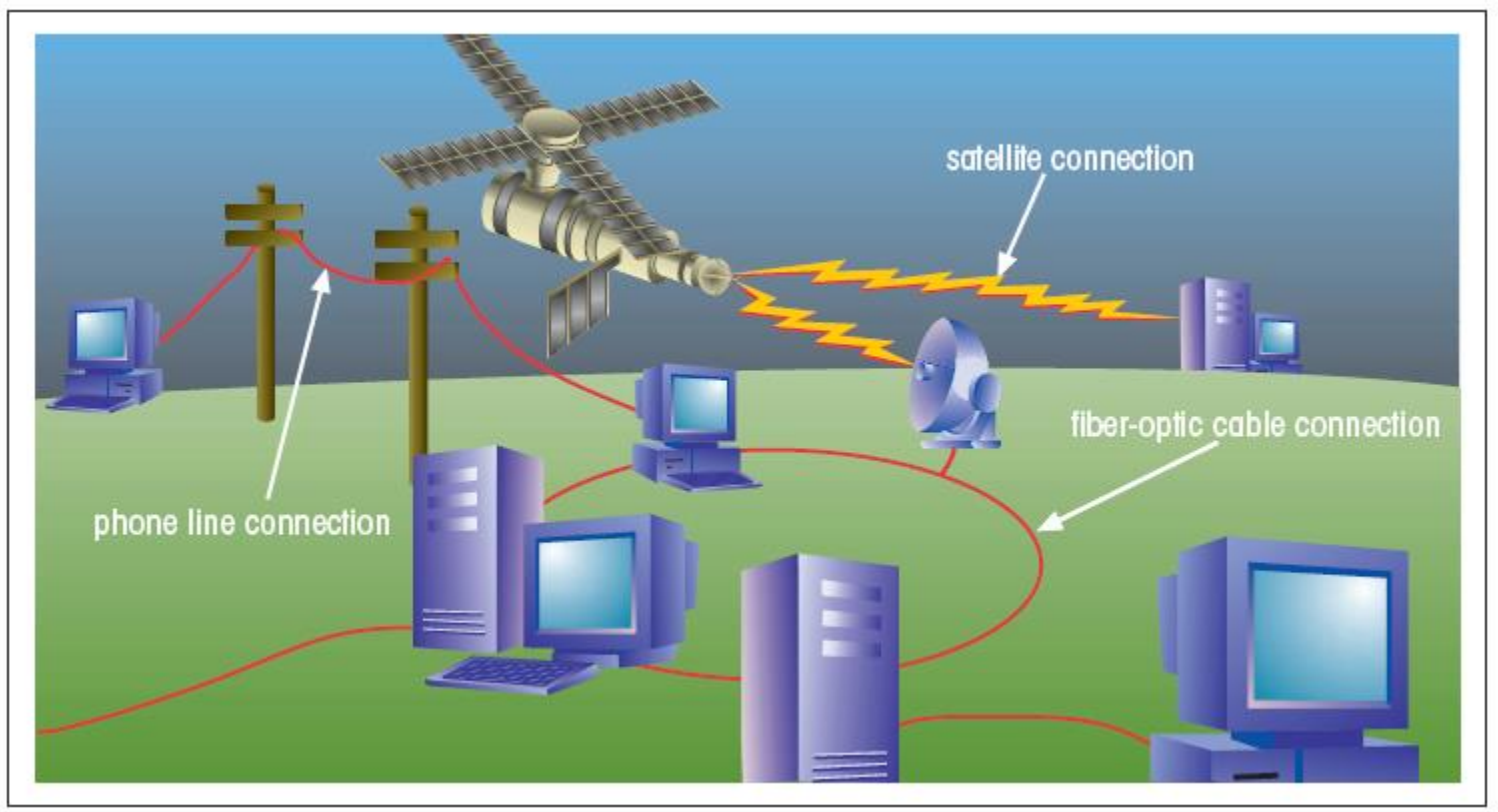
Exploring the History of the World Wide Web

- A network that covers a wide area, such as several buildings or cities, is called a **wide area network (WAN)**
- The largest **WAN** in existence is the **Internet**
- In its early days, the Internet was called **ARPANET** and consisted of two network nodes located at UCLA and Stanford, connected by a phone line

Exploring the History of the World Wide Web

- Today the Internet has grown to include an uncountable number of nodes involving computers, cell phones, PDAs, MP3 players, gaming systems, and television stations
- The physical structure of the Internet uses fiber-optic cables, satellites, phone lines, wireless access points, and other telecommunications media

Structure of the Internet



Exploring the History of the World Wide Web

- Timothy Berners-Lee and other researchers at the CERN nuclear research facility near Geneva, Switzerland laid the foundations for the **World Wide Web**, or the **Web**, in 1989
- They developed a system of interconnected **hypertext** documents that allowed their users to easily navigate from one topic to another
- **Hypertext** is a method of organizing information that gives the reader control over the order in which the information is presented

Hypertext Documents

- When you read a book, you follow a linear progression, reading one page after another
- With hypertext, you progress through pages in whatever way is best suited to you and your objectives
- Hypertext lets you skip from one topic to another

