Tutorial 2: Developing a Basic Web Site

College of Computing & Information Technology
King Abdulaziz University

CPCS-403 – Internet Applications Programming
Objectives

- Learn how to storyboard various Web site structures
- Create links among documents in a Web site
- Understand relative and absolute folder paths
- Work with the base element
- Mark a location with the id attribute
- Create a link to an id
- Mark an image as a link
Objectives

- Create an image map from an inline image
- Client side image maps
- Remove an image border
- Server side image maps
- Link “advice”
Working with Web Site Structures

- A **storyboard** is a diagram of a Web site’s structure, showing all the pages in the site and indicating how they are linked together.

- It is important to **storyboard** your Web site before you start creating your pages in order to determine which structure works best for the type of information the site contains.

- A well-designed structure can ensure that users will be able to navigate the site without getting lost or missing important information.
Linear Structures

- In a **linear structure**, each page is linked with the pages that follow and precede it in an ordered chain.
- **Linear structure** works best for Web pages with a clearly defined order.
- In an **augmented linear structure**, each page contains an additional link back to an opening page.
Linear Structures

A linear structure

Each page is linked with the preceding and following pages.

Act I
Scene 1

Act I
Scene 2

Act I
Scene 3

Act II
Scene 1

Act II
Scene 2

Act II
Scene 3

...

An augmented linear structure

Each page is linked with the preceding and following pages with an additional link to the first scene page.

Act I
Scene 1

Act I
Scene 2

Act I
Scene 3

Act II
Scene 1

Act II
Scene 2

Act II
Scene 3

...
Hierarchical Structures

- In the **hierarchical structure**, the pages are linked going from the home page down to more specific pages.
- Users can easily move from general to specific and back again.
- Within this structure, a user can move quickly to a specific scene within the page, bypassing the need to move through each scene in the play.
Hierarchical Structures

Pages are arranged in a hierarchy from the general down to the specific; users can move up and down the tree.

- Home page
- Acts
  - Act I
  - Act II
  - Act III
- Scenes
  - Scene 1
  - Scene 2
  - Scene 3
  - Scene 4
  - Scene 5
Mixed Structures

- As Web sites become larger and more complex, you often need to use a combination of several different structures.
- The overall form can be hierarchical, allowing the user to move from general to specific; however, the links also allow users to move through the site in a linear fashion.
- A site index is a page containing an outline of the entire site and its contents.
Mixed Structures

Pages are linked in a variety of ways.
Web Site with No Coherent Structure
Protected Structures

Sections of most commercial Web sites are off-limits except to subscribers and registered customers.
CAMshots Website

- Amateur photography website
  - Gerry wants to give advice and info for those new to digital photography
  - Wants a site with several pages
    - Each dedicated to a particular topic
  - Thus far, he has a few basic pages, but hasn’t even linked them together
CAMshots Website

- Amateur photography website
- Here are the three current pages:

<table>
<thead>
<tr>
<th>Pages in the CAMshots Web site</th>
</tr>
</thead>
<tbody>
<tr>
<td>home.htm</td>
</tr>
<tr>
<td>the CAMshots home page</td>
</tr>
</tbody>
</table>
Creating a Hypertext Link

- Amateur photography website
  - Each page should link to the other pages
  - The links should be as follows:
Creating a Hypertext Link

Amateur photography website

Creating links:

- To “link” pages, you must create hypertext links
- Hypertext links are created by enclosing some document content within a set of opening and closing <a> tags
- The general syntax is:

  `<a href="reference">content</a>`

  where `reference` is the location being linked to and `content` is the document content that is being marked as a link
- The reference value can be a page on the web, a local file, an email address, or a network server.
Creating a Hypertext Link

Creating links:

- Filenames are case sensitive on some operating systems, including the UNIX and Macintosh, but not on others.
- The current standard is to use lowercase filenames for all files on a Website and to avoid special characters such as blanks and slashes.
- You should also keep filenames short to avoid typing errors.
Creating a Hypertext Link

- Amateur photography website
  - Each page should link to the other pages
  - As an example, to create a hypertext link to the tips.htm file, you could enter the following code:
    - `<a href="tips.htm">Photography Tips</a>`
    - The text “Photography Tips” is now understood, by the browser, as a link
      - This text will now be underlined, serving as a clue to the user that the text links to another document or web page
    - And if the user clicks on the words “Photography Tips”, the browser will load the linked document (which is tips.htm)
Creating a Hypertext Link

```html
<body>
  <div>
    <img src="camshots.jpg" alt="CAMshots" />
  </div>
  <hr />

  <div>
    [ <a href="home.htm">Home</a> ]
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    [ <a href="tips.htm">Tips</a> ]
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    [ <a href="glossary.htm">Glossary</a> ]
  </div>
</body>
```

---

![Web page example](image_url)

---

**Welcome to CAMshots, a site for people passionate about digital photography. This site has grown out of decades of photographic experience. I offer advice for both beginners and advanced users. I hope you enjoy what you find, but please be considerate of the work it took to do all this. The entire site contents Including all images and articles are copyrighted. Please honor my work and do not copy anything without permission. If you are interested in publishing any of my images or articles or using them in other ways, please contact me and we can discuss your needs. Happy Shooting!**

---

*— Gerry*

**hypertext links**
Creating a Hypertext Link

- Amateur photography website

  - Notice:
    - In the links we just created, we only specified the filename.
    - We did not specify a location of the file we were linking to, such as a specific folder.
    - We **assumed** that the linked files were within the SAME folder as the page we were linking from.
    - When you specify ONLY a filename, the browser will search for the linked file in the **same folder** as the document containing the hypertext link.
    - But consider a site with HUNDREDS of pages.
      - Are you really going to store all of them in one folder?
Managing Files on Large Sites

