The Basics of HTML

Expected Learning Outcomes:

- At the end of this chapter you need to understand the following:
  - Web Page Components and Browser Declarations
  - The Building Blocks For Web Page Markup
  - Basic Text Formatting
  - Working With Other Components
    - Lists
    - Forms
    - Tables
    - Links
    - Special Entity Characters
    - Video & Audio
  - Verification and Checking

Both the theory and the application are important!

Browser Declarations

- The HTML5 <!DOCTYPE> declaration represents the document type, and helps browsers to display web pages correctly.

- It must appear only once, at the top of the page before any HTML tags).

<!DOCTYPE html>

- A far simpler declaration than previous versions of HTML, e.g.,
  - <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>

Markup: Elements, Attributes and Values

- **Elements** are labels that identify the structure and different parts of a web page. Each element consists of an opening tag, the content and closing tag. An element can be empty of content.

```
Definition of <em>Interactive</em>
```

Here is a typical element. The opening and closing tags surround the text that will be affected. In this case, the word "Interactive" will be emphasised, which in most browsers means it will be set in italics.

```
<img src="picturename.jpg" width="300" alt="This is an image" />
```

Self-closing elements, like img shown here, do not surround any text content. They have a single tag which serves both to open and close the document.

Markup: Elements, Attributes and Values

- **Attributes** contain information about the element. Attributes are always located inside an element's opening tag.

```
<td colspan="3">February</td>
```

Here is an element (for a table cell) with a simple attribute-value pair.

- **Values** for element attributes should always be enclosed in quotation marks.

```
<img src="picturename.jpg" width="300" height="170" alt="" />
```

Some elements, like img shown here, can take one or more attributes, each with its own value. The order is not important. Separate each attribute value-pair from the next with a space.

HTML Text Markup

Poets have tried to describe **Ankh-Morpork**. They have failed. Perhaps it's the sheer zestful vitality of the place, or maybe it's just that a city with a million inhabitants and no sewers is rather robust for poets, who prefer daffodils and no wonder.

-- (*Terry Pratchett, Mort*)

Poets have tried to describe Ankh-Morpork. They have failed. Perhaps it's the sheer zestful vitality of the place, or maybe it's just that a city with a million inhabitants and no sewers is rather robust for poets, who prefer daffodils and no wonder.

--(Terry Pratchett, Mort)

HTML Text Markup

- Paragraphs
  - ,

```
Mary had a little lamb, <p> its fleece was white as snow </p>
```

Mary had a little lamb,

its fleece was white as snow

```
Mary had a little lamb, <br /> its fleece was white as snow
```

Mary had a little lamb,
its fleece was white as snow

HTML Text Markup

- Headings

- Blockquotes

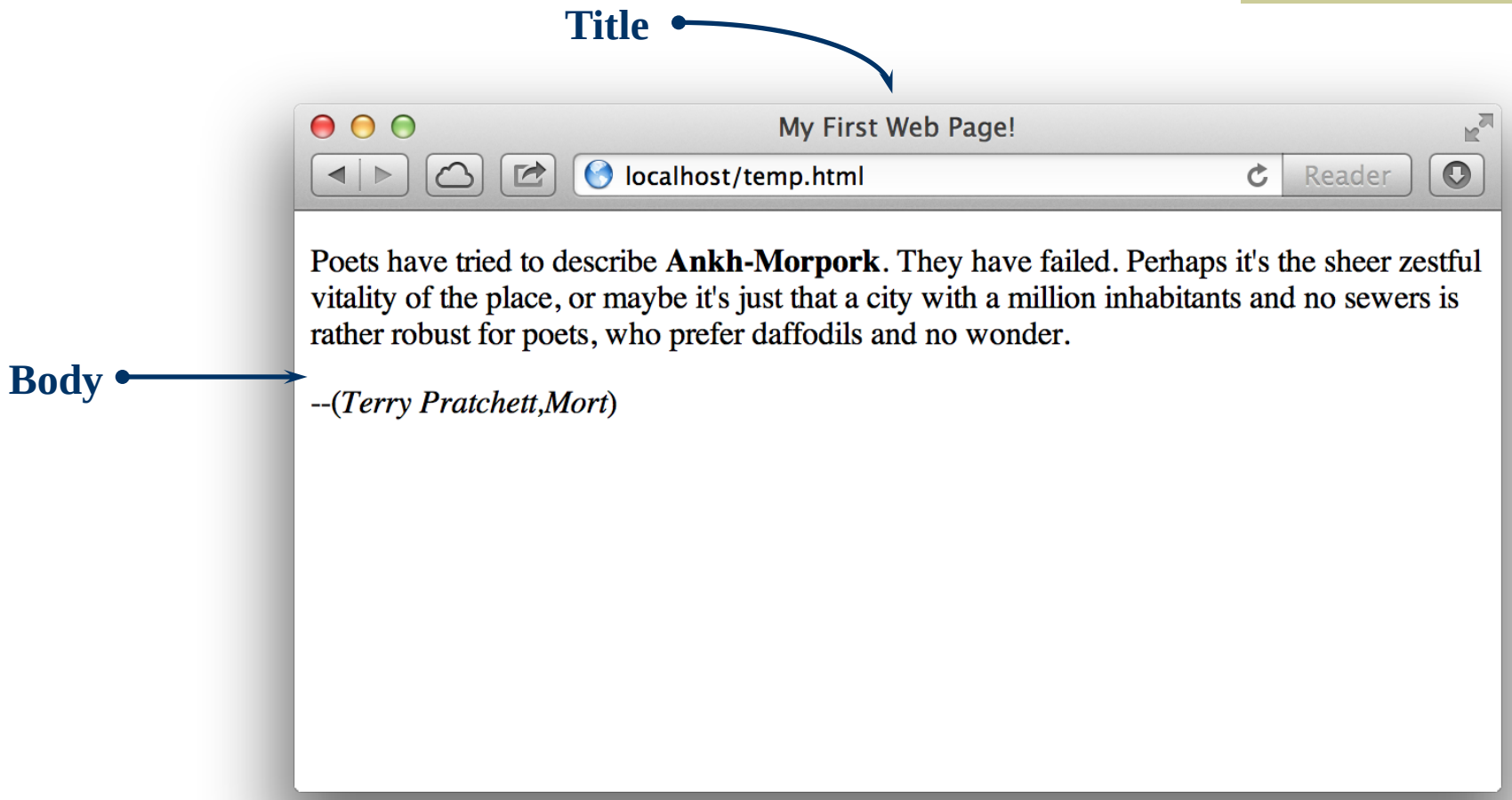
```
<p>Here is a famous quote: <blockquote> "The
only way not to succeed is not to try."
</blockquote> It was written by <cite>Edward
Teller</cite>.</p>
```

```
Here is a famous quote:
      "The only way not to succeed is
      not to try."
It was written by Edward Teller.
```

