Chapter 2

Webpage Markup with HTML5
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
     lang="en" xml:lang="en">
<head>
<meta charset="utf-8"/>
<title>Great Company: homepage</title>
<!--[!-- other head elements as appropriate -->
</head>
<body>  <!--[!-- page content begin -->
 . . .
  . . .
<!--[!-- page content end -->  </body>
</html>
1. Displayed in the *title bar* of the browser window
2. Used in making a bookmark for the page
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en"
xml:lang="en">
<head>
<meta charset="utf-8"/>
<title>My Sample Webpage</title>
</head>
<body style="background-color: cyan; margin: 50px">
<h2>Hi everybody!</h2>
<p>My Name is (put your name here) and today is 
<time>(put in the date yyyymm-dd)</time>.
</p>
<p>HTML5 is cool.</p></body></html>

Demo: Ex: FirstPage
Hi everybody!

My Name is Paul Wang and today is June 6, 2012.

HTML5 is cool.
HTML5 Elements

- There are more than 100 different elements in HTML5.
- Elements for meta information are placed in the `<head>` element.
- Elements for page content are placed in the `<body>` element.
- HTML5 distinguishes between two broad types of content elements:
  1. *flow elements* that occupy their own vertical space in a page and *phrasing elements* that act like words and phrases. Flow elements act like paragraphs, lists, and tables and can contain other flow elements, phrasing elements, and texts.
  2. Phrasing elements can contain other phrasing elements and texts.
HTML5 Elements Classified

- **Top-level elements**: `html`, `head`, and `body`.

- **Head elements**: elements placed inside `head`, including `title` (page title), `style` (rendering style), `link` (related documents), `meta` (data about the document), `base` (URL of the document), and `script` (client-side scripting). These elements are not part of the page display.

- **Block-level elements**: flow elements behaving like paragraphs, including `article`, `h1–h6` (headings), `header`, `footer`, `section`, `p` (paragraph), `figure`, `canvas` (dynamic drawing area), `pre` (preformatted text), `div` (designated block), `ul`, `ol`, `dl` (lists), `table` (tabulation), `form` (user-input forms), and `video` (video). A block element occupies 100% of the available width to it and will be *stacked vertically* with preceding and subsequent block elements. In other words, when displayed, a block-level (or simply block) element
always starts on a new line, and any element immediately after the block element also begins on a new line.

- **Inline elements**: phrasing elements behaving like words, characters, or phrases that flow horizontally to fill the available width before starting new lines. Usually, inline elements are placed within block elements. Inline elements include `a` (anchor or link), `audio` (sound), `br` (line break), `code` (computer code), `img` (picture or graphics), `em` (emphasis), `nav` (navigation), `samp` (sample output), `span` (designated inline scope), `strong` (strong emphasis), `sub` (subscript), `sup` (superscript), `time` (time/date), and `var` (variable name).
In an HTML5 document, certain characters, such as < and &, are used for markup and must be *escaped* to appear literally. HTML provides *entities* (*escape sequences*) to introduce such characters into a webpage. For example, the entity &lt; gives < and &amp; gives &.

Characters not on the regular keyboard can also be included directly or using HTML5-defined character references.
Evolution of HTML

• In 1989, Tim Berners-Lee defined a very simple version of HTML based on SGML. The first common standard for HTML was HTML 3.2 (1997).

• HTML 4.01 became a W3C (the World-Wide Web Consortium) recommendation in December 1999. HTML 4.01 begins to clearly separate the document structure and document presentation and specifies a clear relationship between HTML and client-side scripting (JavaScript).

• In January 2000, W3C released XHTML 1.0 as an XML reformulation of HTML 4.01. XHTML 1.0 is basically HTML 4.01 written under the strict XML syntax.
  – XHTML elements can be used together with other elements defined by XML.
XHTML pages can be processed easily by any XML tool.

- HTML5 combines the XHTML, HTML 4, and CSS3 standards, introduces new elements and APIs, as well as incorporates MathML (Mathematical Markup Language) and SVG (Scalable Vector Graphics) into HTML.

- HTML5 can also easily be written in an XML compliant way. The release of the HTML5 standard promises to bring significant advantages to Web developers and benefits to end users.
<table>
<thead>
<tr>
<th>HTML 4</th>
<th>CSS3</th>
<th>XHTML</th>
<th>New Elements</th>
<th>APIs</th>
<th>MathML</th>
<th>SVG</th>
</tr>
</thead>
</table>

- HTML5 Integration
- HTML 4
- CSS3
- XHTML
- New Elements
- APIs
- MathML
- SVG
1. New elements such as header, footer, article, section, menu, nav, and aside to better organize webpages.

