SENG1050

Web Technologies

Week 02

Lecture Plan

Weekly program (lectures)

- ✓ Week 1 – The Internet, Protocols, TCP/IP, Email, HTTP
- ❑ **Week 2 – HTML basics**
- ❑ Week 3 – XML and DTD
- ❑ Week 4 – CSS
- ❑ Week 5 – More HTML with CSS
- ❑ Week 6 – Revision and Midterm
- ❑ Week 7 – XSLT
- ❑ Week 8 – JavaScript
- ❑ Week 9 – More JavaScript and User Interface
- ❑ Week 10 – Encoding, Compression and Information Retrieval
- ❑ Week 11 – Security and Encryption
- ❑ Week 12 – Ethics and Course review

Key concepts from last lecture

- Data transferred using circuit switching and packet switching
- TCP/IP is a layered protocol model
- TCP and IP works together to transfer data
- IP addressing system: IPv4 and IPv6
- DNS maps domain names to IP address
- SMTP is the protocol for sending emails
- HTTP is the protocol for exchanging files on the Web

Week 02 Lecture 01 Outline

HTML basics

- Universal Resource Identifier (URI)
- Domain name
- HTML history
- Structure of an HTML file
- HTML Element Types and attributes
- Block Elements
- Inline Elements
- List Elements
- Linking

The World Wide Web

- Initial idea by Tim Berners-Lee.
  - http://www.w3.org/People/Berners-Lee/Overview.html

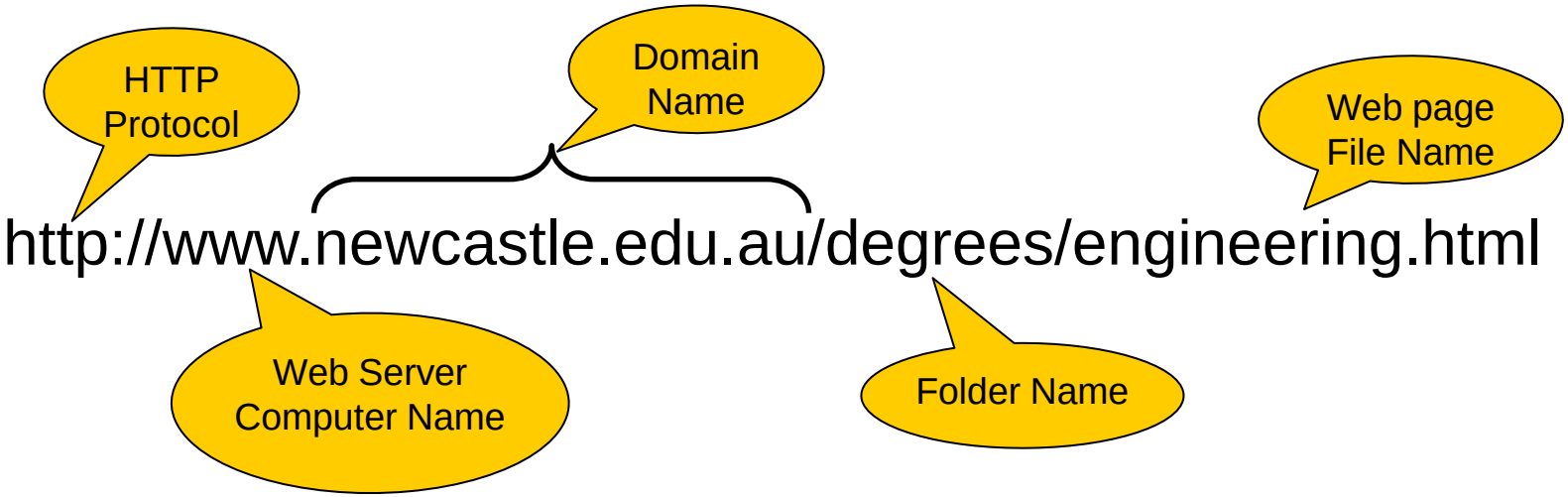
- Has three aspects:
  1. A scheme for locating distributed documents
     - URI: URL or URN
  2. A protocol for accessing these documents
     - HTTP
  3. A “hypertext” language linking together the documents
     - HTML