- Typical Web sites
  - Most websites will have hundreds of documents
    - Many large websites have tens of thousands, if not hundreds of thousands, of documents
    - Clearly, these documents need to be filed away in a reasonable fashion, thereby making them easier to manage
  - File cabinet example:
    - File cabinets have multiple drawers for different files
    - Each drawer is often divided into sections
    - Each section has folders
    - And the folders store the documents.
  - The same thought-process is used for file storage
Managing Files on Large Sites

- CAMshots Website
  - Gerry wants the site files organized into folders
  - He wants the following folders:
    - pages: will store all html pages for the website
    - images: will store all graphics for the website
    - video: will store all video files for the website
  - Since he plans on having many web pages, he further wants the pages folder broken down:
    - tips folder: html pages that give photography tips
    - glossary folder: stores all glossary pages
    - articles folder: stores all miscellaneous articles
Managing Files on Large Sites

Diagram of file structure:
- camshots
  - pages
    - tips
      - tips1.htm
      - tips2.htm
    - glossary
      - glossary.htm
  - articles
    - images
    - video
  - index.htm
Specifying a Folder Path

Creating links to files within specific folders:

To create a link to a file located in a different folder than the current document, you must specify the file’s location, or path.

HTML supports two kinds of paths:

1) Absolute
2) Relative
Specifying a Folder Path

- **Absolute Path:**
  - An **absolute path** specifies a file’s precise location within a computer’s entire folder structure.
  - The syntax is as follows:
    
    \[
    \text{folder1/folder2/folder3/file}
    \]
    
    - where folder1 is the topmost folder in the computer’s folder tree, followed by folder2, folder 3, and so forth, down to the file you want to link to.
  - If files are located on different drives, as well as in different folders, you must include the drive letter:
    
    \[
    \text{/drive1/folder1/folder2/folder3/file}
    \]
Specifying a Folder Path

Absolute Path:

Using absolute paths for CAMshots

Here’s how you would express absolute paths for the four files that Gerry currently has:

<table>
<thead>
<tr>
<th>Absolute Path</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>/camshots/pages/tips/tips1.htm</td>
<td>The tips1.htm file located in the pages/tips subfolder</td>
</tr>
<tr>
<td>/camshots/pages/tips/tips2.htm</td>
<td>The tips2.htm file located in the pages/tips subfolder</td>
</tr>
<tr>
<td>/camshots/pages/glossary/glossary.htm</td>
<td>The glossary.htm file located in the pages/glossary subfolder</td>
</tr>
<tr>
<td>/camshots/index.htm</td>
<td>The index.htm file located in the camshots folder</td>
</tr>
</tbody>
</table>
Specifying a Folder Path

- Relative Paths
  - When many files and folders are involved, absolute paths can be a pain!
    - They can be both cumbersome and confusing
    - Most web designers prefer to use relative paths
  - A **relative path** specifies a file’s location in relation to the location of the current document
  - If the file is in the same location as the current document, the relative path is simply the filename
    - you do not have to specify the folder name
Specifying a Folder Path

Relative Paths

A **relative path** specifies a file’s location in relation to the location of the current document.

- If the file is in a subfolder of the current document, you have to include the name of the subfolder in the path.
  
  `folder1/folder2/file`

- To go farther down the folder tree to other subfolders, include each subfolder in the relative path, separated by slashes.
  
  `folder1/folder2/folder3/file`
Specifying a Folder Path

Relative Paths

- If you want to go one level up the folder tree, you start the relative path with a double period (..), a forward slash, and then provide the name of the file
  
  ```
  ../file
  ```

- To specify a different folder on the same level, known as a sibling folder
  
  - you move up the folder tree using the double period (..) and then down the tree using the name of the sibling folder
    
    ```
    ../folder/file
    ```
Specifying a Folder Path

Relative Paths

- The following table shows the relative paths to Gerry’s files, starting from the camshots/pages/tips subfolder

<table>
<thead>
<tr>
<th>Relative Path from the /camshots/pages/tips Subfolder</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>tips1.htm</td>
<td>The tips1.htm file located in the current folder</td>
</tr>
<tr>
<td>tips2.htm</td>
<td>The tips2.htm file located in the current folder</td>
</tr>
<tr>
<td>../glossary/glossary.htm</td>
<td>The glossary.htm file located in the sibling glossary folder</td>
</tr>
<tr>
<td>../../index.htm</td>
<td>The index.htm file located in the parent camshots folder</td>
</tr>
</tbody>
</table>

Final note:
- You should almost always use relative paths
Changing the Base

Resolving Relative Pathnames

- A browser resolves (translates/understands) relative pathnames based on the location of the current document.
  - If a link points to folder/file, the browser assumes that folder represents a subfolder within the folder that the active document is located in.
  - So the browsers “starting point” for understanding a relative path is based off of the location of the current document.
Changing the Base

Resolving Relative Pathnames

Changing the starting point (changing the base)

If you want the browser to have a different starting point when resolving relative paths, you use the following:

```html
<base href="path" />
```

- where path is the folder location that you want the browser to use when resolving relative paths in the current document

The **base element** is useful when a document is moved to a new folder.