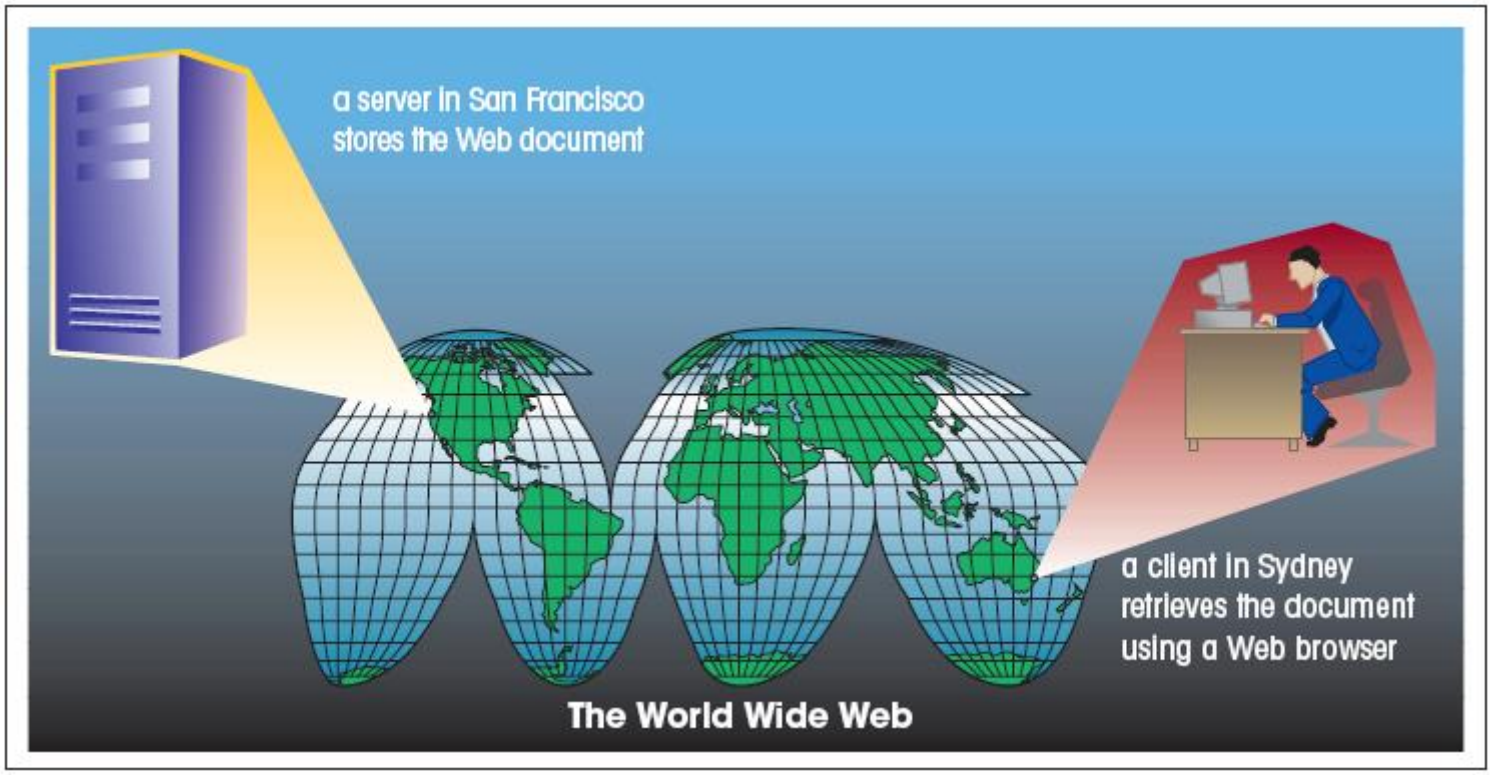
Hypertext Documents

- The key to **hypertext** is the use of **links**, which are the elements in a hypertext document that allow you to jump from one topic or document to another
- A **link** may point to another section of the same document, or to another document entirely
- A **link** can open a document on your computer, or through the Internet, a document on a computer anywhere in the world

Web Pages and Web Servers

- Each document on the World Wide Web is referred to as a **Web page**
- Web pages are stored on **Web servers,** which are computers that make Web pages available to any device connected to the Internet
- A **Web browser** retrieves the page from the Web server and renders it on the user's computer or other device
- The earliest browsers, known as **text-based browsers**, were incapable of displaying images

Web Pages and Web Servers



Introducing HTML

- A Web page is a text file written in a language called **Hypertext Markup Language**
- A **markup language** is a language that describes a document's content and structure
- HTML is not a programming language or a formatting language
- **Styles** are format descriptions written in a separate language from HTML that tell browsers how to render each element for particular devices

The History of HTML

- The first version of HTML was created using the **Standard Generalized Markup Language (SGML)**
- In the early years of HTML, Web developers were free to define and modify HTML in whatever ways they thought best
- Competing browsers introduced some differences in the language The changes were called **extensions**

The History of HTML

- A group of Web developers, programmers, and authors called the **World Wide Web Consortium**, or the **W3C**, created a set of standards or specifications that all browser manufacturers were to follow
- The **W3C** has no enforcement power
- The recommendations of the **W3C** are usually followed since a uniform approach to Web page creation is beneficial to everyone

History of HTML and XHTML

Version	Date of Release	Description
HTML 1.0	1989	The first public version of HTML which included browser support for inline images and text controls.
HTML 2.0	1995	The first version supported by all graphical browsers. It introduced interactive form elements such as option buttons and text boxes. A document written to the HTML 2.0 specification is compatible with almost all browsers on the World Wide Web.
HTML 3.0	1996	A proposed replacement for HTML 2.0 that was never widely adopted.
HTML 3.2	1997	This version included additional support for creating and formatting tables and expanded the options for interactive form elements. It also supported limited programming using scripts.
HTML 4.01	1999	This version added support for style sheets to give Web designers greater control over page layout. It added new features to tables and forms and provided support for international features. This version also expanded HTML's scripting capability and added increased support for multimedia elements.
HTML 5.0	not yet released	This version supports elements that reflect current Web usage, including elements for Web site navigation and indexing for use with search engines. This version also removes support for purely presentational elements because those effects can be better handled with styles.
XHTML 1.0	2001	This version is a reformulation of HTML 4.01 in XML and combines the strength of HTML 4.0 with the power of XML. XHTML brings the rigor of XML to Web pages and provides standards for more robust Web content on a wide range of browser platforms.
XHTML 1.1	2002	A minor update to XHTML 1.0 that allows for modularity and simplifies writing extensions to the language.
XHTML 2.0	not yet released	The latest version, designed to remove most of the presentational features left in HTML. XHTML 2.0 is not backward compatible with XHTML 1.1.
XHTML 5.0	not yet released	A version of HTML 5.0 written under the specifications of XML, unlike XHTML 2.0, XHTML 5.0 will be backward-compatible with XHTML 1.1.

The History of HTML

- Older features of HTML are often **deprecated**, or phased out, by the W3C That does not mean you can't continue to use them—you may need to use them if you are supporting older browsers
- Current Web developers are increasingly using **XML (Extensible Markup Language)**
- **XML (Extensible Markup Language)** is a metalanguage like SGML, but without SGML's complexity and overhead

The History of HTML

- **XHTML (Extensible Hypertext Markup Language)** is a stricter version of HTML and is designed to confront some of the problems associated with the different and competing versions of HTML
- **XHTML** is also designed to better integrate **HTML** with other markup languages such as **XML**
- **HTML** will not become obsolete anytime soon

The History of HTML

- **XHTML 2.0** is still in the draft stage, and is not backward-compatible with earlier versions of HTML and XHTML
- HTML 5 is being developed under the XML specifications as **XHTML 5.0**

Writing HTML Code

- Become well-versed in the history of HTML
- Know your market
- Test your code on several different browsers and browser versions
- Read the documentation on the different versions of HTML and XHTML

Tools for Creating HTML Documents

- Basic text editor such as Windows Notepad
- **HTML Converter** – translates formatted text into HTML code
  - Can create the source document in a word processor and then convert it
  - HTML code created using a converter is often longer and more complicated than necessary, resulting in "bloated" code