2. With the new audio and video elements, HTML5 makes sound and video media as easy to place in a webpage as a still image, thus eliminating the need for proprietary technologies, such as Flash and Silverlight, or browser plug-ins.

3. With the new canvas dynamic drawing area element, interactive graphics and animation can easily be deployed on the client side with JavaScript control.

4. Browser support for drag-and-drop API.

5. Browser support for form input and input checking.

6. Websites can list files to be cached by the browser for offline
use, either for browsing offline or for supporting a Web application running offline.

7. With HTML5 Web Storage, Web applications can store sizable data (up to 5MB per Web domain) locally on the browser side. Such local storage can be *per session* (lost if browser is closed) or persistent (not lost even if you close the browser or shut down the computer) but private (not transmitted back to the Web).

8. Browser support for editing (by the end user) of webpage content.

9. Native support of mathematical formulas and 2D graphics markup with MathML and SVG, respectively.

10. Enabling location-dependent information presentation with *geolocation* API.
Webpage Syntax

- All tags begin with < and end with >. The tag name is given immediately following the leading <. Make sure the tag is spelled correctly. Unrecognized tags are ignored by browsers. Any attributes are given following the tag name in the form:

  `<tag attribute_1 ="value" attribute_2 ="value" ... >`

You may use single quotes ('), instead of double quotes ("), for the value part of any attribute. Be careful; forgetting to close a quote can result in a blank page display.

- Tag names and attributes are lowercase. Attributes are always given in either of the two forms:

  `attribute_name="value"
  attribute_name=’value’`
where the value is case sensitive and can be empty. For Boolean attributes, those that are either on or off, use either of these forms for “on”:

```
attribute_name = "attribute_name"
attribute_name = ""
```

and omit the attribute for “off”.

- Unrecognized tags and attributes are ignored by browsers.
- Most elements involve start and end tags. Other elements, such as `<br/>` (line break) and `<img .../>` (image), do not have closing tags and are known as void elements. The slash (/) at the end is optional for HTML5 but needed for polyglot documents.
- Elements must be well-formed. This means no missing opening or closing tags and no improper element nesting. For example,
Learning <strong>HTML5</strong> overlaps the tags and is not properly nested. The correct nesting is

Learning <strong>HTML5</strong>

- Attributes can be required or optional and can be given in any order. If an attribute is not given, its initial (default) value, if any, is used.

- Extra white space and line breaks are allowed between the tag name and attributes and around the = sign inside an attribute. Line breaks and white space within attribute values are also allowed but should be avoided because they may be treated inconsistently by browsers.

- The body element may contain only flow (block) elements. Freestanding text (not enclosed in block elements) or inline elements are not allowed directly in the body element.
HTML5 Core Attributes

- **id**—Uniquely identifies the element in a page. All ids in a document must be distinct. Among other uses, a URL ending in `#some_id` can lead directly to an element inside a document.

- **style**—Gives presentation styles for the individual element. For example, the code
  
  ```html
  <body style="background-color: cyan">
  ```

gives the color value **cyan** to the style property **background-color** for this element. Several style properties separated by semicolons can be given. The **style** attribute is a direct but inflexible way to specify presentation style. Although this attribute is sometimes necessary, better and much more versatile methods for assigning styles can be found in Chapter 4.
• **class**—Specifies a *style class* or a space separated list of style classes for the element. For example, `class="fineprint"` or `class="footnote fineprint"`. Thus, you may place HTML elements in different classes and associate presentation styles to all elements belonging to the same class.

• **title**—Provides a title for the element. This may be used for tool-tip displays by browsers.

• **hidden**—Prevents the element from being displayed by a browser when set to `true`.

Other core attributes include `contenteditable`, `draggable`, `dropzone`, `spellcheck`, and so on.
A typical webpage is organized into the following parts inside the root element html.

The head element contains child elements: the page title (title), the page character encoding with a meta tag

\[\text{<meta charset=\"UTF-8\"/>}\]

both are required by HTML5. Usually head contains additional elements for styling, scripting, and other meta info.

The body element provides the page content, often organized into a header part for the top banner and a horizontal navigation bar at the top of the page.

After the header, the page may also have a vertical navigation bar on the left side. The flow (block) element nav is used for
navbars that organize links. Often, site-wide links are on top and page-specific links are on the left side of a page.