- Rather than rewriting all of the relative paths to reflect the document’s new location, the base element can redirect browsers to the document’s old location, allowing any relative paths to be resolved
Using the id Attribute

- **id Attribute**
  - To **jump** to a specific location **within a document**, you first need to **mark that location**
  - One way to identify elements in an HTML document is to use the **id attribute**
    - Example: mark an h2 element with the id value of “H”:
      `<h2 id="H">H</h2>`
  - **Id names** must be unique
  - **Id names** are not case sensitive
Using the id Attribute

**id Attribute**

- Gerry’s glossary page is long and wants links that will simply jump down to an alphabetical section of the glossary (based on letter)

![Jumping to a location within a Web page](image)

- clicking the letter D from the alphabetical list ...
- ...jumps the user to the D section of the glossary
Using the id Attribute

- **id Attribute**
  - Here’s how you mark those sections with the id attribute:

```html
<h2 id="A">A</h2>
<dl>
  <dt><b>Ambient Light</b></dt>
  <dd>The natural light in a scene.</dd>
  <dt><b>Aperture</b></dt>
  <dd>The maximum size of the hole through which light enters the camera.</dd>
  <dt><b>Artifact</b></dt>
  <dd>Unwanted distortions in an image caused by image compression.</dd>
  <dt><b>Aspect Ratio</b></dt>
  <dd>The ratio between the width and height of an image.</dd>
</dl>

<h2 id="B">B</h2>
<dl>
  <dt><b>Bit</b></dt>
  <dd>The smallest unit of computer memory.</dd>
  <dt><b>Bitmap</b></dt>
  <dd>A method of storing information that maps an image pixel bit by bit.</dd>
  <dt><b>Byte</b></dt>
  <dd>A group of 8 bits, the basic unit of information for the computer.</dd>
</dl>
```
Linking to Locations within Documents

- Linking to a marked element:
  - Once an element is marked, you can now link to it.
  - Syntax:
    ```html
    <a href="#id">content</a>
    ```
    - where id is the value of the id attribute of the element being linked to.
  - Example: create a link to the h2 heading for the letter R in the glossary document
    - First we mark the h2 letter “R” as follows:
      ```html
      <h2 id="R">R</h2>
      ```
    - Then we link to this section as follows:
      ```html
      <a href="#R">R</a>
      ```
Linking to Locations within Documents

- Linking to a marked element:
  - Now we can click an alphabetical character and go to that particular section of the glossary:

![Glossary page](image-url)

- [Home] [Tips] [Glossary]

Glossary

Creating Links between Documents

- You know how to:
  - Link from one document to another
  - Link from one document to a location within that same document

- Now, what about linking to a specific location in another document:

  `<a href="reference#id">content</a>`

  where reference is a reference to an HTML or XHTML file and id is the id of an element marked within that file

  thereby letting you link directly to that marked section
Creating Links between Documents

Here we link to different sections of the glossary.htm page:

```
<ul>
  <li>camera:
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Nikon D50</li>
  <li><a href="glossary.htm#f-stop">F-stop</a>:
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;f/7.1</li>
  <li><a href="glossary.htm#exposure">Exposure</a>:
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;1/200 sec.</li>
  <li><a href="glossary.htm#focal_length">Focal Length</a>:
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;18mm</li>
  <li><a href="glossary.htm#aperture">Aperture</a>:
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;3.6</li>
  <li><a href="glossary.htm#flash_mode">Flash Mode</a>:
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;No Flash</li>
</ul>
```
Brief Interlude: Human Stupidity
Working with Linked Images and Image Maps

Linked Images:

- Thus far, we’ve looked at making text links
- But you can also make an image into a link
- To mark an inline image as a hypertext link, you enclose the `<img>` tag within a set of `<a>` tags

Syntax:

```
<a href="reference"><img src="file" alt="text" /></a>
```

- Once the image has been linked, clicking anywhere within the image jumps the user to the linked file
Working with Linked Images and Image Maps

Linked Images:

- Making an image as a link is straightforward
  - But the entire image links to the destination file
  - What if you only wanted part of the image to be a link?
- HTML also allows you to divide an image into different zones, or hotspots, each linked to a different destination
- So one inline image can link to multiple destinations
Working with Linked Images and Image Maps

Linked Images:

- Gerry wants his “header” image/logo to do just this
- If the user clicks anywhere on the CAMshots circular logo, the user should jump to the home page
- If the user clicks “Tips”, they should go to the Tips page
- And finally, if the user clicks “Photo Glossary”, they should go to the Glossary page
Working with Linked Images and Image Maps
Working with Linked Images and Image Maps

- Image Map:
  - To define these hotspots, you create an image map that matches a specified region of the inline image to a specific destination.

- HTML supports two kinds of image maps:
  1) Client-side image maps
  2) Server side image maps
Client-Side Image Maps

Client-side image map

- Handled entirely by the Web browser running on the user’s computer
- Defined with the map element (tag):

```html
<map id="map" name="map">
  hotspots
</map>
```

- Where map is the name of the image map and hotspots are the locations of the hotspots within the image
Client-Side Image Maps

Client-side image map
- Defined with the map element (tag):
  `<map id="map" name="map">`
  hotspots
  `</map>`
- You must include **both** the id and name attributes
  - and they **must** have the same name/value
  - because HTML requires the name attribute and XHTML requires the id attribute
Client-Side Image Maps

- **Client-side image map**
  - Map elements can be placed anywhere within the body of the Web page
    - because they are not actually displayed by the browser
    - They are simply references for mapping hotspots of inline images
  - Common practice is to place the image map element below the inline image
  - The browser’s status bar displays the target of each hotspot
  - Older browsers do not support client-side images
Defining Hotspots

- Define a hotspot using the area element
- Syntax of the area element:

  `<area shape="shape" coords="coordinates" href="url" alt="text" />`

- **shape** is the shape of the hotspot region
  - rectangular, circular, polygon
  - default shape refers to remaining area not covered by hotspots
- **coordinates** are the list of points that define the boundaries for the hotspot region
- **url** is the file or location that the hotspot links to
- and **text** is the alternate text for nongraphical browsers
Defining Hotspots

- More info on Hotspots:
  - No limit to number of elements you can add to an image map
  - Hotspots can overlap
    - If user clicks an overlapping area, the browser opens the link of the first hotspot defined in the image map
  - Hotspot coordinates are measured in pixels
    - pixels are the smallest unit or dot in a digital image
    - Example: your monitor may display 1680 x 1050 pixels
      - Means your display is 1680 pixels wide and 1050 pixels tall
    - The pixel values, given with the coords attribute, define exactly the location and size of the hotspot region
Defining Hotspots

More info on Hotspots:

- How do you determine the coordinate values?
- You need to open up your image into a graphics program such as Adobe Photoshop or GIMP.
- From there, you can move your mouse over parts of the image and determine the necessary coordinate points.
Creating a Rectangular Hotspot

- Two points define a **rectangular hotspot**:
  - the upper-left corner
  - the lower-right corner

- **Syntax for a rectangular hotspot is:**
  ```html
  <area shape="rect" coords="x1, y1, x2, y2" ... />
  ```
  where x1, y1 are the coordinates of the upper-left corner of the rectangle, and x2, y2 are the coordinates of the lower-right corner.