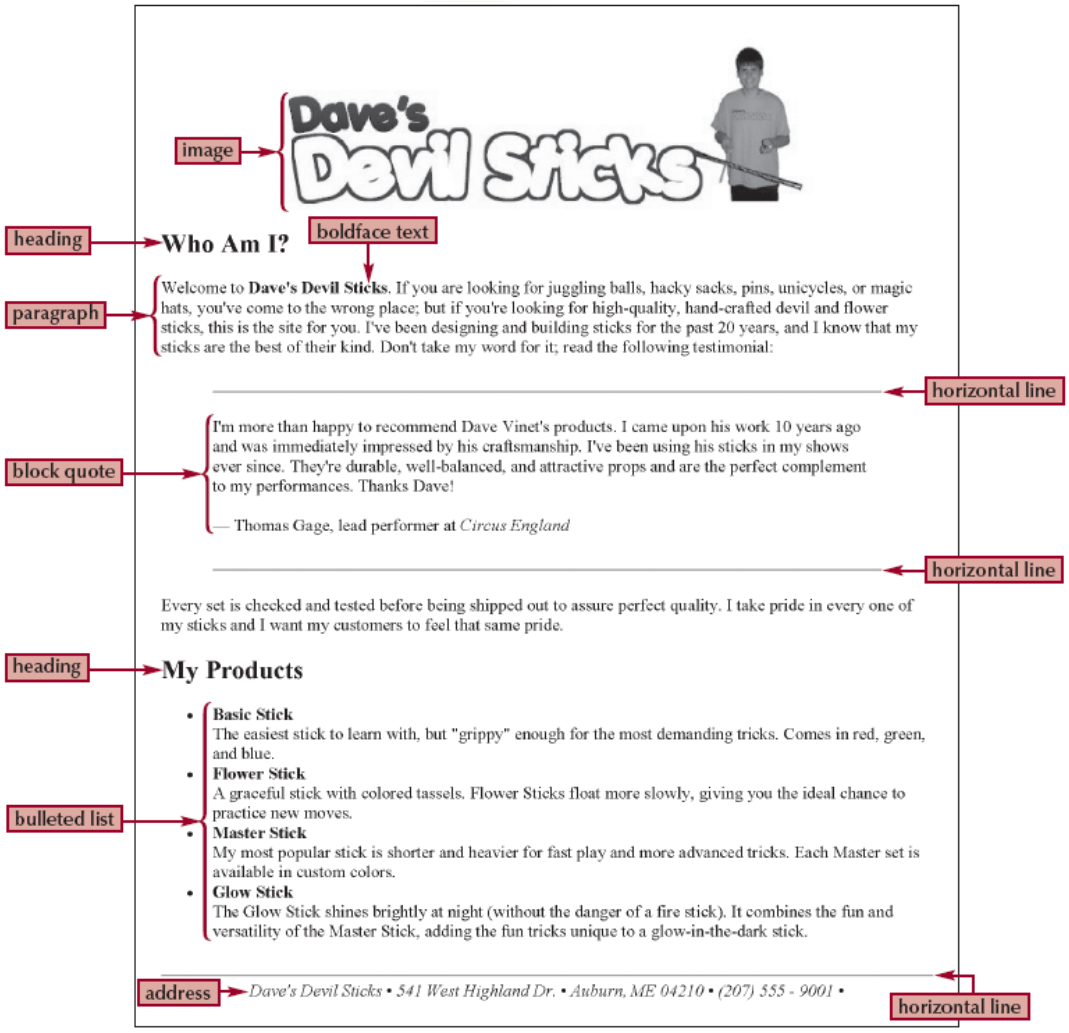
Tools for Creating HTML Documents

- **HTML Editor** – helps you create an HTML file by inserting HTML codes for you as you work
  - They can save you a lot of time and help you work more efficiently
  - Advantages and limitations similar to those of HTML converters
  - Allow you to set up a Web page quickly
  - Will usually still have to work with HTML code to create a finished document

Creating an HTML Document

- Plan out your Web page before you start coding
- Draw a planning sketch or create a sample document using a word processor
- Preparatory work can weed out errors or point to potential problems

Creating an HTML Document



Creating an HTML Document

- In planning, identify a document's various elements An **element** is a distinct object in the document, like a paragraph, a heading, or a page's title
- Formatting features such as **boldfaced** font, and *italicized* text may be used

Marking Elements with Tags

- The core building block of HTML is the **tag**, which marks the presence of an element
- A **two-sided tag** is a tag that contains some document content General syntax for a two-sided tag:

<element>content</element>

Marking Elements with Tags

- A two-sided tag's opening tag ( ) and closing tag ( ) should completely enclose its content
- Elements can contain other elements
  - Tags cannot overlap

** Welcome to Dave's Devil Sticks **

The Structure of an HTML File

- The opening **<html>** tag marks the start of an HTML document, and the closing **</html>** tag tells a browser when it has reached the end of that HTML document
  - <html> marks the **root element**
- Anything between these two tags makes up the document content, including all other elements, text, and comments

The Structure of an HTML File

- An HTML document is divided into two main sections: the **head** and the **body**
- The **head element** contains information about the document, for example the document title or the keywords
- The content of the **head** element is not displayed within the Web page

The Structure of an HTML File

- The **body element** contains all of the content to appear on the Web page
- The **body element** can contain code that tells the browser how to render the content
- The **title element** contains the page's title A document's title is usually displayed in the browser's title bar

Converting an HTML Document into XHTML

- There is considerable overlap between HTML and XHTML
- You can convert an HTML file into an XHTML file by replacing the opening <html> tag with the following three lines of code:
  - <?xml version="10" encoding="UTF-8" standalone="no" ?>
  - <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 10 Strict//EN“ “http://wwww3org/TR/xhtml1/DTD/xhtml11-strictdtd">
  - <html xmlns=http://wwww3org/1999/xhtml>