My First Web Page!

```
<!DOCTYPE html>
<html>
    <head>
      <title> My First Web Page! </title>
    </head>
    <body>
        <p>
        Poets have tried to describe <b>Ankh-Morpork</b>. They
          have failed. Perhaps it's the sheer zestful vitality of
          the place, or maybe it's just that a city with a
          million inhabitants and no sewers is rather robust for
          poets, who prefer daffodils and no wonder.
        </p>
        --(<i>Terry Pratchett,Mort</i>)
    </body>
</html>
```

My First Web Page!



Standard HTML Template

`<!DOCTYPE html>`	→	This lets the web browser know what version of HTML the page is written in.
`<html>`	→	This indicates the start of the web page.
`<head>`	→	This starts the head of the document.
`<title>`	→	The document title which appears at the top of a browser window.
`</title>`	→	The end of the document title.
`</head>`	→	The end of the header section.
`<body>`	→	The start of the main body of the document (content).
`</body>`	→	The end of the document body,
`</html>`	→	The end of the web page.

HTML5 Structural Elements

- HTML5 recognises that web pages have a structure – with each webpage in a website generally following the same structure.

Wireframe components:

<section> - to define sections of page

<header> - defines the head**er** of a page

<footer> - defines the footer of a page

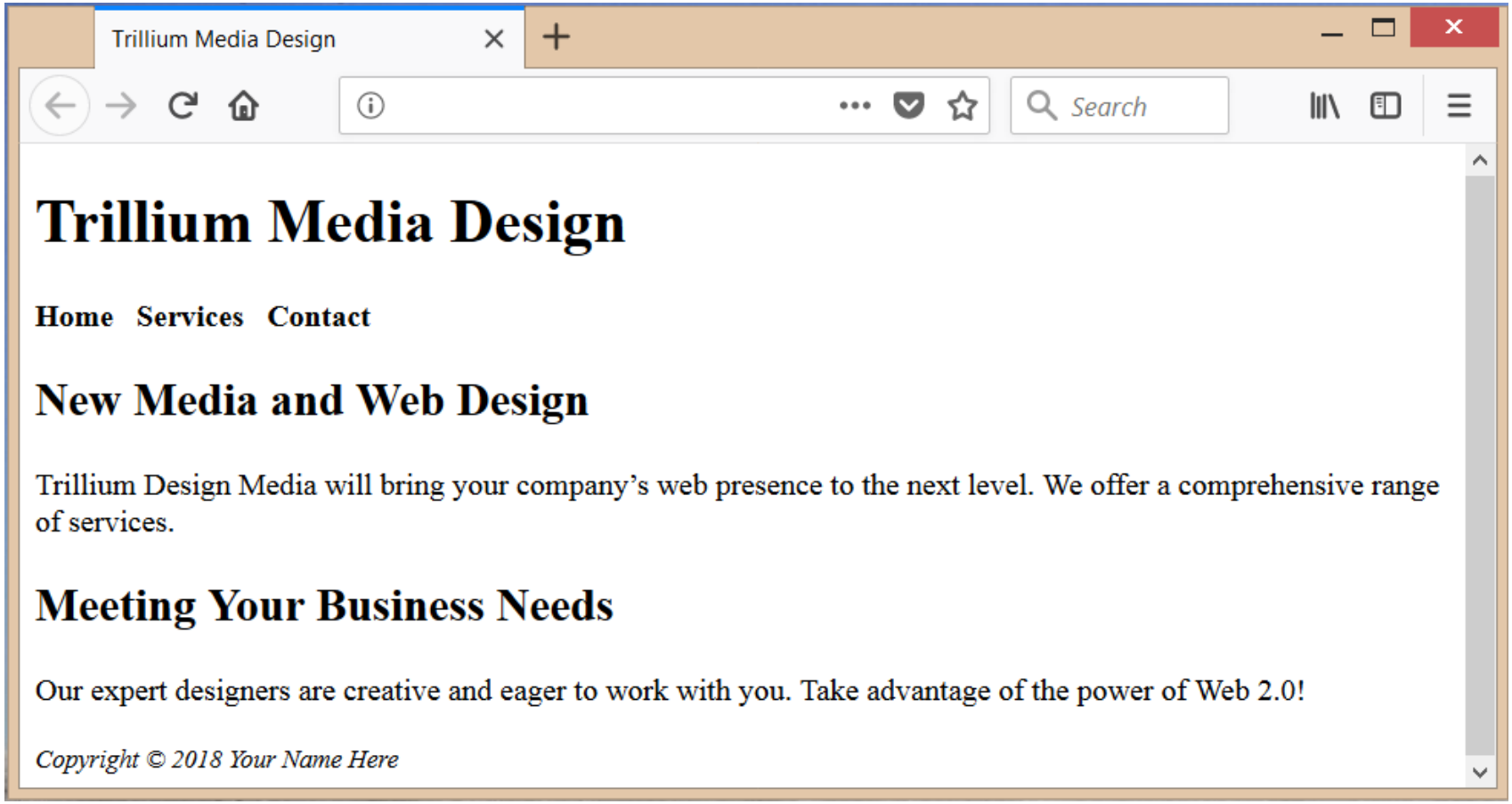
<nav> - defines the navigation on a page

<article> - defines the article or primary content on a page

<aside> - defines extra content like a sidebar on a page

<figure> - defines images that annotate an article

HTML5 Structural Elements



HTML5 Structural Elements

```
<!DOCTYPE html>
<html>
        <head>
                <meta charset="UTF-8" />