- Coordinates are always expressed relative to the image’s top-left corner
  - So a coordinate of (50, 85) refers to a point that is 50 pixels to the right and 85 pixels down from the images top-left corner
Creating a Rectangular Hotspot

- CAMshots header/logo

```
<area shape="rect" coords="240, 110, 402, 145" href="glossary.htm" />
```

- So now the “Photo Glossary” portion of the image, when clicked, will link to the glossary.htm file
Creating a Circular Hotspot

- **Circular hotspot** is defined by the location of its center and its radius.
- Syntax for a **circular hotspot** is:
  ```html
  <area shape="circle" coords="x, y, r" ... />
  ```
  where *x* and *y* are the coordinates of the center of the circle and *r* is the circle’s radius.

- The following shows an example using the CAMshots logo...
Creating a Circular Hotspot

- CAMshots header/logo

- `<area shape="circle" coords="82, 78, 80" href="home.htm" />

- So now the actual logo portion of the image, when clicked, will link to the home page (home.htm)
Creating a Polygonal Hotspot

- To create a polygonal hotspot, you enter the coordinates for each vertex in the shape.

Syntax for a **polygonal hotspot** is:

```html
<area shape="poly" coords="x1, y1, x2, y2, x3, y3, x4, y4, ..." />
```

- where (x1, y1), (x2, y2), (x3, y3) and so forth, define the coordinates of each corner in the multisided shape.

- Polygonal hotspots allow you to create a wide variety of hotspot shapes/regions.
  - As long as you know the coordinates of each corner.
Creating a Polygonal Hotspot

- CAMshots header/logo

Assume we wanted a small triangular portion of the logo to link to a specific area on the website.

- We could use the poly shape:

```html
<area shape="poly" coords="30, 142, 76, 80, 110, 142" href="somepage.htm" />
```
Creating a Default Hotspot

- The final shape option is the “default” hotspot
- If you specify the default hotspot, then any portion of the image, that is NOT covered by a hotspot, when clicked, will link to the page specified in the “default” hotspot

**Syntax:**

```
<area shape="default" coords="0, 0, x, y" ... />
```

- where x is the width of the inline image in pixels and y is the image’s height

- Any spot that is not covered by another hotspot will activate the default hotspot link
Creating a Client-Side Image Map

- To create a client-side image map, insert the map element
  ```html
  <map name="map" id="map">
    hotspots
  </map>
  ```
  anywhere within the Web page body, where map is the name and id of the image map and hotspots is a list of hotspot areas defined within the image map.

- To add a hotspot to the image map, place the element
  ```html
  <area shape="shape" coords="coordinates" href="reference"
       alt="text" />
  ```
  within the map element, where shape is the shape of the hotspot region, coordinates are the list of points that define the boundaries of the region, reference is the file or location that the hotspot is linked to, and text is alternate text displayed for nongraphical browsers.

- To define a rectangular-shaped hotspot, use the area element
  ```html
  <area shape="rect" coords="x1, y1, x2, y2" ... />
  ```
  where x1, y1 are the coordinates of the upper-left corner of the rectangle and x2, y2 are the coordinates of the rectangle’s lower-right corner.

- To define a circular hotspot, use
  ```html
  <area shape="circle" coords="x, y, r" ... />
  ```
  where x and y are the coordinates of the center of the circle and r is the circle’s radius.

- To define a polygonal hotspot, use
  ```html
  <area shape="poly" coords="x1, y1, x2, y2, x3, y3, ..." ... />
  ```
  where (x1, y1), (x2, y2), (x3, y3), and so forth define the coordinates of each corner in the multisided shape.

- To define the default hotspot, use
  ```html
  <area shape="default" coords="0, 0, x, y" ... />
  ```
  where x is the width of the inline image in pixels and y is the height in pixels.

- To apply an image map to an inline image, add the usemap attribute
  ```html
  <img src="file" alt="text" usemap="#map" />
  ```
  to the img element, where map is the name or id of the map element.
Applying an Image Map

How to apply an image map to an inline image?

- Add the “usemap” attribute to the image
- Syntax:
  `<img src="file" alt="text" usemap="#map" />`
  - where map is the id or name of the map element
- So now we apply the image map to the CAMshots logo:
Remove the Image Border

- Image all of a sudden has an ugly border!
  - When you make an image into a link, a border is added around the image
    - 99,999 out of 100,000 times, you don’t want this
  - How to remove the border?
    - add a border-width style to the inline image

```html
<body>
  <div>
    <img src="camshots.jpg" alt="CAMshots" usemap="#logomap" style="border-width: 0" />
    <map id="logomap" name="logomap">
      <area shape="circle" coords="82, 78, 80" href="home.htm" alt="Home" />
      <area shape="rect" coords="168, 110, 225, 145" href="tips.htm" alt="Tips" />
      <area shape="rect" coords="240, 110, 402, 145" href="glossary.htm" alt="Glossary" />
    </map>
  </div>
  <hr />
</body>
```
Server-Side Image Maps

- In a **server-side image map**, the image map is stored on the Web server.
  - This was the original standard type of image map.
  - When you click on a hotspot, coordinates of mouse are sent to server, which then activates the corresponding link.