Scheme for locating distributed document - URI

- Universal Resource Identifier (URI)
  - A string that Identifies a resource on the internet by location by name or both
  - Two specializations
    - Universal Resource Locator (URL)
    - Universal Resource Name (URN)
      - Does not imply availability of the identified resource
      - Uses the URN scheme: urn:<NID>:<NSS>
      - Example:
        - » urn:isbn:0451450523
        - » urn:mpeg:mpeg7:schema:2001

Universal Resource Locator - URL

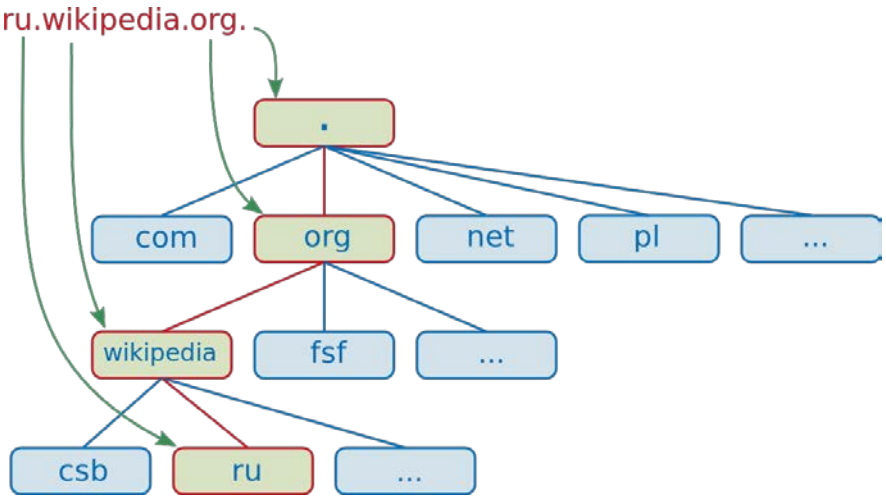
- Represents the network location (physical address) of a resource such as webpage, a graphic file, or an MP3 file.



Domain Name

- A unique name that locates an organization, a web site or other entity on the internet.
- **Purpose**:
  - divide the internet into logical groups
  - easily memorable names for internet resources
    - Can use http://www.google.com instead http://74.125.237.115/
- **Organization**: hierarchical

Domain Name



- Top-level domain (TLD): identifies the rightmost part of the domain name.
- TLD is
  - Either a generic top-level domain: .com, .edu, .org,
  - Or a country-code top-level domain: .au, .fr, .jp

www.movies.yahoo.co.jp

Hypertext

- **Hypertext**: “a body of written or pictorial material [emphasis on text] interconnected in a complex way that it could not be conveniently represented on paper. It may contain summaries or maps of its contents and their interrelations; it may contain annotations, additions and footnotes from scholars who examined it.” – Ted Nelson

- **Hypermedia**: The combination of hypertext and multimedia [emphasis on sound, video, etc.] in an electronic document.

HTML

- HTML = HyperText Markup Language
  - The file format proposed by Tim Berners-Lee to encode hypertext documents for the World-Wide Web
  - Still used for the majority of Web pages …
  - … although sometimes generated from other sources …
  - … and this is slowly changing

HTML

- HTML has tags that…
  - influence the visual presentation of the document
  - insert images and other media
  - add hyperlinks taking the user from this document to another document
  - create forms that the user can fill in and submit back to the Web server
  - insert small computer programs to be run by the user's browser

The History of HTML

Core HTML specification

The combined timelines for HTML 5.0, HTML 5.1 and HTML 5.2:

	2012	2013	2014	2015	2016
HTML 5.0	Candidate Rec	Call for Review	Recommendation		
HTML 5.1	1st Working Draft		Last Call	Candidate Rec	Recommendation
HTML 5.2[29]				1st Working Draft	

From Wikipedia

Browser Support

Audiences of your website

- Two audiences
  - Human
  - Spiders, bot, web-crawler

Wikipedia:

> "A **Web crawler** is an Internet bot that systematically browses the World Wide Web, typically for the purpose of Web indexing.
>
> Web search engines and some other sites use Web crawling or spidering software to update their web content or indexes of others sites' web content. Web crawlers can copy all the pages they visit for later processing by a search engine that indexes the downloaded pages so that users can search them much more quickly."