                <title>Trillium Media Design </title>
        </head>
        <body>
                <header><h1> Trillium Media Design </h1>
                        <nav><b>Home   Services    Contact   </b></nav>
                </header>
                <h2>New Media and Web Design</h2>
                <p>Trillium Design Media will bring your company's web presence to the next level. We
                 offer a comprehensive range of services.</p>
                 <h2>Meeting Your Business Needs</h2>
                 <p>Our expert designers are creative and eager to work with you.  Take advantage of the
power of Web 2.0!</p>
                <footer>
                        <small><i>Copyright &copy; 2018 Your Name Here</i></small>
                </footer>
        </body>
</html>
```

Useful Tips

- Nested Tags
  - Improperly nested tags confuse the browser and the results can be unpredictable. Under the XHTML/HTML standard, improperly nested tags are illegal.
    - E.g., `<b><i> … </b></i>`
- Comments
  - Comments in programs increase the readability of programs. Comments in HTML have the same purpose.
  - Browsers ignore comments.
    - E.g., `<!-- whatever you want to say -->`
    - E.g., `<!- myHome.html`
      `This document describes my homepage`
      `and is… -->`

Useful Tips

- The lang Attribute
  - The language of the web page can be declared in the <html>.
  - The language is declared with the lang attribute.
  - Declaring a language can be important for accessibility applications (screen readers) and search engines.
    - E.g.,

```
<!DOCTYPE html>
<html lang="en-US">
<body>
…
</body>
</html>
```

The first two letters specify the language. Common examples are “en” for English, “en” for Spanish, “fr” for France, and so on.

If there is a dialect, use two more letters.