- Problems:
  - Can be SLOW to operate.
  - Can’t test your web page without server access.

- The message: use Client-side image maps!
Writing Effective Hypertext Links

You want a site that is easy to navigate!

Follow some basic design tips:

- Write the TEXT of your links so that the link tells readers what the document points to.
- For example, this underlined link:
  
  Click here for more information.

- doesn't tell the user what type of document will appear when “here” is clicked.
  - So try to be descriptive with the textual content of the link.
- Also, if the link points to a non-HTML file, such as a PDF doc., an image, a video, etc, include that info in the link.
  - Ex: Download the video clip here (16 MB).
Developing a Basic Web Site

WASN’T THAT AMAZING!
Daily Demotivator

ADVERSITY

That which does not kill me postpones the inevitable.
Tutorial 2: Developing a Basic Web Site

College of Computing & Information Technology
King Abdulaziz University

CPCS-403 – Internet Applications Programming
Objectives

- Understand URLs
- Link to a site on the Web
- Link to an FTP site
- Link to an e-mail address
- Work with hypertext attributes
- Work with metadata
Linking to Other Sites on the World Wide Web

- Thus far we’ve seen how to link pages to each other
  - So long as those pages are part of the same website
- And we’ve seen how to link images to pages as well
- We clearly need to link to other resources
  - Other web pages on the World Wide Web

- But first, we need to understand a few details…
Introducing URLs

- To create a link to a resource on the Internet, you need to know its **URL**

A **Uniform Resource Locator (URL)**:

- specifies the precise location of a resource on the Internet

Examples:

- [www.whitehouse.gov](http://www.whitehouse.gov)
- [www.ucf.edu](http://www.ucf.edu)
- [www.w3.org](http://www.w3.org)
Introducing URLs

- **Uniform Resource Locator (URL):**
  - All URLs share the common form:
    - `scheme : location`
    - where `scheme` indicates the type of resource referenced by the URL and `location` is the location of that resource
  - For web pages, the location refers to the actual location of the HTML file
  - The name of the scheme is taken from the protocol used to access the resource
    - A `protocol` is a set of rules defining how information is exchanged between two resources
Introducing URLs

- Internet Protocols:
  - Your Web browser communicates with Web servers using the **Hypertext Transfer Protocol (HTTP)**
  - Therefore, all **URLs** for Web pages must start with the scheme “http”
  - This tells the browser to use http when it tries to access the web page

- Other Internet resources use different **protocols** and have different scheme names
Here is a list of other internet protocols:

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Used To</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>access documents stored locally on a user’s computer</td>
</tr>
<tr>
<td>ftp</td>
<td>access documents stored on an FTP server</td>
</tr>
<tr>
<td>gopher</td>
<td>access documents stored on a gopher server</td>
</tr>
<tr>
<td>http</td>
<td>access Web pages stored on the World Wide Web</td>
</tr>
<tr>
<td>https</td>
<td>access Web pages over a secure encrypted connection</td>
</tr>
<tr>
<td>mailto</td>
<td>open a user’s e-mail client and address a new message</td>
</tr>
<tr>
<td>news</td>
<td>connect to a Usenet newsgroup</td>
</tr>
<tr>
<td>telnet</td>
<td>open a telnet connection to a specific server</td>
</tr>
<tr>
<td>wais</td>
<td>connect to a Wide Area Information Server database</td>
</tr>
</tbody>
</table>
Linking to a Web Site

- URL for a Web page has the general form:
  - `http://server/path/filename#id`
  - where `server` is the name of the Web server, `path` is the path to the file on that server, `filename` is the name of the file, and if needed, `id` is the name of an id or anchor within the file.

- Here’s a sample URL with all parts identified:
  
  `http://www.camshots.com/articles/glossary.htm#aperture`

  - protocol
  - server
  - path
  - file
  - id
Linking to a Web Site

More details:

- If a **URL** includes no path, then it indicates the topmost folder in the server’s directory tree.
- If a **URL** does not specify a filename, the server searches for the default home page:
  - Many servers use index.html (or index.htm) as the filename for the default home page.
- The server name portion of the URL is also called the **domain name**.
Understanding Domain Names

- The server name portion of the URL is also called the **domain name**
  - Each domain name contains a hierarchy of names separated by periods (.), with the top level appearing at the end
    - http://www.cs.ucf.edu
  - The top level, called an extension, indicates the general audience of the website
    - For example: .edu is reserved for educational institutions, .gov is for agencies of the U.S. government, and .com is for commercial sites and general-use sites.
Understanding Domain Names

**Domain name**

http://www.cs.ucf.edu

- The next level down (ucf, in the above URL) displays the name of the individual or organization hosting the site.
  - A domain like http://www.camshots.com indicates a commercial or general-use site owned by CAMshots

**Avoiding duplicate website names:**

- The two top-most levels of the domain must be registered with the IANA (Internet Assigned Numbers Authority)
  - ...which is what you will do later in the semester
  - This prevents us from having duplicate websites
Understanding Domain Names

- **Domain name**
  - `http://www.cs.ucf.edu`
  
- The lowest levels of the domain (farthest to the left in the domain name), are assigned by the individual or company hosting the site
  - Large web sites will often divide their domain names into several levels
  - Example for Microsoft:
    - `http://downloads.microsoft.com`
    - `http://service.microsoft.com`
    - `http://broken.microsoft.com`
    - (just kidding)
Understanding Domain Names

- **Domain name**

  http://www.cs.ucf.edu

- Finally, the lowest level of the domain (the first part of the domain name), displays the hard drive or resource that stores the Web site files.