- Then, there are one or more articles (**article**) for the main content, followed by a **footer** at the end of the webpage.
- An article may contain one or more sections (**section**) that consist of headings (**h1** through **h6**), paragraphs (**p**), tables, figures, audio, and video.
- A paragraph may contain text, pictures (**img**), audio, and video.
- The footer often provides information on copyright, author, and links to terms of use, privacy policy, customer service, and so on.
- An **aside** flow (block) element can set aside related information, such as links to references, outside resources, and advertisements, that are not the primary content of the page.
Sections and Paragraphs

The Green Earth Project

Project Background
Put first paragraph here
Put second paragraph here

A Successful Past
Another paragraph here

Current Status of Green Earth
Another paragraph here

Future Goals
The Green Earth Project

<section>
  <h3>Project Background</h3>
  <p>Put first paragraph here</p>
  <p>Put second paragraph here</p>
  <h4>A Successful Past</h4>
  <p>Another paragraph here</p>
</section>

<section>
  <h3>Current Status of Green Earth</h3>
  <p>Another paragraph here</p>
</section>

<section>
  <h3>Future Goals</h3>
</section>

Demo: Ex: GreenEarth
• The `<p>` (a paragraph) is a flow (block) element which may contain texts and phrasing elements.

• A flow element is typically displayed with a leading and a trailing blank line.

  The element content will be formatted to fit the available width. Line breaks are inserted automatically (line wrapping) where needed to render the contents. Extraneous white spaces between words and lines within the source text of the content are normally ignored (white-space collapsing).

• If you need a line break at a specific point in the content, you can use the `<br />` tag to call for a line break. For a long-running text without spaces, such as an email or Web
address, you can insert the void element `<wbr />` to indicate a line break opportunity. The browser will do a line break indicated by `wbr` only if necessary. For example,

```html
<p>Please visit www.somelong.<wbr />andcomplicated.com.</p>
```

- Inside a flow element, you can place other phrasing (inline) elements such as `q`, `em`, `mark`, `strong`, `img`, `video`, and `audio`.

- By default, browsers usually display headings and paragraphs left-aligned and without indenting the lead line. You may use `<br/>` to call for a line break between phrasing (inline) elements.

- The `&nbsp;` is a nonbreaking space. Use it instead of a regular SPACE between two words that must be kept together on one line or use several nonbreaking spaces to add more spacing between two words.
In his famous *I Have A Dream* speech, Martin Luther King said:

I have a dream that my four little children will one day live in a nation where they will not be judged by the color of their skin but by the content of their character.
In his famous <em>I Have A Dream</em> speech, Martin Luther King said:

<blockquote cite="http://www.mlkonline.net">
I have a dream that my four little children will one day live in a nation where they will not be judged by the color of their skin but by the content of their character.
</blockquote>

Demo: Ex: Quote
<p>Confucius: <q>Don’t employ a person due to words or dismiss words due to the person.</q></p>

Demo: Ex: InlineQuote
Confucius: “Don't employ a person due to words or dismiss words due to the person.”
<hr style="height: 4px; width: 50%;
    margin-left: auto; margin-right: auto" />

Demo: Ex: Hrule
White Space and Line Wrapping

• HTML uses white space separates text into words. HTML regards the following as white-space characters:

  SPACE: ASCII 32 (entity &amp;#x0020;)
  RETURN: ASCII 13, NEWLINE: ASCII 10
  TAB: ASCII 9 (entity &amp;#x0009;)
  FORMFEED: ASCII 12 (entity &amp;#x000C;)
  Zero-width space: a non-ASCII character (entity &amp;#x200B;)

• Words can be separated by one or more white-space characters but will only result in at most one rendered interword space.

• Tags do not break words. For example,

  &lt;p&gt;The HTML&lt;strong&gt;5&lt;/strong&gt; standard.&lt;/p&gt;
Manage Displayed Line Breaks

- To force a line break, use the `<br/>` element.
- To keep two words on the same line, use the nonbreaking space (`&#xA0;` or the non-polyglot `&nbsp;`) instead of a regular space.
- To mark places where a long word can be broken across lines, you may use the non-polyglot *soft hyphen* (`&shy;`), which is rendered as a HYPHEN (–) only at the end of a line. Browsers generally do not break a word that is hyphenated in the source code.
- To indicate where long words can be broken across lines without adding a hyphen, use the `<wbr/>` element.
Preformatted Text