Working With Images

- Although images can be digitally stored in many different formats, web pages tend to use three common forms:
  - Graphic Interchange Format (GIF)
  - Joint Photographics Experts Group (JPEG)
  - Portable Network Graphic (PNG)

Working With Images

- The <img /> Tag
  - Specifies an image that is about to appear in a document.
    - `<img src= "terry.gif" />`

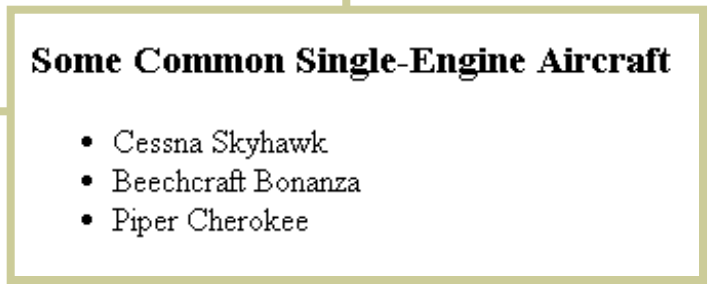
- Alternate Text – use the **`alt`** attribute
  - Specifies an alternate text for an image.
    - The text should describe the image if the image contains information
    - Use **`alt`**`=""` if the image is only for decoration
    - HTML screen readers can read the **alt** attribute
    - … as can bots and spiders.

    - `<img src="../images/submitButton.gif" alt= "Submit button." />`

Types of Lists

- Unordered Lists

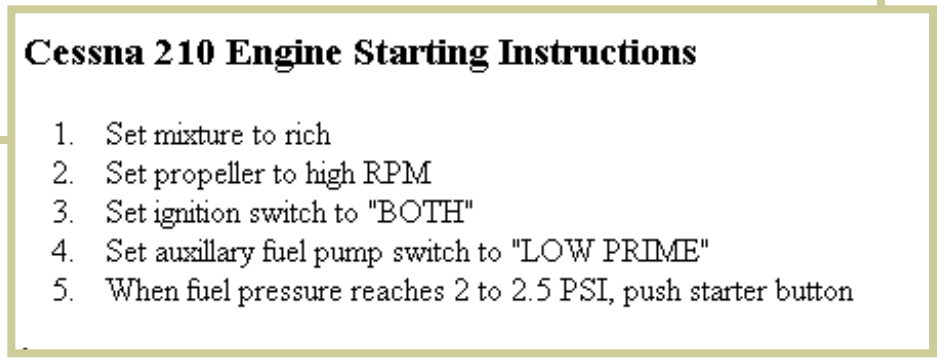
```
<!-- unordered.html -->
<h3>Some Common Single-Engine Aircraft</h3>
<ul>
  <li>Cessna Skyhawk</li>
  <li>Beechcraft Bonanza</li>
  <li>Piper Cherokee</li>
</ul>
```



Types of Lists

- Ordered Lists

```
<!-- ordered.html -->
<h3> Cessna 210 Starting Instuctions </h3>
<ol>
  <li> Set mixture to rich </li>
  <li> Set propeller to high RPM </li>
  <li> Set ignition switch to "BOTH" </li>
  <li> Set auxillary fuel pump switch to "LOW PRIME"/li>
  <li> When fuel pressure reaches 2 to 2.5 PSI, push the
       starter button </li>
</ol>
```



Types of Lists

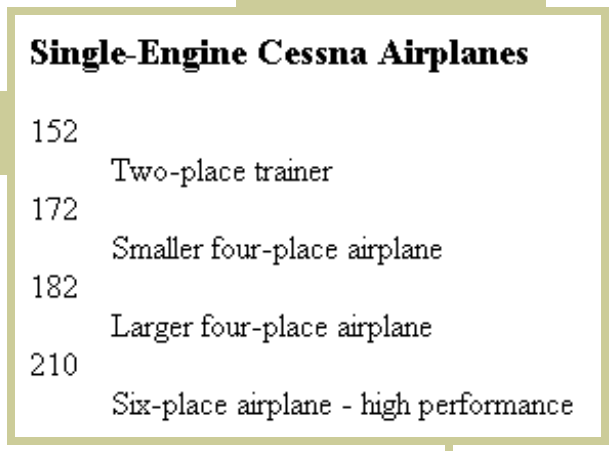
- Ordered Lists
  - A **type** attribute can be added to an **ordered list**, to define the type of the marker:

Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

Types of Lists

- Definition Lists

```
<!-- definition.html -->
<h3>Single-Engine Cessna Airplanes</h3>
<dl>
        <dt> 152 </dt>
        <dd> Two-place trainer</dd>
        <dt> 172</dt>
        <dd> Smaller four-place airplane</dd>
        <dt> 182</dt>
        <dd> Larger four-place airplane</dd>
        <dt> 210</dt>
        <dd> Six-place airplane - high performance</dd>
</dl>
```



Tables

- Tables provide a highly readable way of presenting information.
- A table is a matrix of rows and columns, each position in which is called a cell.

Number	First Name	Last Name	Points
1	Eve	Jackson	94
2	John	Doe	80
3	Adam	Johnson	67
4	Jill	Smith	50

- The information in a cell can be almost any document element, including text, headings, images, and nested tables.
- Details in relation to padding, text alignment and border-spacing will be discussed when we cover the chapter on CSS.