Audiences or your website

- As a web designer you need to take care of both of your audiences
  - Human
    - They get the needed information easily and possibly in an interesting and nicely presented
  - Spiders, bot, web-crawler
    - They can identify the information you want to spread and appropriately index them that your website has a very high hit count

Document Type Definition (DTD)

- **Purpose**: To identify the type of markup language used in the document
- DTD identifies the version of HTML
- Browser and code validators use that information for processing
- Placed at the top of the web page document
- Common DOCTYPE declarations
- HTML5

```
<!DOCTYPE html>
```

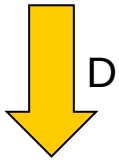
- HTML4.01

```
<!DOC                    HTML PUBLIC "-//W3C//DTD HTML 4.01 T          al//EN"
"http://                                    4/html4/loose.dtd >
```

HTML

- A HTML document is a “plain text” document in which special markup symbols or “**tags**” are used
  - Tags identify different structural elements
  - There are tags to say “this piece of text is a heading”, “this piece of text is a paragraph”, “this piece of text is an item in a list”
  - Tags are generally not displayed by the browser

```
<p>This is a paragraph</p>
```



Displayed as

This is a paragraph.

HTML element syntax

- An HTML element starts with a **start tag / opening tag**
- An HTML element ends with an **end tag / closing tag**
- Some HTML elements have **empty content**
- Empty elements do not have any **end/closing tag**
- Most HTML elements can have **attributes**

The structure of an HTML5 document


```
<!DOCTYPE html>
<html>
        <head>
                <meta charset="UTF-8">
                <title>Title of the document</title>
        </head>

        <body>
        Content of the document......
        </body>

</html>
```

Marking elements with tags

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

<element> content </element>

```
<p>This is a paragraph</p>
```

- An HTML element is everything from start tag to end tag

The structure of an HTML5 document

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

- Inside **<html>** there must be a single **<head>** and a single **<body>**
  - **<head>** contains information about the document – not (usually) displayed by the browser
  - **<body>** contains the displayed content of the document

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Title of the document
        </title>
    </head>

    <body>
        Content of the
document.....
    </body>

</html>
```

The structure of an HTML5 document

- There are several tags that belong inside **`<head>`**
  - **`<title>…</title>`** is used to define the title of the document
    - This is *not* the same as a heading
    - It is often displayed in the browser tab– but it may not be
    - provides a title for the page when it is added to favorites
    - displays a title for the page in search-engine results
  - **`<meta>`** provides metadata about the HTML document.
    - Metadata will not be displayed on the page, but will be machine parsable.
    - Typically used to specify page description, keywords, author of the document, last modified, and other metadata.
    - **`charset`**: The character encoding for the HTML document

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Title of the document
        </title>
    </head>

    <body>
        Content of the
document.....
    </body>

</html>
```

The structure of an HTML5 document

- The rest of the tags go inside **<body> </body>**
- Most HTML elements can be nested
  - One HTML element can contain another HTML element inside it

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Title of the document
        </title>
    </head>

    <body>
        Content of the
document......
    </body>

</html>
```

The first HTML5 document

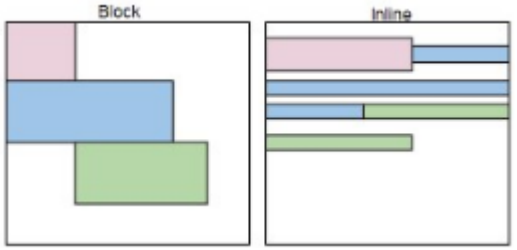

```
<!DOCTYPE html>
<html>
        <head>
                <meta charset="UTF-8">
                <title>Title of the document</title>
        </head>

        <body>
                Hello World!