<figure style="width: 12em; background-color: cyan">
<pre>
  North

  West       East

  South
</pre></figure>
Demo: Ex: Pre
Phrasing Elements

- **a**: a link
- **br**: an explicit line break
- **cite**: a citation
- **em**: emphasis, usually displayed in italics
- **strong**: strong emphasis, usually displayed in boldface
- **mark**: stronger emphasis with highlighting
- **code**: computer code, usually displayed in a monospaced font
- **del**: deleted words displayed with a line through them
- **sub**: subscript (e.g., $x_{0}$)
- **sup**: superscript (e.g., $x^{2}$)
- **samp**: sample computer output

- **span**: a general phrasing element that can contain other phrasing (inline) elements, providing a simple way to attach presentation styles to enclosed elements; for example,

  `<span style="font-weight: bold; color: blue">Important point</span>`

- **var**: a variable

- **kbd**: keyboard text
<p>Fireworks start at <time datetime="2011-07-04T19:00">7pm on Independence Day</time></p>
<p>The final NASA space shuttle Atlantis launched on <time datetime="2011-07-08T11:29-04:00">the morning of Friday, 08 July 2011</time> in Cape Canaveral, Florida USA.</p>
Publication Date

- Put a `time` element as child of `body` or child of the desired article.
- Give `datetime` a date string with optional time string.
- Add the attribute `pubdate="pubdate"`

```html
<body>
<time timedate="2012-07-07" pubdate="pubdate"></time>
...
</body>
```
You control document presentation by attaching *style rules* to elements. There are three ways to attach style rules:

1. Place style rules for individual and groups of elements in separate *style sheets* and then attach them to HTML documents with `<link ... />` in the `<head>` element.

2. Include `<style>` elements in the `<head>` element.

3. Define the `style` attribute for an individual element.
   
   `<h1 style="color: darkgreen">The Green Earth Project</h1>`

All three ways of attaching style rules can be used in the same page. The `style` attribute takes precedence over styles in the `<style>` element, which takes precedence over those specified in external style sheets.
**style Attribute**

General form:

```
style="property_1:value_1; property_2:value_2; ... "
```

Foreground and Background Colors:

- `color`: `some_color`
- `background-color`: `some_color`
**Text Alignment**

- `text-align: left`—lines are left justified
- `text-align: right`—lines are right justified
- `text-align: center`—lines are centered
- `text-align: justify`—lines are justified left and right
Font Sizes

<footer style="font-size: x-small">
  <p> ... </p>
  ...
  <p> ... </p>
</footer>

Demo: **Ex: FontSize**
Indenting Content

To indent first line

```html
<p style="text-indent: 3em"> ... </p>
```

To indent entire flow (block) element

```html
margin-left: length
margin-right: length
```

```html
<div style="margin-left: 5em; margin-right: 5em">
<p> ... </p>
</div>
```
Style Length Units

- **em**—the *font-size* of the current font
- **ex**—the *x-height* of the current font
- **ch**—the size of 0 (zero) of the current font
Color values in style properties can be a color name such as *magenta* or *darkblue*. Currently there are about 150 color names defined in CSS.
Color values can also be given in a number of standard notations, including RGB (red-green-blue) and HSL (hue-saturation-lightness):

1. \#rrggbb—where the first two, middle two, and last two of the six hexadecimal digits specify red, green, and blue values, respectively (e.g., \#0ace9f). This is 24-bit color.

2. \#rgb—shorthand for the above notation when the first two, middle two, and last two digits are the same (e.g., \#03c stands for \#0033cc).
3. \texttt{rgb}(r, g, b)—where base-10 integers between 0 and 255 inclusive are used (e.g., \texttt{rgb}(0,204,108)). This is the decimal equivalent of notation 1.

4. \texttt{rgb}(r\%, g\%, b\%)—where integral percentages are used for the three color components.

5. \texttt{hsl}(h, s\%, l\%)—where \(h\) (in 0–360 degrees) indicates the hue on the color wheel.

6. \texttt{rgba}(r, g, b, a)—adding an \textit{alpha} opacity value to notation 2 where \(a\) is a decimal point value with a range of 0 (totally transparent) to 1 (totally opaque). An \texttt{rgb}(\ldots) value in notation 2 is the same as \texttt{rgba}(\ldots, 1).

7. \texttt{rgba}(r\%, g\%, b\%, a)—adding an \textit{alpha} opacity value to notation 4 in the same way.