- Many companies have standardized on using “www” as the name for the lowest level.
Linking to a Web Site

Making a link:

- To create a link, you must mark some text as a hypertext link.
- And you use the URL of the web site as the value of the href attribute.
- So if you wanted to link to UCF, you could type:

  `<a href="http://www.ucf.edu">I go to school here</a>`

Now we add some links to CAMshots…
# Linking to a Web Site

## Photography Sites on the Web

The web is an excellent resource for articles on photography and digital cameras. Here are a few of my favorites.

<table>
<thead>
<tr>
<th>Link</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apogee Photo</td>
<td>An established online photography magazine with articles by top pros, discussion forums, workshops, and more.</td>
</tr>
<tr>
<td>Outdoor Photographer</td>
<td>The premier magazine for outdoor photography. The site includes extensive tips on photographing wildlife, action sports, scenic vistas, and travel sites.</td>
</tr>
<tr>
<td>PCPhoto</td>
<td>An excellent site for novices and professionals with informative reviews and buying guides for the latest equipment and software.</td>
</tr>
<tr>
<td>Popular Photography and Imaging</td>
<td>A useful and informative site with articles from the long-established magazine of professional and amateur photographers.</td>
</tr>
</tbody>
</table>
Linking to Other Resources

- Web pages are only one type of resource available that you can link to
- There are many other resources that you can direct users to
  - FTP Servers
  - Local Files
  - Email addresses
  - and more
Linking to FTP Servers

- **FTP servers** are another method of storing and sharing files on the Internet.
  - Think of an FTP server as a file cabinet in which users can store and retrieve data files.
    - Just like you store and retrieve data files on your computer.
- FTP servers transfer information using a communications protocol called **File Transfer Protocol**, or **FTP** for short.
Linking to FTP Servers

- Accessing FTP servers
  - Syntax:
    - ftp://server/path/
    - where server is the name of the FTP server and path is the folder path on the server that contains the files you want to access
  - Once you access the server, you can navigate through the folders just as you would on your own computer
Linking to FTP Servers

To view this FTP site in Windows Explorer, click Page, and then click Open FTP Site in Windows Explorer.

Welcome to FTP.MICROSOFT.COM. Also visit http://www.microsoft.com/downloads.
Linking to FTP Servers

- Accessing FTP servers
  - FTP servers require each user to enter a username and password to gain access
    - The standard username is anonymous and requires no password
    - Your browser tries this automatically
    - But some FTP servers do not allow anonymous access
    - For these servers, your browser will prompt you for the username and password
Linking to FTP Servers

- Accessing FTP servers
  - Browsers work just fine for accessing files on FTP servers
  - However, they are a bit cumbersome
  - Much better to use an FTP client
    - A program specifically designed to access, modify, and place files on a FTP server
    - Examples:
      - Filezilla: http://filezilla-project.org/download.php
      - Core FTP: http://www.coreftp.com/download.html
  - You will be using an FTP client to upload your Web site files, to your website, later in the semester
Linking to a Local File

- On occasion, you may choose to link to files on your own computer or network (local files)
- Syntax for URL of a local file:
  \[
  \text{file://server/path/filename}
  \]
  - Where server is the name of the local network server, path is the path on that server to the file, and filename is the name of the file
  - If you are accessing a file from your own computer, the server name might be omitted and replaced by an extra slash (/)
  - So a file from the documents/articles folder, on your own computer, might have the URL as follows:
  \[
  \text{file:///documents/articles/tips.htm}
  \]
Linking to a Local File

- On occasion, you may choose to link to files on your own computer or network (local files)
  - If the file is on a different disk drive within your computer, the hard drive letter would be included in the URL as follows:
    
    \[\text{file:///D:/documents/articles/tips.htm}\]

- Other Notes:
  - The file scheme here does not imply any particular communication protocol
  - instead the browser retrieves the document using whatever method is the local standard for the type of file specified in the URL
Linking to an E-Mail Address

Linking to E-mails:

Many Web sites use e-mail to allow users to communicate with a site’s owner, sales representative, or technical support staff.

You can turn an e-mail address into a hypertext link, so that when a user clicks on an address, the browser starts an e-mail program and automatically inserts the address into the “To” field of the new outgoing message.

Syntax: mailto:address

where address is the e-mail address to be emailed.
Linking to an E-Mail Address

- Linking to E-mails:
  - Example:
    - mailto:jcazalas@cs.ucf.edu
    - Upon clicking this link, the email client would open and the email address would be inserted into the “To” field

- Adding other Information:
  - The mailto protocol allows you to add other information to the email, including the subject line and actual body text of the email message
Linking to an E-Mail Address

- Linking to E-mails:
  - Adding other Information:
    - The mailto protocol allows you to add other information to the email, including the subject line and actual body text of the email message.
  - Syntax:
    ```
    mailto:address?header1=value1&header2=value2&...
    ```
    - where header1, header2, etc are different e-mail headers and value1, value2, and so on are the values of those headers.
Linking to an E-Mail Address

Linking to E-mails:

Make a link to create this email message:

TO: jcazalas@cs.ucf.edu
SUBJECT: test
BODY: CGS 3175 is the best course ever

You would use the following URL:

mailto:jcazalas@cs.ucf.edu?Subject=test?Body=CGS %203175%20is%20the%20best%20course%20ever

notice the **spaces** in the message body have been replaced with `%20` escape characters

- This is necessary since URLs cannot have blank spaces
- The browser interprets the escape character `%20` as a blank space and resolves the text accordingly
## Linking to an E-Mail Address