        </body>

</html>
```

HTML Element Types



- HTML Elements can be divided into two groups:
  - **Block element**
    - Block level elements normally start (and end) with a new line when displayed in a browser.
      - For example: paragraphs, headings, lists, tables
  - **Inline element**
    - Inline elements are normally displayed without starting a new line.
      - For example: hyperlinks, emphases, inline images

HTML Element attribute

- HTML elements can have **attributes**
- Attributes provide **additional information** about an element
- Attributes are always specified in **the start tag**
- Attributes come in name/value pairs like: **name="value“**

Example:

```
<meta charset="UTF-8">

<a href="http://www.google.com"> Google Search </a>
```

Heading Elements

- **<h1>…</h1>** – the most important heading in the document (there will usually only be one of these, e.g. Title of the document)
  - **<h2>** – headings slightly less important
  - **<h3>**, **<h4>**, **<h5>**
  - **<h6>** - the least important heading

<h1>h1 tag</h1>

<h2>h2 tag</h2>

<h3>h3 tag</h3>

<h4>h4 tag</h4>

<h5>h5 tag</h5>

<h6>h6 tag</h6>

Heading Elements - Importance

- Google likes them!!
- In generating the ranking of relevant documents Google takes in to account many things – heading is one of them
- Google assumes the contents of <h1> tags are more important than <h2> tags and so on

- Then why don't add many <h1> elements in your page ?
  - Google does not like that!
  - Spamdexing

HTML – Examples


```
<body>
  <h1>My Book</h1>
   <h2>Chapter 1</h2>
    <h3>Section 1.1</h3>
    <h3>Section 1.2</h3>
     <h4>Section 1.2.1</h4>
    <h3>Section 1.3</h3>
   <h2>Chapter 2</h2>
    <h3>Section 2.1</h3>
   <h2>Chapter 3</h2>
</body>
```

My Book

Chapter 1

Section 1.1

Section 1.2

Section 1.2.1

Section 1.3

Chapter 2

Section 2.1

Chapter 3

HTML – Block Elements

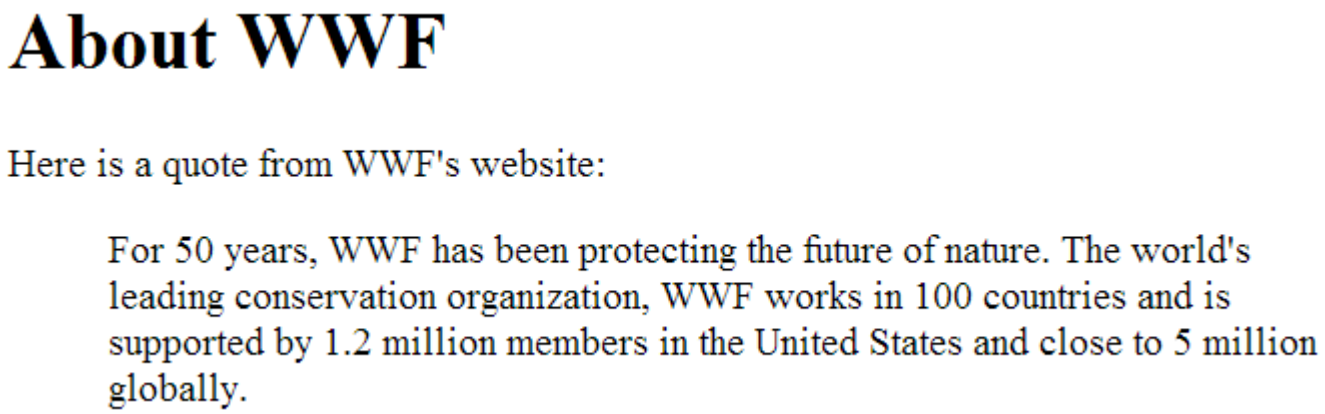
- **`<p>…</p>`** – a paragraph of text
  - Attribute: **`align`** can one of {“left”,”right”,”center”,”justify”}
  - Not supported in HTML5

- **`<blockquote>…</blockquote>`** – a block of text quoted from someone/somewhere else
  - Attribute: **`cite`**, Value URL; species the source of the quotation
  - Avoid using blockquote just to indent the text

HTML – Examples


```
<body>
   <h1>About WWF</h1>
   <p>Here is a quote from WWF's website:</p>

   <blockquote cite="http://www.worldwildlife.org/who/index.html">
        For 50 years, WWF has been protecting the future of nature. The world's
        leading conservation organization, WWF works in 100 countries and is
        supported by 1.2 million members in the United States and close to 5
        million globally.
   </blockquote>
</body>
```



HTML – Block Elements

- **`<address>…</address>`** – contact information for the author/owner of a document or an article.
- If the <address> element is inside the <body> element, it represents contact information for the document.
- If the <address> element is inside an <article> element, it represents contact information for that article.
- The text in the <address> element usually renders in *italic*. Most browsers will add a line break before and after the address element.
- Many browsers add a line break before and after the address element.

HTML – Examples