8. \texttt{hsla}(h, s\%, l\%, a)—adding an \textit{alpha} opacity value to notation 5.
The Color Wheel
Font Styles

font-family
font-style
font-variant
font-weight
font-size

font-family: Times
font-family: Arial, Helvetica, sans-serif

Demo: Ex: FontFamily
Some Fonts

Times Arial Helvetica
Courier Monospace
Generic Font Families

- serif—for example, Times
- sans-serif—for example, Arial or Helvetica
- cursive—for example, Zapf-Chancery
- fantasy—for example, Western
- monospace—for example, Courier
Font Weight

font-weight: normal
font-weight: bold
font-weight: bolder
font-weight: lighter
Relative Font Sizes

xx-small  x-small  small  medium  large
x-large    xx-large

Absolute Font Sizes

- pt (points; 1 pt = 1/72 in.)
- pc (picas; 1 pc = 12 pt)
Modifying Font Sizes

- smaller
- larger
- xx% (a percentage of the current font size)
Itemized Lists

- Bullet list: The `<ul>` element provides an *unordered list* where the ordering of the items is unimportant. A `<ul>` is usually presented as a set of bulleted items.

- Ordered list: The `<ol>` element offers a *numbered list* where the ordering of the items is important. An `<ol>` is typically displayed as a sequence of enumerated items.

- Definition list: The `<dl>` element is handy for a *definition list* where each term (`<dt>`) is given a definition or description (`<dd>`).
<ul>
<li>Tropical Fruits
  <ol>
    <li>Pineapple</li>
    <li>Banana</li>
    <li>papaya</li>
  </ol>
</li>
<li>Cereals
  <ol>
    <li>Barley</li>
    <li>Rice</li>
    <li>Wheat</li>
  </ol>
</li>
<li>Vegetables
  <ol>
    <li>Broccoli</li>
    <li>Onion</li>
    <li>Yam</li>
  </ol>
</li>
</ul>

Demo: <strong>Ex: List</strong>
Lists

- Tropical Fruits
  1. Pineapple
  2. Banana
  3. papaya
- Cereals
  1. Barley
  2. Rice
  3. Wheat
- Vegetables
  1. Broccoli
  2. Onion
  3. Yam
A Definition List

**HTML5**
Hypertext Markup Language, a W3C Standard

**PHP**
The Hypertext Preprocessor, a popular active-page language

**MySQL**
A freely available relational database system
<table>
<thead>
<tr>
<th><strong>dt style=&quot;font-style: italic&quot;</strong></th>
<th><strong>dd</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML5</td>
<td>Hypertext Markup Language, a W3C Standard&lt;br /&gt;</td>
</tr>
<tr>
<td>PHP</td>
<td>The Hypertext Preprocessor, a popular active-page language&lt;br /&gt;</td>
</tr>
<tr>
<td>MySQL</td>
<td>A freely available relational database system</td>
</tr>
</tbody>
</table>

Demo: **Ex: Defs**
<ul style="list-style-type: circle"> ... </ul>
<ol style="list-style-type: upper-alpha"> ... </ol>

<p>The following list has inside positioning</p>
<ol style="list-style-position: inside">
  <li style="list-style-type: square; color: green">
    <span style="color: black">First item in the list with a green square marker.</span>
  </li>
  <li style="list-style-type: square; color: red">
    <span style="color: black">Second item in the list with a red square marker.</span>
  </li>
  <li style="list-style-type: square; color: blue">
    <span style="color: black">Third item in the list</span>
  </li>
</ol>
with a blue square marker.</span></li>
</ul>

list-style: circle inside

Demo: **Ex: MarkerStyle**
Links in Webpages

\(<a \text{ href}="URL" \text{anchor} </a>\)

\(<a \text{ href}="bio.html" \text{Brief Bio} </a>\)
\(<a \text{ href}="http://www.w3.org/" \text{The W3C Consortium} </a>\)
\(<a \text{ href}="../pic/dragonfly.jpg" \text{type}="image/jpeg" \text{title}="dragonfly.jpg" \text{Picture of Dragonfly} </a>\)
\(<a \text{ href}="sound/cthd.mp3" \text{type}="audio/mpeg" >
  Tan Dun, Yo Yo Ma - Crouching Tiger, Hidden Dragon - Theme
</a>\)
<h3 id="products">Our Quality Products</h3>

<a href="URL#products"> ... </a>

<article>

<nav><ul>
<li><a href="#product">Products</a></li>
<li><a href="#service">Services</a></li>
<li><a href="#testimonial">Testimonials</a></li>
</ul></nav>