Adding Comments

- The **comment tag** adds notes to your HTML code

<!-- comment -->

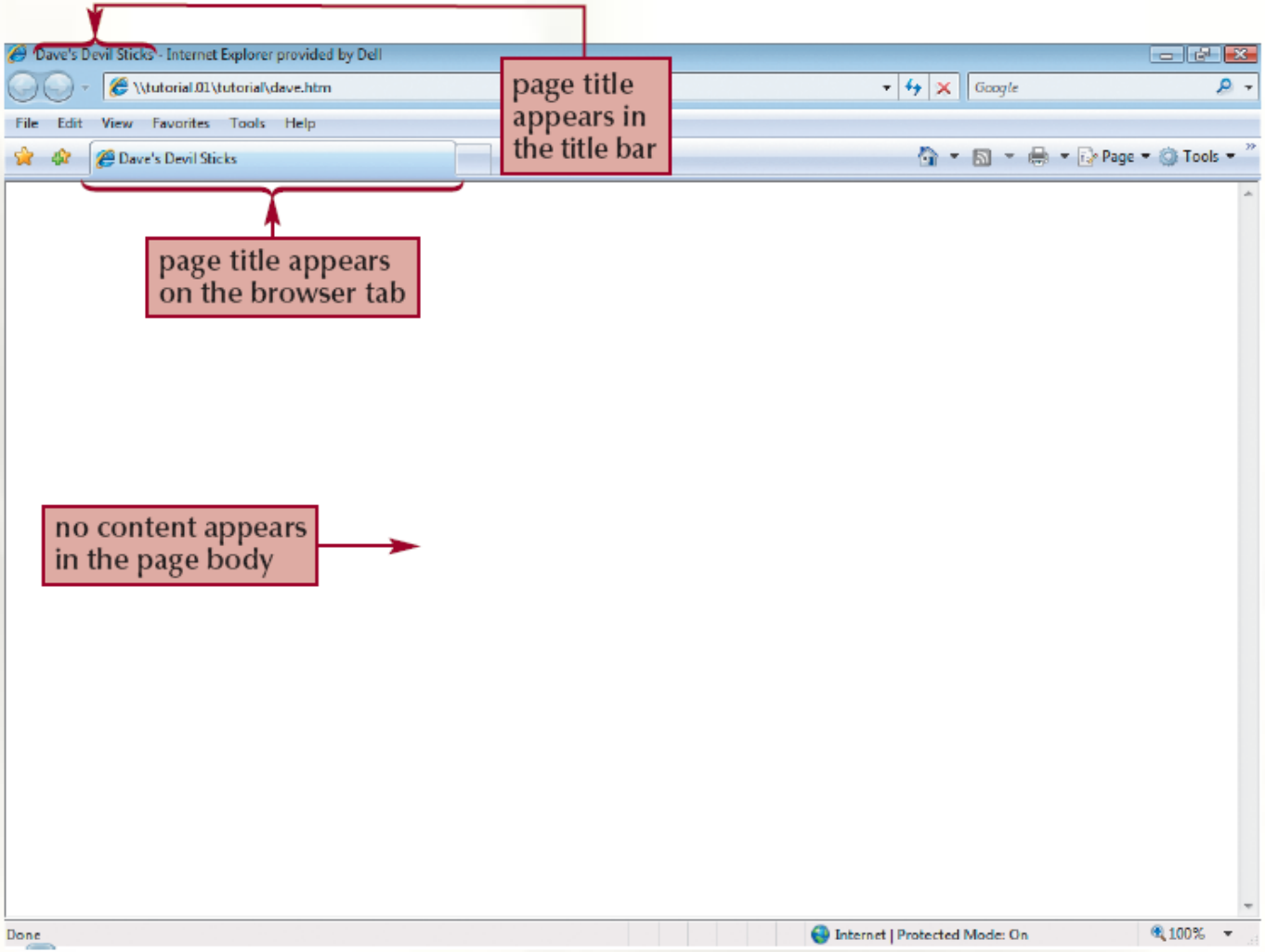
- Comments can be spread over several lines
- Comments are useful in documenting your HTML code for yourself and others

Adding Comments

Displaying an HTML File

- As you continue modifying the HTML code, you should occasionally view it with your Web browser to verify that you have not introduced any errors
- You may want to view the results using different browsers to check for compatibility

Displaying an HTML File



Working with Block-Level Elements

- **Block-level** elements are elements that contain content that is viewed as a distinct block within the Web page
- **Heading elements** are block-level elements that contain the text of main headings on the Web page
  - <h*n*>*content*<h*n*>
  - *n* is an integer between 1 and 6
    - <h1> is the largest heading
    - <h6> is the smallest heading

Marking Block-Level Elements

- To mark a heading, enter

  <h*n*>*content*</h*n*>

  where *n* is an integer from 1 to 6 and *content* is the text of heading
- To mark a paragraph, enter

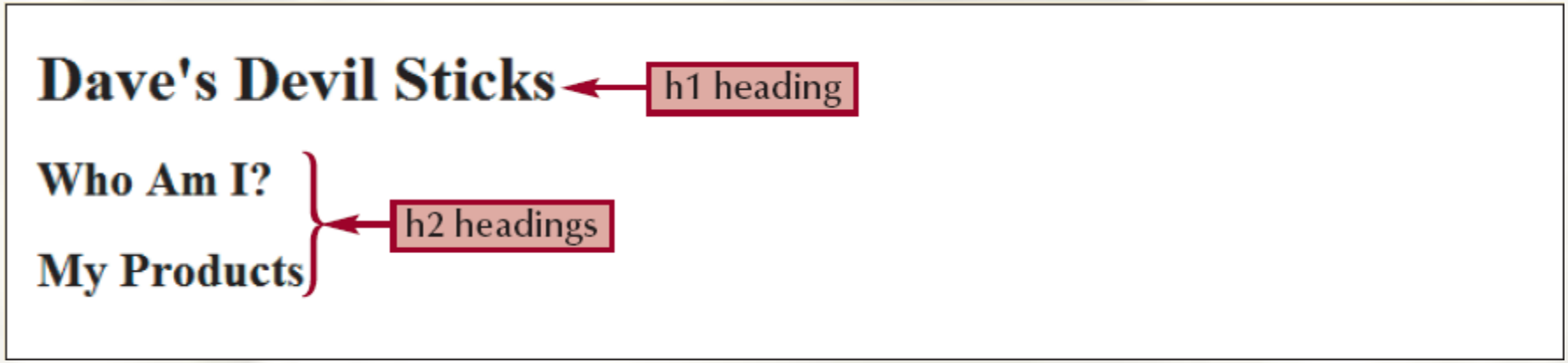
  <p>*content*</*p*>
- To mark a block quote, enter