<table>
<thead>
<tr>
<th>Escape Character Code</th>
<th>Character</th>
<th>Escape Character Code</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>%20</td>
<td>space</td>
<td>%5B</td>
<td>[</td>
</tr>
<tr>
<td>%0D%0A</td>
<td>new line</td>
<td>%5D</td>
<td>]</td>
</tr>
<tr>
<td>%3C</td>
<td>&lt;</td>
<td>%60</td>
<td>`</td>
</tr>
<tr>
<td>%3E</td>
<td>&gt;</td>
<td>%3B</td>
<td>;</td>
</tr>
<tr>
<td>%23</td>
<td>#</td>
<td>%2F</td>
<td>/</td>
</tr>
<tr>
<td>%25</td>
<td>%</td>
<td>%3F</td>
<td>?</td>
</tr>
<tr>
<td>%7B</td>
<td>{</td>
<td>%3A</td>
<td>:</td>
</tr>
<tr>
<td>%7D</td>
<td>}</td>
<td>%40</td>
<td>@</td>
</tr>
<tr>
<td>%7C</td>
<td>l</td>
<td>%3D</td>
<td>=</td>
</tr>
<tr>
<td>%5C</td>
<td>\</td>
<td>%26</td>
<td>&amp;</td>
</tr>
<tr>
<td>%5E</td>
<td>^</td>
<td>%24</td>
<td>$</td>
</tr>
<tr>
<td>%7E</td>
<td>~</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Linking to an E-Mail Address

Dealing with SPAM:

- Unfortunately, having your own website results in a lot of spam!
- If you need to include an e-mail address in your Web page, you can take a few steps to reduce problems with spam:
  - Replace all e-mail addresses in your page with inline images of those addresses
  - Write a program in JavaScript (or another language) to scramble any e-mail address in the HTML code
  - Replace the characters of the e-mail address with character codes
Linking to Various Interent Resources

- The URL for a Web page has the form
  \[ \text{http://server/path/filename#id} \]
  where \textit{server} is the name of the Web server, \textit{path} is the path to a file on that server, \textit{filename} is the name of the file, and if necessary \textit{id} is the name of an id or anchor within the file.
- The URL for an FTP site has the form
  \[ \text{ftp://server/path/filename} \]
  where \textit{server} is the name of the FTP server, \textit{path} is the folder path, and \textit{filename} is the name of the file.
- The URL for an e-mail address has the form
  \[ \text{mailto:address?header1=value1&header2=value2& ...} \]
  where \textit{address} is the e-mail address; \textit{header1}, \textit{header2}, etc. are different e-mail headers; and \textit{value1}, \textit{value2}, and so on are the values of the headers.
- The URL to reference a local file has the form
  \[ \text{file://server/path/filename} \]
  where \textit{server} is the name of the local server or computer, \textit{path} is the path to the file on that server, and \textit{filename} is the name of the file. If you are accessing a file on your own computer, the server name is replaced by a third slash (/).
Brief Interlude: Human Stupidity
Working with Hypertext Attributes

- HTML provides several attributes to control the behavior and appearance of your links
  - Opening a Secondary Window or Tab
    - By default, links open in the SAME page in the browser window
      - Meaning, they replace whatever page you were at!
    - Most developers (and business owners) would be against this
      - You don’t want customer’s to leave your Web site!
      - So you want your site to stay open
Working with Hypertext Attributes

- HTML provides several attributes to control the behavior and appearance of your links

- Opening a Secondary Window or Tab
  - You can force a document to appear in a secondary window or tab by adding the `target` attribute to the `<a>` tag
  - Syntax:
    
    ```html
    <a href="url" target="window">content</a>
    ```
    
    - where `window` is the name assigned to the new browser window or browser tab
  - If several links all have the same target name, they will all open in the same location, replacing the previous content
Working with Hypertext Attributes

- Target names for browser windows and tabs:

<table>
<thead>
<tr>
<th>Target Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>target</td>
<td>Opens the link in a new window or tab named target</td>
</tr>
<tr>
<td>_blank</td>
<td>Opens the link in a new, unnamed window or tab</td>
</tr>
<tr>
<td>_self</td>
<td>Opens the link in the current browser window or tab</td>
</tr>
</tbody>
</table>

Opening a Link in a New Window or Tab

- To open a link in a new browser window or browser tab, add the attribute `target="window"` to the `<a>` tag, where `window` is a name assigned to the new browser window or tab.
Working with Hypertext Attributes

- HTML provides several attributes to control the behavior and appearance of your links.
  - If you want to provide additional information to your users, you can provide a **tooltip** to your links.
    - A **tooltip** is a descriptive text that appears whenever a user positions the mouse pointer over a link.
  - Syntax:
    ```html
    <a href="url" title="text">content</a>
    ```
    - where text is the text that you want to appear in the tooltip.
  - Note: some browsers do not support tooltips.
    - So you should not place critical information here.
Working with Hypertext Attributes
Creating a Semantic Link

Semantic Links:
- Should ideally describe the document that you are linking to
- However, you can include two attributes that add information about the link:
  - rel and rev
- The rel attribute describes the content of the destination document
- Syntax:
  `<a href="glossary.htm" rel="glossary">Glossary</a>`
Creating a Semantic Link

Semantic Links:

- The **rev** attribute describes the reverse relationship: how the linked document views the current doc.
  - It complements the **rel** attribute by describing the contents of the source document as viewed from the destination document’s perspective.
  - For example, if you’re linking to the Glossary page from the home page, the reverse relation is “home”, as that is how the Glossary page views the home page.
  - The HTML code would be:
    ```html
    <a href="glossary.htm" rel="glossary" rev="home">Glossary</a>
    ```
Creating a Semantic Link

Semantic Links:

Links containing the `rel` and `rev` attributes are called semantic links because the tag contains information about the relationship between the link and its destination.