<section>
<h3 id="product">Our Quality Products</h3>
 ...
</section>
</section><section>
<h3 id="service">Responsive Services</h3>
...
</article>
Site Internal and External Links

- Is clearly indicated as going off site. Often a webpage will include external links, such as references and advertisements, in an aside element displayed on the right-hand side of the page.
- Is displayed in a new browser window or tab so the visitor can come back by closing that new window or tab. A simple way is to use the attribute target="_blank" to cause the referenced page to display in a new window/tab:

  `<a href="http://www.w3.org/" target="_blank">The W3C Consortium</a>`
Site Organization and Navigation

- Organize the pages for a site into a hierarchy of files and directories (folders) stored on the hard disk of the server host. Avoid nonalphanumeric characters in file and directory names. Otherwise, the file name must be URL encoded before becoming part of a URL.

- Place the site entry page (usually, `index.html`) in the server root directory.

- Use subdirectories such as `images/`, `videos/`, `css/` (for style sheets), `js/` (for JavaScript code), `products/`, `services/`, `contractors/`, `members/`, and `affiliates/` to organize the site. The `index.html` page within each subdirectory is usually the lead page for that part of the site.
• Keep the organization simple and avoid using more than three levels of subdirectory nesting.

• Design a navigation system that is clear, easy to use, and effective in getting visitors where they want to go in your site.

• Use relative URLs exclusively for linking within the site and make sure the link is in one of these forms:
  1. Relative to the host page itself (href="file" or href="dir/file")
  2. Relative to the server root (href="/path-to-file")
In creating the content-only site, consider establishing pages with these parts:

1. *Major navigation*—Links to the main page, and first-level pages. If the top banner of a page includes a logo of the business or site, link the logo image to the site entry (main page).

2. *Minor navigation*—Links to subpages of this page and links to directly related sibling pages.

3. *In-page navigation*—Links to parts of this page when appropriate.

4. *Draft page content*—Includes text, pictures, and other media types.
Linking to Services

• Email links—A link in the form

\[
\text{<a href="mailto:email-address?SUBJECT=\text{line}\"}>}
\]

tells the browser to launch a program to send email to the given address using the indicated subject line. The subject line (from ? on) is optional. For example,

\[
\text{<a href= "mailto:pwang@cs.kent.edu?SUBJECT= Web%20Design%20and%20Programming">contact Paul</a>}
\]

Note spaces (%20) and other nonalphanumeric characters should be URL encoded. Generally, the mailto URL may have zero or more & separated header=value pairs. Useful headers include to (additional recipient address), cc, and body (message body). For example,
Joint web design and programming email listserv group provides an easy way to join a listserv.

- Download links—A link in the form
  
  `<a href="ftp:host:port/path-to-file">`

  tells the browser to launch an FTP program to connect to the given host and to download the specified file by anonymous FTP. This is useful for downloading large files such as programs and compressed (ZIP or GZIP) files. If port is not given, then the standard port 21 for FTP is assumed. For example,

  Download `<a href="ftp://monkey.cs.kent.edu/file.zip">file.zip</a>`

  (35439 bytes).

  An FTP URL can also supply username, password, and file location information for file retrieval.
• Telephone/SMS/Fax links—Links in these forms

  <a href="tel:phone_number">
  <a href="sms:phone_number">
  <a href="fax:phone_number">

  are useful for mobile phone and tablet devices.

• VOIP call links—A link in the form

  <a href="callto:screen_name or phone_number">

  asks the browser to launch Skype™ or a similar program to make voice-over-IP calls or to conduct voice/video conference.
Display Style for Links

- Visual browsers pay special attention to the presentation of links. Usually, different display styles are used to indicate whether a textual link is not visited yet, under the mouse (hover), being clicked (active), or visited already (visited).

- Browsers settings define default colors for links.

- An image anchoring a link may by default be displayed with a distinct border.

- The appearance of links can be controlled by style settings.

- Web users are accustomed to seeing links underlined. Therefore, avoid underlining regular text because it can cause confusion. Image links, on the other hand, are almost always presented without the any border. Web users understand that
clicking an image often leads to another page.