  <blockquote>*content*</blockquote>
- To mark a generic block-level element, enter

  content

Adding <h1> and <h2> Markup Tags

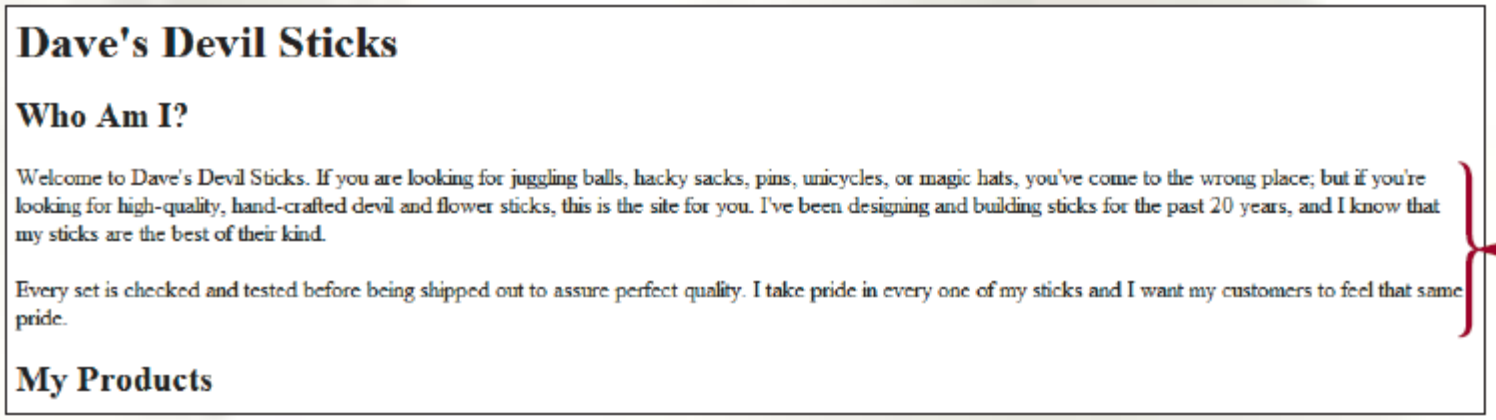
```
<body>
   <h1>Dave's Devil Sticks</h1>
   <h2>Who Am I?</h2>
   <h2>My Products</h2>
</body>
```



Marking Paragraph Elements

paragraphs

```
<body>
   <h1>Dave's Devil Sticks</h1>
   <h2>Who Am I?</h2>
   <p>Welcome to Dave's Devil Sticks. If you are looking for juggling balls,
      hacky sacks, pins, unicycles, or magic hats, you've come to the wrong
      place; but if you're looking for high-quality, hand-crafted devil and
      flower sticks, this is the site for you. I've been designing and building
      sticks for the past 20 years, and I know that my sticks are the best of
      their kind.</p>

   <p>Every set is checked and tested before being shipped out to assure perfect
      quality. I take pride in every one of my sticks and I want my customers to
      feel that same pride.</p>

   <h2>My Products</h2>
</body>
```



paragraphs

White Space and HTML

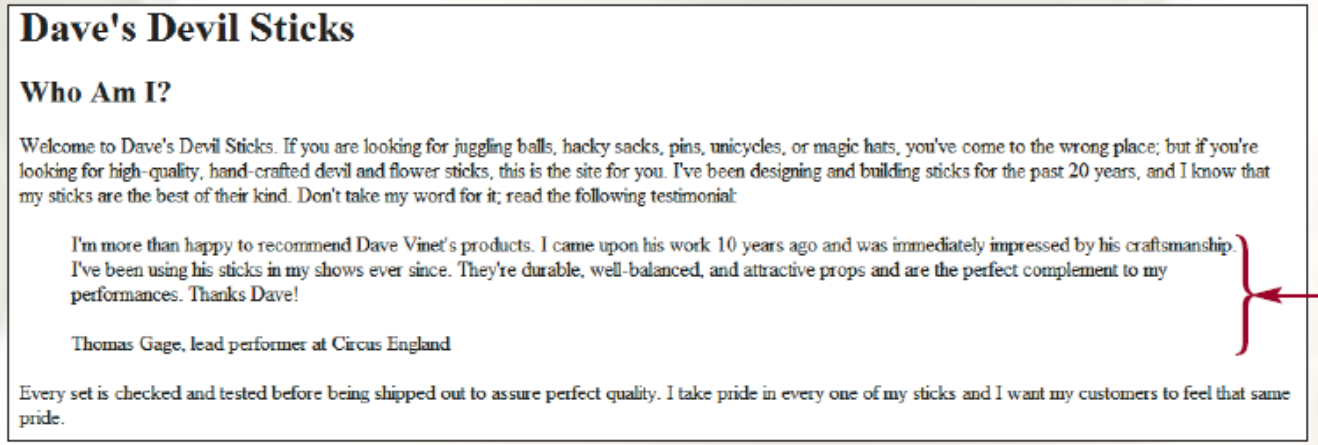
- HTML file documents are composed of text characters and **white space**
- **White space** is the blank space, tabs, and line breaks within the file
- HTML treats each occurrence of **white space** as a single blank space
- You can use **white space** to make your document more readable

Marking a Block Quote

- The syntax for making an extended quote is
  - <blockquote>content</blockquote>

```
<p>Welcome to Dave's Devil Sticks. If you are looking for juggling balls,
   hacky sacks, pins, unicycles, or magic hats, you've come to the wrong
   place; but if you're looking for high-quality, hand-crafted devil and
   flower sticks, this is the site for you. I've been designing and building
   sticks for the past 20 years, and I know that my sticks are the best of
   their kind. Don't take my word for it; read the following testimonial:</p>

<blockquote>
   <p>I'm more than happy to recommend Dave Vinet's products. I came upon his
      work 10 years ago and was immediately impressed by his craftsmanship.
      I've been using his sticks in my shows ever since. They're durable,
      well-balanced, and attractive props and are the perfect complement to
      my performances. Thanks Dave!</p>

   <p>Thomas Gage, lead performer at Circus England</p>
</blockquote>

<p>Every set is checked and tested before being shipped out to assure perfect
   quality. I take pride in every one of my sticks and I want my customers to
   feel that same pride.</p>
```

block quote



block quote

Marking a List

- HTML supports three kinds of lists: **ordered, unordered, and definition**
- You use an **ordered list** for items that must appear in a numerical order
- You use an **unordered list** for items that do not need to occur in any special order
- One **list** can contain another list This is called a nested list

Marking a List

Marking Lists

- To mark an ordered list, enter

```
<ol>
   <li>item1</li>
   <li>item2</li>
...
</ol>
```

where *item1*, *item2*, and so forth are the items in the list.

- To mark an unordered list, use

```
<ul>
   <li>item1</li>
   <li>item2</li>
...
</ul>
```

- To mark a definition list, use

```
<dl>
   <dt>term1</dt>
   <dd>description1</dd>
   <dt>term2</dt>
   <dd>description2a</dd>
   <dd>description2b</dd>
...
</dl>
```

where *term1*, *term2*, etc. are the terms in the list and *description1*, *description2a*, *description2b*, etc. are the descriptions associated with each term.