Key Table Tags

- <table> … </table>
  - Begin/End Table
- <tr>…</tr>
  - Begin/End Table Row.
- <th>…</th>
  - Begin/End Table Cell Header.
- <td>…</td>
  - Begin/End Table Cell
  - Cell data is provided between these tags

An Example Table

`<table>`	The start of the table.
`<tr>`	The beginning of the first row.
`  <td>One</td>`	The first cell of this row with the text 'One'.
`  <td>Two</td>`	
`</tr>`	End of the first row.
`<tr>`	Row 2 begins.
`  <td>Three</td>`	
`  <td>Four</td>`	
`</tr>`	End of row 2
`</table>`	End of the table.

An Example Table

```
<!- table.html -->
<table>
<caption> Fruit Juice Drinks </caption>
  <tr>
      <th></th>
      <th scope="col"> Apple </th>
      <th scope="col"> Orange </th>
      <th scope="col"> Screwdriver </th>
  </tr>
  <tr>
      <th scope="row"> Breakfast </th>
      <td> 0 </td>
      <td> 1 </td>
      <td> 0 </td>
  </tr>
  <tr>
      <th scope="row"> Lunch </th>
      <td> 1 </td>
      <td> 0 </td>
      <td> 0 </td>
   </tr>
   <tr>
      <th scope="row"> Dinner </th>
      <td> 0 </td>
      <td> 0 </td>
      <td> 1 </td>
   </tr>
</table>
```

Fruit Juice Drinks

	Apple	Orange	Screwdriver
Breakfast	0	1	0
Lunch	1	0	0
Dinner	0	0	1

Forms

- HTML forms are used to collect user input.
- The **<form>** element defines an HTML form.

```
<form> …form elements … </form>
```

- HTML forms contain **form elements**.
  - Form elements are different types of input elements, checkboxes, radio buttons, submit buttons, and more.
  - The **<input>** element has many variations, depending on the type attribute.

Type	Description
text	Defines normal text input
radio	Defines radio button input (for selecting one of many choices)
submit	Defines a submit button (for submitting the form)

Forms

- Text Input:
  - **<input type="text">** defines a one-line input field for **text input**:

```
<form>
First name:<br>
<input type="text" name="firstname">
<br>
Last name:<br>
<input type="text" name="lastname">
</form>
```

This is how it will look like in a browser:

First name:

Last name:

Note: The form itself is not visible. Also note that the default width of a text field is 20 characters.

Forms

- Radio Button Input:
  - **<input type="radio">** defines a **radio button**:

```
<form>
<input type="radio" name="sex" value="male" checked>Male
<br>
<input type="radio" name="sex" value="female">Female
</form>
```

This is how the HTML code above will be displayed in a browser:

(•) Male
( ) Female

- Submit Button Input:
  - **<input type="submit">** defines a button for submitting a form to a form-handler.
  - The form-handler is typically a server page with a script for processing input data.
  - The form-handler is specified in the form's **action** attribute:

```
<form action="action_page.php">
First name:<br>
<input type="text" name="firstname" value="Mickey">
<br>
Last name:<br>
<input type="text" name="lastname" value="Mouse">
<br><br>
<input type="submit" value="Submit">
</form>
```

This is how the HTML code above will be displayed in a browser:

First name:
Mickey
Last name:
Mouse

Submit

Forms

- The **`action`** attribute defines the action to be performed when the form is submitted.
- The **`method`** attribute specifies the HTTP method (**`GET`** or **`POST`**) to be used when submitting forms.

```
<form action="action_page.php" method="GET">
```

or

```
<form action="action_page.php" method="POST">
```

Forms

- When to use **`GET`** (default method):
  - If the form submission is passive (like a search engine query) and without sensitive information.
  - Best suited to small amounts of data.
  - When you use **`GET`** the form data will be visible in the page address:

```
action_page.php?firstname=Mickey&lastname=Mouse
```

- When to use **`POST`**:
  - If the form is updating data, or includes sensitive information (e.g., password).
  - **`POST`** offers better security because the submitted data is not visible in the page address.

Forms

- The **`Name`** Attribute:
  - To be submitted correctly, each input field must have a name attribute.
  - This example will only submit the “Last name” input field:

```
<form action="action_page.php">
First name:<br>
<input type="text" value="Mickey">
<br>
Last name:<br>
<input type="text" name="lastname" value="Mouse">
<br><br>
<input type="submit" value="Submit">
</form>
```

Forms

- An HTML <form> element, with all possible attributes set, will look like:

```
<form action="action_page.php" method="GET" target="_blank" accept-
charset="UTF-8"
enctype="application/x-www-form-urlencoded" autocomplete="off" novalidate>
.
form elements
.
</form>
```

More on forms and form handling will be covered in lab sessions.