• A consistent set of link styles and colors is important for site design. Style sheets give you much control over
A Sample Navbar

SuperStore.com
Shop and Save

Groceries   Hardware   Automotive   Office Supply
<header>
  <section style="margin-left: 50px">
    <h1>SuperStore.com</h1>
    <h3>Shop and Save</h3>
  </section>
  <nav style="background-color: darkgrey; padding-left: 40px">
    <a style="color:#fff; margin:10px" href="gr/">Groceries</a>
    <a style="color:#fff; margin:10px" href="hw/">Hardware</a>
    <a style="color:#fff; margin:10px" href="au/">Automotive</a>
    <a style="color:#fff; margin:10px" href="of/">Office Supply</a>
  </nav>
</header>

Demo: Ex: Navbar
Pictures and Images in Webpages

A Clickable Image

```html
<img src="hat.jpg" alt="A nice hat"
style="width:160px; height:200px" />
```

Go to Paul's homepage

http://www....edu/~pwang
<a title="Go to Paul’s homepage"
    href="http://www.cs.kent.edu/~pwang">
    <img src="http://www.cs.kent.edu/~pwang/paul.jpg"
        alt="photo of the author Paul S. Wang"/>
</a>

Demo: **Ex: ImgLink**
<p>For this green monkey, the new Chevy Volt is just the car I have been waiting for. ...</p>
<p>I love this car. On a recent trip to ...</p>

Demo: <b>Ex: Float</b>
For this green monkey, the new Chevy Volt is just the direction I've been waiting for. It has great looks and driving range. Most importantly, it is an extended range vehicle allowing me to totally avoid gas stations for my daily commute. I don't have to worry about running out of juice ever again. I love this car. On a recent trip to ...
For this green monkey, the new Chevy Volt is just the car waiting for. It has great looks and drive, most importantly, it is an extended range electric vehicle, totally avoid gas stations for my daily commutes. I do not have to worry about running out of juice.

I love this car. On a recent trip to …

Demo: **Ex: FloatClear**
Image Alignment within A Line

Here is some text and an image `<img src="URL" style="vertical-align: baseline" />`

- `vertical-align: baseline`—Aligns baselines of image and text.
- `vertical-align: middle`—Aligns middle of image with middle of x character in preceding text.
- `vertical-align: text-top`—Aligns top of image with font top of preceding text.
- `vertical-align: text-bottom`—Aligns bottom of image with font bottom of preceding text.
- `vertical-align: xx%`—Raises the bottom of image xx percent of the text `line height`.

Demo: **Ex: ImgAlign**
Inline Alignments with Preceding Text

\[\text{vertical-align: middle} \quad \text{OR} \quad \text{vertical-align: baseline}\]

\[\text{vertical-align: text-bottom} \quad \text{OR} \quad \text{vertical-align: text-top}\]

\[\text{vertical-align: 20\%} \quad \text{OR} \quad \text{vertical-align: -25\%}\]
• **vertical-align: top**—Aligns top of image to tallest element on the line, which could be another image or some other tall element in the same line.

• **vertical-align: bottom**—Aligns bottom of image to lowest element on the line, which could be another image or some other element in the same line.
Whole-Line Alignments

vertical-align: top in a line with another image.

vertical-align: middle in a line with another image.

vertical-align: bottom in a line with another image.
<h3 style="color: blue">Alignments with respect to the whole line:</h3>
<p><span style="color:blue">vertical-align: top</span><img alt="yinyang" src="01.png" style="vertical-align: top"/> in a line with another <img alt="yinyang" src="01.png" width="20" style="vertical-align: middle"/> image.</p>

<p><span style="color:blue">vertical-align: middle</span> <img alt="yinyang" src="01.png" width="20" style="vertical-align: middle" /><span style="color:blue"> vertical-align: bottom</span><img alt="yinyang" src="01.png" style="vertical-align: bottom"/></p>
Fig. 7: Dragonfly, an insect belonging to the order Odonata, the suborder Epiprocta or, in the strict sense, the infraorder Anisoptera. (Wikipedia)
<figure style="text-align: center; font-style: italic">
<img src="dragonfly.jpg" alt="a blue-winged dragonfly" />
<figcaption>
<strong>Fig. 7:</strong> Dragonfly, an insect belonging to the order Odonata, the suborder Epiprocta or, in the strict sense, the infraorder Anisoptera. (Wikipedia)
</figcaption>
</figure>

Demo: Ex: FigCap
Image Encoding Formats

- Graphics Interchange Format (GIF)—A raster format suitable for icons, logos, cartoons, and line drawings. GIF images can have up to 256 colors (8-bit).

- Joint Photographic Experts Group (JPEG) format—A raster format usable for color and black-and-white pictures with continuously changing color tones for display. JPEG images can store up to 16.8M colors (24-bit). Images created using a scanner or digital camera are usually stored in TIFF (Tagged Image File Format), JPEG, or GIF.