Creating a Definition List

- The **definition list** contains a list of terms, each followed by the term's description
- Web browsers typically display the definition description below the definition term and slightly indented:

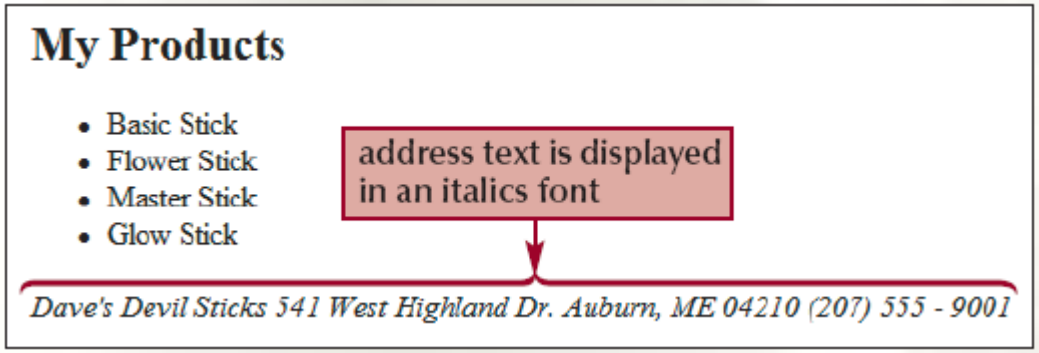
Basic Stick

Easiest stick to learn

Using Other Block-Level Elements

- HTML supports the **address element** to indicate contact information Most browsers display an address element in an italicized font, and some right-justify or indent addresses

```
    <address>Dave's Devil Sticks
             541 West Highland Dr.
             Auburn, ME 04210
             (207) 555 - 9001
    </address>
</body>

</html>
```

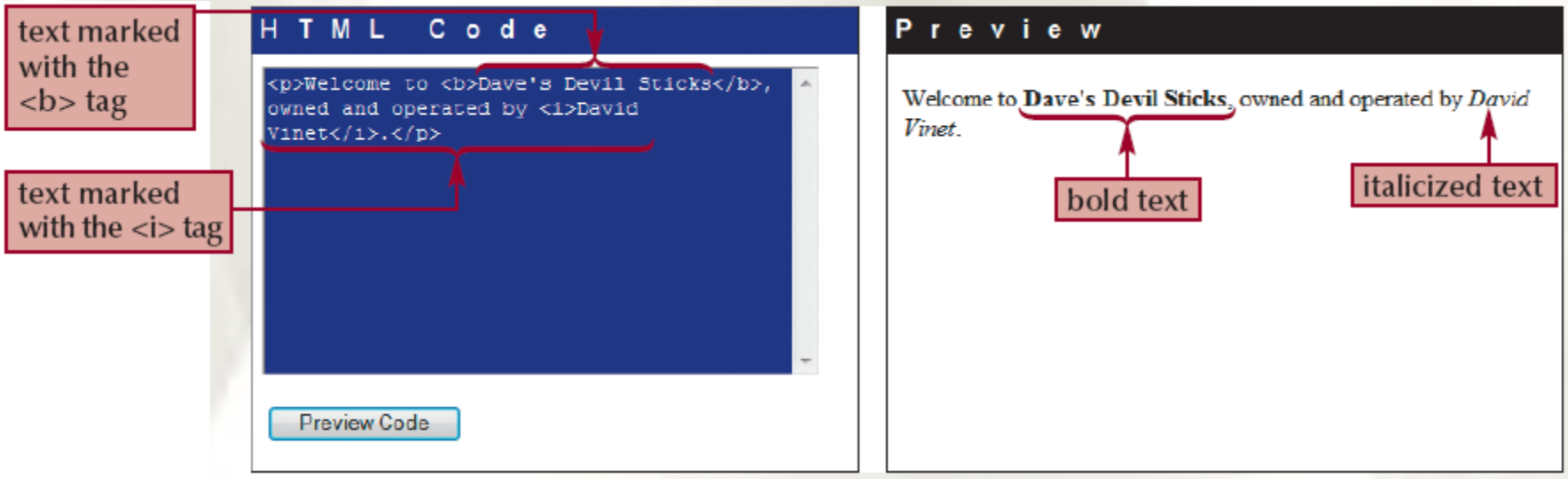


Using Other Block-Level Elements

Block-Level Element	Marks	Usual Visual Appearance
`<address> ... </address>`	Contact information	*Italicized text*
`<blockquote> ... </blockquote>`	An extended quotation	Plain text indented from the left and right
`<center> ... </center>`	Text horizontally centered with the block (**deprecated**)	Plain text, centered
`<dd> ... </dd>`	A definition description	Plain text
`<dir> ... </dir>`	A multicolumn directory list (**deprecated**)	Plain text
`<div> ... </div>`	A generic block-level element	Plain text
`<dl> ... </dl>`	A definition list	Plain text
`<dt> ... </dt>`	A definition term from a definition list	Plain text
`<hn> ... </hn>`	A heading where *n* is a value from 1 to 6 with h1 as the most prominent heading and h6 the least prominent	**Boldfaced text of various font sizes**
`<li> ... </li>`	A list item from an ordered or unordered list	Bulleted or numbered text
`<menu> ... </menu>`	A single column menu list (**deprecated**)	Plain text
`<ol> ... </ol>`	An ordered list	Plain text
`<p> ... </p>`	A paragraph	Plain text
`<pre> ... </pre>`	Preformatted text, retaining all white space and special characters	`Fixed width text`
`<ul> ... </ul>`	An unordered list	Plain text

Working with Inline Elements

- An **inline element** marks a section of text within a block-level element
- Often used to format characters and words
  - Also referred to as **character formatting elements**



Working with Inline Elements

Inline Element	Marks	Usual Visual Appearance
`<abbr> ... </abbr>`	An abbreviation	Plain text
`<acronym> .. </acronym>`	An acronym	Plain text
`<b> ... </b>`	Boldfaced text	**Boldfaced text**
`<big> ... </big>`	Big text	Larger text
`<cite> ... </cite>`	A citation	*Italicized text*
`<code> ... </code>`	Program code	`Fixed width text`
`<del> ... </del>`	Deleted text	~~Strikethrough text~~
`<dfn> ... </dfn>`	A definition term	*Italicized text*
`<em> ... </em>`	Emphasized content	*Italicized text*
`<i> ... </i>`	Italicized text	*Italicized text*
`<ins> ... </ins>`	Inserted text	Underlined text
`<kbd> ... </kbd>`	Keyboard-style text	`Fixed width text`
`<q> ... </q>`	Quoted text	"Quoted text"
`<s> ... </s>`	Strikethrough text (**Deprecated**)	~~Strikethrough text~~
`<samp> ... </samp>`	Sample computer code	`Fixed width text`
`<small> ... </small>`	Small text	Smaller text
`<span> ... </span>`	A generic inline element	Plain text
`<strike> ... </strike>`	Strikethrough text (**Deprecated**)	~~Strikethrough text~~
`<strong> ... </strong>`	Strongly emphasized content	**Boldfaced text**
`<sub> ... </sub>`	Subscripted text	Subscripted text
`<sup> ... </sup>`	Superscripted text	Superscripted text
`<tt> ... </tt>`	Teletype text	`Fixed width text`
`<u> ... </u>`	Underlined text (**Deprecated**)	Underlined text
`<var> ... </var>`	Programming variables	*Italicized text*