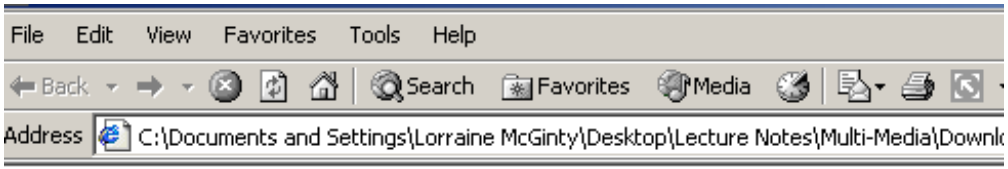
Here is the list of <form> attributes:

Attribute	Description
accept-charset	Specifies the charset used in the submitted form (default: the page charset).
action	Specifies an address (url) where to submit the form (default: the submitting page).
autocomplete	Specifies if the browser should autocomplete the form (default: on).
enctype	Specifies the encoding of the submitted data (default: is url-encoded).
method	Specifies the HTTP method used when submitting the form (default: GET).
name	Specifies a name used to identify the form (for DOM usage: document.forms.name).
novalidate	Specifies that the browser should not validate the form.
target	Specifies the target of the address in the action attribute (default: _self).

Links

- Links are a distinguishing feature of the World Wide Web, and typically they have three parts:
  - A destination
    - What will happen when a user clicks the link.
      - e.g., send an e-mail, show an image, play a sound, connect to other web pages, and sometimes to specific location on other web pages called "anchors".
  - A label
    - The part the user sees and clicks on to reach the destination.
      - i.e., Text, an image or both.
  - A target
    - Determines where the destination will be displayed. Often ignored or left up to the browser.
      - e.g., a named window, a frame, or a new window.

Links



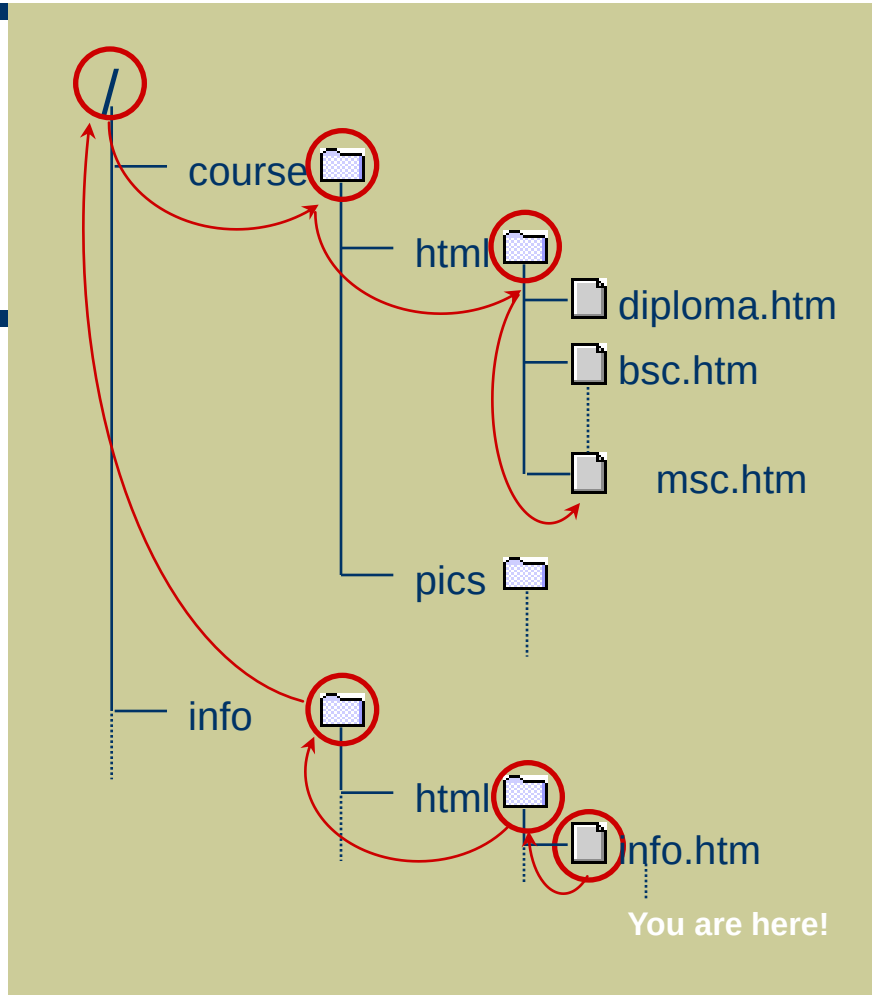
Starsearch Enterprises

New programs
Press releases
Upcoming events
About Starsearch Enterprises

- Key Points:
  - The difference between *Absolute* and *Relative* links
  - The Syntax of Hyperlinks
  - Inter- *vs* Intra-Document links