- Portable Network Graphics (PNG) format—A format designed to replace GIF. PNG really has three main advantages over GIF: alpha channels (variable transparency), gamma correction (cross-platform control of image brightness); (a method of
progressive display). Browser support for PNG is increasing steadily, and ideally, PNG will soon replace GIF.
Aliasing in Raster Images
Colors in Raster Images

- Monochrome—black and white
- Gray scale—different levels of gray (up to 256 shades)
- Indexed—Each pixel color is indicated by an index into a color palette. The palette may contain a set of up to 256 colors. The smaller the palette, the fewer bits needed for each index.
- High color—thousands of colors, 15 to 16 bits per pixel
- True color—16.8 million colors, 24 bits per pixel
Image Maps

```html
<map name="samplemap">
  <area shape="rect" coords="0,0,100,150"
        href="some-url" alt="item 1" />
  <area shape="poly" coords="0,0,10,32,98,200"
        href="some-url" alt="item 2" />
  <area shape="circle" coords="0,0,100"
        href="some-url" alt="item 3" />
  <area shape="default"
        href="some-url" alt="item otherwise" />
</map>

<img src="img-url" usemap="#map-name" />
```
Image Map Example

<figure>
  <img src="planets.jpg" usemap="#planets"
       alt="Planets image map" />
  <map name="planets" id="planets">
    <area shape="circle"
          coords="40,176,7"
          href="mercury.html" alt="Mercury" title="Mercury"/>
    <area shape="circle"
          coords="82,158,10"
          href="venus.html" alt="Venus" title="Venus"/>
    <area shape="circle"
          coords="127,132,11"
          href="earth.html" alt="Earth" title="Earth"/>
  </map>
</figure>
<area shape="circle"
    coords="157,103,10"
    href="mars.html" alt="Mars" title="Mars"/>
<area shape="circle"
    coords="234,116,27"
    href="jupiter.html" alt="Jupiter" title="Jupiter"/>
<area shape="poly"
    coords="254,53,327,54,373,102,300,107"
    href="saturn.html" alt="Saturn" title="Saturn"/>
<area shape="default"
    href="solar.html"
    alt="List of solar system planets" />
</map>
</figure>

Demo: Ex: Planets
The Planets

Click on a planet to see its description.

Click elsewhere in the picture to get a list of all the planets in the solar system.
# A Sample Page Layout

## Top Banner

<table>
<thead>
<tr>
<th>Links</th>
<th>to</th>
<th>Top-level</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navbar</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpage Link</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpage Link</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpage Link</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpage Link</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Content title

- Contents ... here
- Contents ... here
- Contents ... here

### Sidebar

- External Link
- Advertisement
- Resource Link

---

End-of-page Footer
Webpage Layout

<header>
<h1 style="text-align: center">Top Banner</h1>
<nav style="background-color: black; color: white; padding-left: 40px">Links to Top-level Pages</nav></header>

<div style="background-color: darkgrey">
<!-- three columns here -->
<section style="clear: both"></section>
</div>

<!-- page footer here -->
<nav style="float: left; padding: 1em; color: white">
<p>Navbar</p>
</nav>
<a href="#">Subpage Link</a><br /><br />
</ul>
</nav>

<section style="float: left; padding: 10px;
     width:50%; background-color: white">
<h2>Content title</h2>
<article class="sectionArticle">
<p>Contents ... here</p>
</article>
</section>

<aside style="float: left; color: white;
     padding: 1em">
<p>Sidebar</p>
<ul><li><p>External Link</p></li>
<li><p>Advertisement</p></li>
</ul>
<li><p>Resource Link</p></li>
</aside>
<footer style="border: thin solid black;
    text-align: center">End-of-page Footer</footer>

Demo: **Ex: Layout**
Debugging and Validation

• A spell checker can help you find typos and spelling errors. Careful proofreading can catch grammar and other writing errors.

• Test your webpage with different browsers under different operating systems. Even when a page looks OK, it may still contain coding errors. That is because Web browsers will ignore elements and attributes not recognized, as well as other problems in your HTML code.

• A page displaying correctly is no evidence that your code is error free. The W3C markup validator (validator.w3.org) is useful for debugging your HTML code.

• The W3C CSS validator (jigsaw.w3.org/css-validator) is
handy to check your style code.

- You also need to make sure all links in your webpage are correct and not broken. For this, you can use the W3C link checker ([validator.w3.org/checklink](http://validator.w3.org/checklink)).