Logical Elements vs Physical Elements

- A logical element describes the nature of the enclosed content, but not necessarily how that content should appear
- A physical element describes how content should appear, but doesn't indicate the content's nature
- You should use a logical element that accurately describes the enclosed content whenever possible, and use physical elements only for general content

Using Element Attributes

- Many tags contain attributes that control the use, behavior, and in some cases the appearance, of elements in the document
- Attributes are inserted within the tag brackets

<element attribute1="value1" attribute2="value2" …>content</element>

The Style Attribute

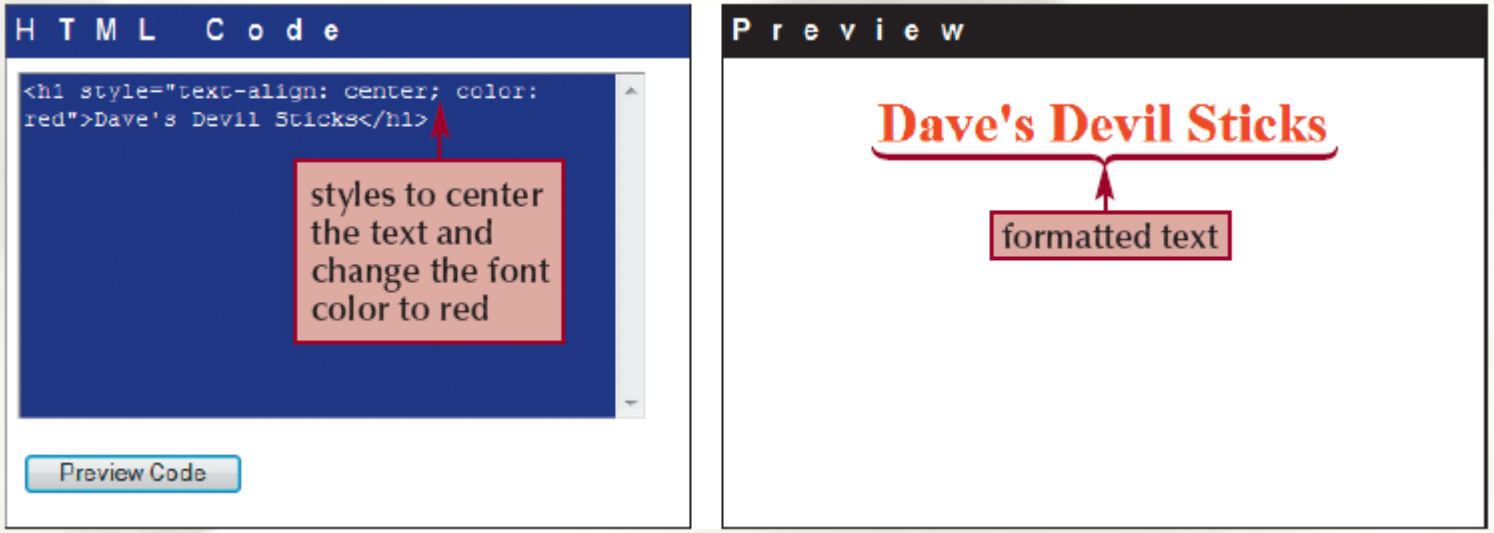
- Use the **style attribute** to control the appearance of an element, such as text alignment
- The **text-align style** tells the browser how to horizontally align the contents of an element
- The **color style** tells the browser to render the text in a certain color
- **Presentational attributes** specify exactly how the browser should render an element

The Style Attribute

Applying the Style Attribute

- To add the style attribute, in the opening tag enter
  `style="name1:value1; name2:value2; ..."`
  where *name1*, *name2*, etc. are style names and *value1*, *value2* and so forth are the values of those styles.
- To center text horizontally, use
  `style="text-align: alignment"`
  where *alignment* is left, right, center, or justify.
- To set the font color, use
  `style="color: color"`
  where *color* is a color name.

The Style Attribute



Working with Empty Elements

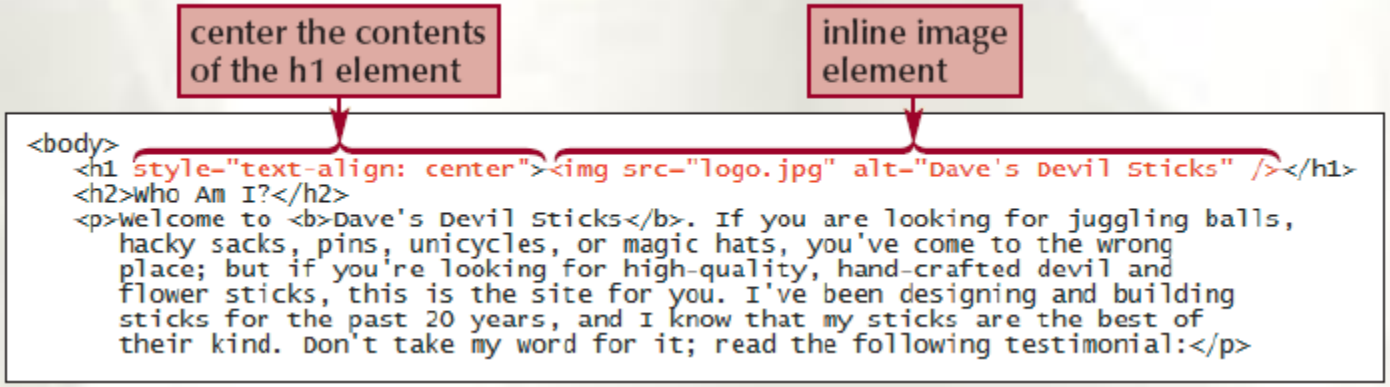
- An **empty element** contains no content
- Empty elements appear in code as **one-sided tags**
  - <element />
- The one-sided tag to mark a line break is
  -
- The horizontal rule element places a horizontal line across the Web page
  - <hr />