A browser can use the information that these attributes provide in many ways:

- for example to build a custom toolbar containing a list of links specific to the page being viewed
- Few browsers currently take advantage of semantic links, but future browsers may do so
Creating a Semantic Link

Built-in Semantic Link values:

<table>
<thead>
<tr>
<th>Link Relation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alternate</td>
<td>A substitute version of the current document, perhaps in a different language or in a different medium</td>
</tr>
<tr>
<td>appendix</td>
<td>An appendix</td>
</tr>
<tr>
<td>bookmark</td>
<td>A bookmark in a collection of documents</td>
</tr>
<tr>
<td>chapter</td>
<td>A document serving as a chapter in a collection of documents</td>
</tr>
<tr>
<td>contents</td>
<td>A table of contents</td>
</tr>
<tr>
<td>copyright</td>
<td>A copyright statement</td>
</tr>
<tr>
<td>glossary</td>
<td>A glossary</td>
</tr>
<tr>
<td>help</td>
<td>A help document</td>
</tr>
<tr>
<td>index</td>
<td>An index</td>
</tr>
<tr>
<td>next</td>
<td>The next document in a linear sequence of documents</td>
</tr>
<tr>
<td>prev</td>
<td>The previous document in a linear sequence of documents</td>
</tr>
<tr>
<td>section</td>
<td>A document serving as a section in a collection of documents</td>
</tr>
<tr>
<td>start</td>
<td>The first document in a collection of documents</td>
</tr>
<tr>
<td>top</td>
<td>The Web site’s home page</td>
</tr>
<tr>
<td>stylesheet</td>
<td>An external style sheet</td>
</tr>
<tr>
<td>subsection</td>
<td>A document serving as a subsection in a collection of documents</td>
</tr>
</tbody>
</table>
Working with Metadata

- What is metadata?
  - Metadata is information about your Web site

- Purpose of metadata:
  - Search engines use metadata so they can accurately index (store) information about your site and then present it to users as search results
  - Metadata allows you to store information about the document that can be read by the author, other users, or Web servers
  - Metadata can be used to control how the browser handles (views) the document
Working with Metadata

Adding metadata to your web page:

Add metadata to your Web pages by adding a meta element to the head section of the document:

```
<meta name="text" content="text" scheme="text" http-equiv="text" />
```

where name specifies the type of metadata, content stores the metadata value, scheme defines the metadata format, and http-equiv attaches commands to the communication stream between the Web server and the browser.

Example: the following metadata element stores the name of the Web page’s author:

```
<meta name="author" content="Gerry Hayward"/>
```
Working with Metadata

- Adding metadata to your web page:
  - For search engines, you should include metadata describing the site and its topics
  - This is done using the meta elements **description** and **keywords**
  - Example:
    ```html
    <meta name="description" content="CAMshots provides advice on digital cameras and photography" />
    <meta name="keywords" content="photography, cameras, digital imaging" />
    ```
# Working with Metadata

<table>
<thead>
<tr>
<th>Meta Name</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>author</td>
<td><code>&lt;meta name=&quot;author&quot; content=&quot;Gerry Hayward&quot; /&gt;</code></td>
<td>Supplies the name of the document author</td>
</tr>
<tr>
<td>classification</td>
<td><code>&lt;meta name=&quot;classification&quot; content=&quot;photography&quot; /&gt;</code></td>
<td>Classifies the document</td>
</tr>
<tr>
<td>copyright</td>
<td><code>&lt;meta name=&quot;copyright&quot; content=&quot;&amp;copy; 2011 CAMshots&quot; /&gt;</code></td>
<td>Provides a copyright statement</td>
</tr>
<tr>
<td>description</td>
<td><code>&lt;meta name=&quot;description&quot; content=&quot;Digital photography and advice&quot; /&gt;</code></td>
<td>Provides a description of the document</td>
</tr>
<tr>
<td>generator</td>
<td><code>&lt;meta name=&quot;generator&quot; content=&quot;Dreamweaver&quot; /&gt;</code></td>
<td>Indicates the name of the program that created the HTML code for the document</td>
</tr>
<tr>
<td>keywords</td>
<td><code>&lt;meta name=&quot;keywords&quot; content=&quot;photography,cameras,digital&quot; /&gt;</code></td>
<td>Provides a list of keywords describing the document</td>
</tr>
<tr>
<td>owner</td>
<td><code>&lt;meta name=&quot;owner&quot; content=&quot;CAMshots&quot; /&gt;</code></td>
<td>Indicates the owner of the document</td>
</tr>
<tr>
<td>rating</td>
<td><code>&lt;meta name=&quot;rating&quot; content=&quot;general&quot; /&gt;</code></td>
<td>Provides a rating of the document in terms of its suitability for minors</td>
</tr>
<tr>
<td>reply-to</td>
<td><code>&lt;meta name=&quot;reply-to&quot; content=&quot;ghayward@camshots.com (G. Hayward)&quot; /&gt;</code></td>
<td>Supplies a contact e-mail address and name for the document</td>
</tr>
</tbody>
</table>
Working with Metadata

- You can add information and commands to this communication stream with the meta element’s http-equiv attribute
  - Force the Web browser to refresh the Web page at timed intervals
    <meta http-equiv="refresh" content="60" />
  - Redirect the browser from the current document to a new document
    <meta http-equiv="refresh" content="5;url=www.camshots.com" />
  - Specify the character set
    <meta http-equiv="Content-Type" content="text/html;charset=ISO-8859-1" />
Tutorial Summary

- Create a Web site with several linked pages
- Storyboarding and complex Web structures
- Creating Web pages linked together
- Links to locations within documents and from another document
- Inline images and image maps
- Create links to sites and to non-Web locations
- Different hypertext attributes
- Meta element
Developing a Basic Web Site

WASN’T THAT AWESOME!
Daily Demotivator

BLAME

The Secret to Success is Knowing Who to Blame for Your Failures.
Tutorial 2: Developing a Basic Web Site