Absolute vs Relative URLs

- An absolute URL is analogous to a complete postal address!
  - It shows the entire path to the file, including scheme, server name, the complete path, and the filename itself.
- A relative URL is analogous to giving directions to a neighbour's house!
  - It describes the location of the desired file relative to the file that contains the URL itself.




```
ABSOLUTE URL: http://www.cs.ucd.ie/course/html/msc.htm
RELATIVE URL: ../../course/html/msc.htm
```

Locates the root directory from info.htm. ("../" = "parent of")

Absolute *vs.* Relative Links

- Relative links can only reference local documents.
- Relative links offer greater flexibility and portability…
  - We could transfer Web site files to a new machine without changing the links.
  - Useful during site development and testing.

Hypertext Link Syntax

- All links are specified in an attribute of an `<a>` tag.
  - A document that includes an `<a>` tag that specifies a link is called the **source** of that link.
  - The document whose URL is specified in a link is called the **target** or **anchor** of that link.
  - When the target is in the same document as the link to it, the document is both the source and target of that link.

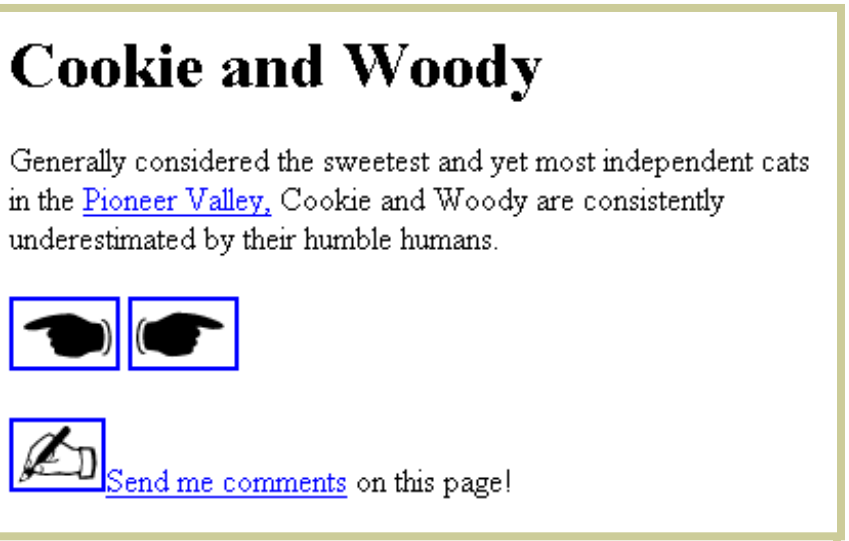
- `href` stands for *hypertext reference*.
  - Generic example: *link text*

Links

- As a general rule, use relative URLs for links to Web pages on your site and absolute URLs for links to Web pages on other sites.
- It is a good idea to use all lowercase letters for your URLs to avoid problems on the many servers that are case sensitive.
- Try not to use "Click here" for a label. Instead use the key words that already exist in your text to identify the link. Alternatively an image could act as a label.
- To create a link to a particular place on a page (*inter-document links*), use an anchor.
- To make the link appear in a given window or frame (*intra-document links*), use a target.

Using Images to Label Links

- We use the `alt` attribute to specify alternate text that should appear, if for some reason the image does not.




```
<h1>Cookie and Woody</h1>
<p>
   Generally considered the sweetest and yet most independent cats in the
   <a href = "pioneerval.html">Pioneer Valley,</a> Cookie and Woody are
   consistently underestimated by their humble humans.
</p>
<p>
   <a href = "prevpage.html">
        <img src="pointleft.gif" alt="Previous page" />
   </a>
   <a href = "nextpage.html">
        <img src="pointright.gif" alt="Next page" />
   </a>
</p>
<p>
   <a href = "mailto:lcastro@crocker.com"> <img src="writeletter.gif" alt =
   "Send mail" /> Send me comments</a> on this page!
</p>
```

Inter/Intra Document Links

- Generally, a click on a link brings the user to the *top* of the target web page. If you want to jump to a specific section of that page you need to create an ***anchor*** and then reference that anchor in the link.
- Creating an Anchor:

```
<a id="anchorName"> Add content here… words,images etc… </a>
or, alternatively you could use
<h2 id="anchorName"> This is the heading I want to link to. </h2>
```

- Linking to a Specific Anchor:

```
<a href="#anchorName"> This is what the user will click on. </a>
```

- Note
  - If the anchor is in a separate document use `<a href="page.html#anchorName">` to the reference section. There should be no space between the URL and the #.
  - An absolute link using an anchor may look like this:

```
<a href="http://www.site.com/path/page.html#anchorName">
```

Other Kinds of Links

1. Type <a href="

1. Type the URL:

   - For a link to any file on the Web, including movies, sounds, programs, Excel spreadsheets or whatever, type:

     ```
     http://www.site.com/path/file.ext
     ```

   - For a link to an FTP site, type:

     ```
     ftp://ftp.site.com/path
     ```

   - For a link to an e-mail address, type:

     ```
     mailto:name@site.com
     ```

2. Type ">.

Other Special Entity Characters

- In order to use special characters such as quotation marks in your web page document, you need to use “entity characters”. Examples include:

Character	Entity Name	Code
“	Quotation mark	&quot;
©	Copyright symbol	&copy;
&	Ampersand	&amp;
Empty space	Nonbreaking Space	
’	Right single quote	&rsquo;
<	Less than	&lt;

iframes

- An iframe is used to display a web page within a web page.

```
<iframe src="URL"></iframe>
```

- The **src** attribute specifies the URL (web address) of the iframe page.

```
<iframe src="demo_iframe.html" width="200" height="200"></iframe>
```

- The attribute values are specified in pixels by default

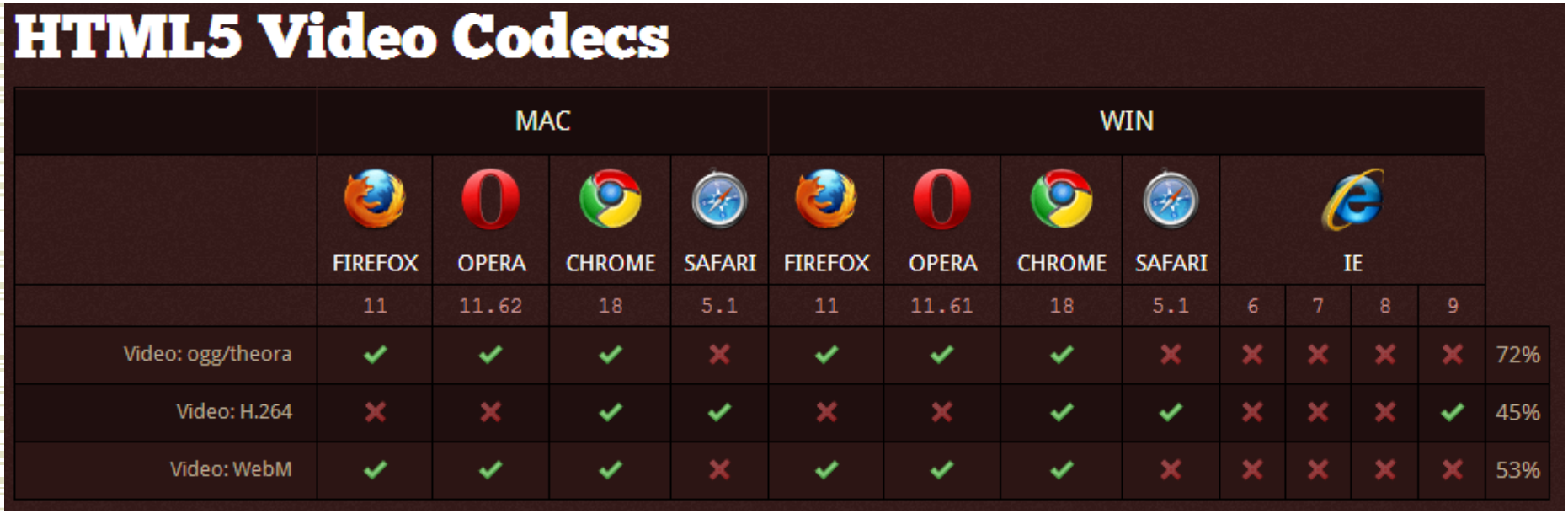
<video> tag

```
<!DOCTYPE html>
<head>
<title>A title</title>
</head>
<html>
<body>

<video width="320" height="240" controls="controls">
  <source src="movie.mp4" type="video/mp4" />
  <source src="movie.ogg" type="video/ogg" />
  <source src="movie.webm" type="video/webm"/>
Your browser does not support the video tag.
</video>

</body>
</html>
```

Any text within the <video> tags will be displayed in any browsers that do not support the tag

Video Formats & Browser Support

Currently, there are 3 supported video formats for the <video> element: MP4, WebM, and Ogg



<audio> tag

```
<!DOCTYPE html>
<html>
        <head>
                <title> My song </title>
        </head>
        <body>

        <audio controls="controls">
          <source src="song.ogg" type="audio/ogg" />
          <source src="song.mp3" type="audio/mpeg" />
        Your browser does not support the audio element.
        </audio>
   </body>
</html>
```

Any text within the <audio> tags will be displayed in any browsers that do not support the tag

Audio Formats & Browser Support

Currently, there are 4 supported file formats for the <audio> element: MP3, Wav, ACC and Ogg:

HTML Syntax Validation

- The W3C has a free Markup Validation Service available at: http://validator.w3.org .