Working with Empty Elements

```
    <blockquote>
       <hr />
       <p>I'm more than happy to recommend Dave Vinet's products. I came upon his
          work 10 years ago and was immediately impressed by his craftsmanship.
          I've been using his sticks in my shows ever since. They're durable,
          well-balanced, and attractive props and are the perfect complement to
          my performances. Thanks Dave!</p>

       <p>Thomas Gage, lead performer at <i>Circus England</i></p>
       <hr />
    </blockquote>

    <p>Every set is checked and tested before being shipped out to assure perfect
       quality. I take pride in every one of my sticks and I want my customers to
       feel that same pride.</p>

    <h2>My Products</h2>
    <ul>
       <li><b>Basic Stick</b><br />
              The easiest stick to learn with, but "grippy" enough for the most demanding
              tricks. Comes in red, green, and blue.</li>
       <li><b>Flower Stick</b><br />
              A graceful stick with colored tassels. Flower sticks float more slowly, giving
              you the ideal chance to practice new moves.</li>
       <li><b>Master Stick</b><br />
              My most popular stick is shorter and heavier for fast play and more advanced
              tricks. Each Master set is available in custom colors.</li>
       <li><b>Glow Stick</b><br />
              The Glow Stick shines brightly at night (without the danger of a fire stick).
              It combines the fun and versatility of the Master Stick, adding the fun tricks
              unique to a glow-in-the-dark stick.</li>
    </ul>

    <hr />

    <address style="text-align: center">Dave's Devil Sticks
             541 West Highland Dr.
             Auburn, ME 04210
             (207) 555 - 9001
    </address>
</body>
```

Working with Empty Elements

- To display a graphic, you insert an **inline image** into the page An **inline image** displays a graphic image located in a separate file within the page
  -



Working with Empty Elements

Working with Character Sets and Special Characters

- **Character sets** come in a wide variety of sizes, based on the number of symbols required for communication in the chosen Language
  - **ASCII (American Standard Code for Information Interchange)**
  - **Latin-1**
  - **ISO 8859-1**
  - **Unicode**
  - **UTF-8**

Working with Character Sets and Special Characters

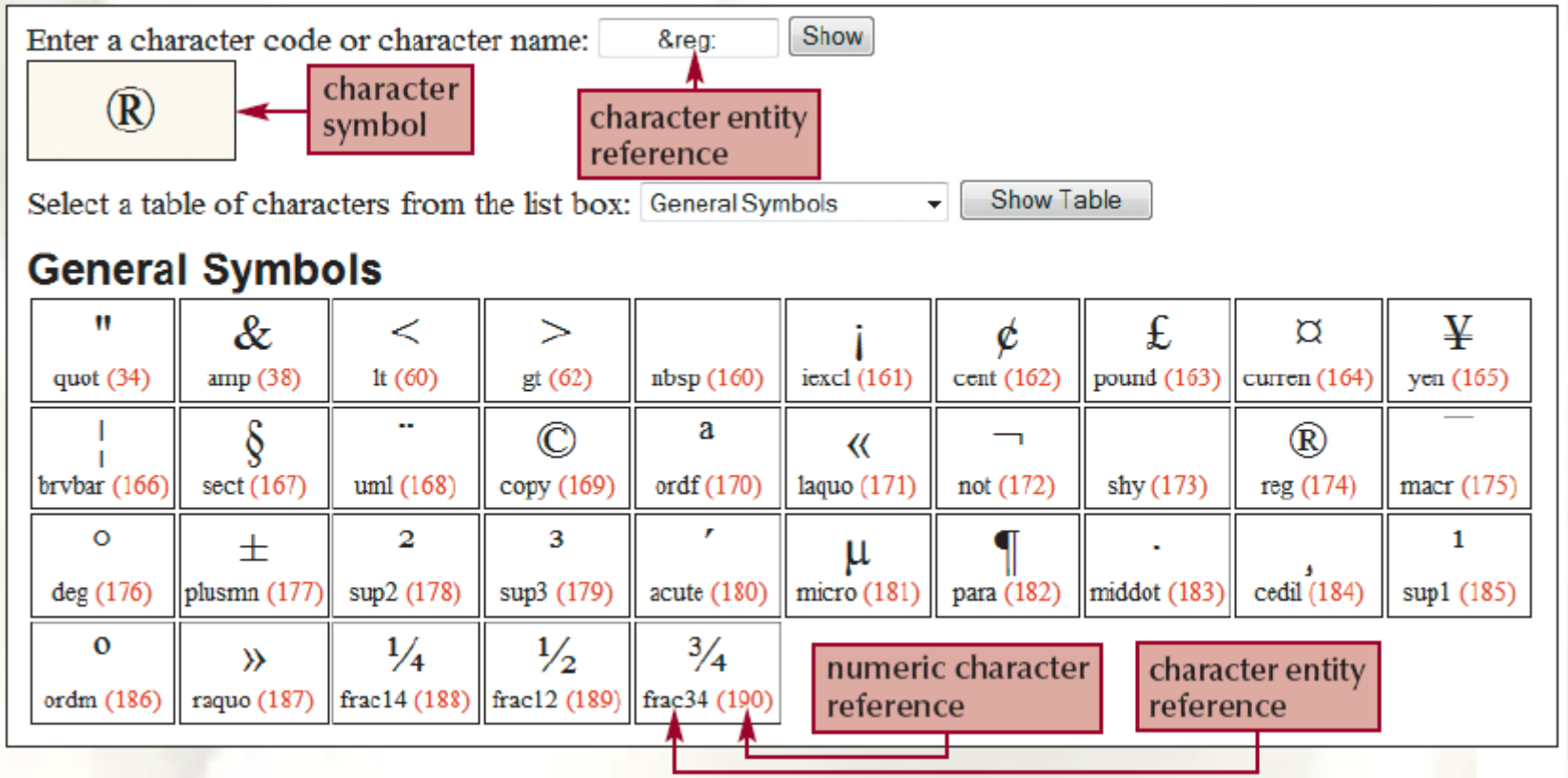
- To store a character set, browsers need to associate each symbol with a number in a process called **character encoding**
- Another way to insert a special symbol is to use a **character entity reference**, in which a short memorable name is used in place of the numeric character reference

Working with Character Sets and Special Characters

Inserting Character Codes

- To insert a character based on a numeric character reference, use
  `&#code;`
  where *code* is the character code number.
- To insert a character based on the character entity reference, use
  `&char;`
  where *char* is the name assigned to the character.
- To insert a nonbreaking space, use
  ` `
- To insert the < symbol, use
  `<`
- To insert the > symbol, use
  `>`

Working with Character Sets and Special Characters



Tutorial Summary

- Create a basic Web page using HTML
- Concepts and history surrounding networks and the development of the World Wide Web
- History of HTML
- Creation of a simple Web age
- Block-level elements
- Inline elements
- Element attributes
- Character sets and special character symbols