```
<body>
  <h1>Contact details</h1>
  <address>
    Mickey Mouse <br>
    Room MMXYZ <br>
    Disneyland, USA.<br>
  </address>
</body>
```

HTML – Block Elements

- **`<div>…</div>`** – a generic division of the document
  - Defines a division or a section in an HTML document.
  - Used to group tags together and apply formatting
  - Used to group block-elements to format them with CSS.
  - Attribute **`align`** Not supported in HTML5

HTML – Break elements

- **
** – goes to the next line before displaying the next element or portion of the text.
  - **Note 1:** Not a paired element, **stand-alone** or **void** element
  - **Note 2:** In XHTML line break tag is **
**
  - Use the
 tag to enter line breaks, not to separate paragraphs.

- **<hr>** – draw a horizontal rule between blocks to visually separate those areas
  - This is also a void element
  - XHTML syntax is: **<hr />**

HTML – Examples


```
<body>

    <p>This is some text before the article.</p>
    <hr>
    <div>
        <h2>This is the heading in a div element</h2>
        <p>This is content of the article in a div element.</p>

        <address>
                Written by: Mickey Mouse.<br>
                Visit us at:<br>
                Example.com<br>
                Box 564, Disneyland<br>
                USA
        </address>
    </div>
    <hr>
    <p>This is some text after the article.</p>
</body>
```

HTML – Examples

This is some text before the article.

This is the heading in a div element

This is content of the article in a div element.

Written by: Mickey Mouse.
Visit us at:
Example.com
Box 564, Disneyland
USA

This is some text after the article.

HTML – Inline Elements

- Inline tags are placed around phrases of text
- … to make the text more or less important
- **…** – the enclosed text is important and displayed in bold
- **…** – the makes the enclosed text bold.
- They look the same to human but not to spiders
- According to the HTML 5 specification, the tag should be used as a LAST resort when no other tag is more appropriate. The HTML 5 specification states that headings should be denoted with the <h1> to <h6> tags, emphasized text should be denoted with the tag, important text should be denoted with the tag, and marked/highlighted text should use the <mark> tag.

HTML – Inline Elements

- **…** – the enclosed text should be emphasised.
- **…** – the enclosed text should defines a part of text in an alternate voice or mood. The content of the tag is usually displayed in italic.
  - The tag can be used to indicate a technical term, a phrase from another language, a thought, or a ship name, etc.
- Use the element only when there is not a more appropriate semantic element, such as: **** (emphasized text), **** (important text), **<mark>** (marked/highlighted text), **<cite>** (the title of a work), **<dfn>** (a definition term)

HTML – Examples


```
<p>
  I want to <em>emphasise</em>
  this but <i> not this </i>. <br><br>

  In fact, I want to <strong>strongly
   emphasise</strong> this but <b>not this </b>.
</p>
```

I want to *emphasise* this but *not this* .

In fact, I want to **strongly emphasise** this but **not this** .

HTML – Phrase elements

- A set of elements that apply to a specific type of text content
- Initially existed out of a need for formatting, but have since come to assume a new role as semantical text identifiers in HTML5.
- … to add information about what type text is contained in the tag

- **`<code>…</code>`** – the enclosed text is a piece of computer code

- **`<samp>…</samp>`** – the enclosed text is a sample of (program) output

- **`<kbd>…</kbd>`** – the enclosed text represents input that would be typed on a keyboard
- Note: **`<samp>, <code>`** **and** **`<kbd>`** do not have the same semantic meaning

HTML – Examples


```
<p>
  If you compile the C code
  <code>scanf("%f", &amp;scale);
  printf("%f\n", scale*2);</code>
  and enter <kbd>21.12</kbd> on
  the keyboard, then the program
  will output <samp>42.24</samp>.
</p>
```

If you compile the C code `scanf("%f", &scale); printf("%f\n", scale*2);` and enter `21.12` on the keyboard, then the program will output `42.24`.

HTML – Phrase Elements

- **`<cite>…</cite>`** – In HTML5, the <cite> tag defines the title of a work.
  - Note: A person's name is not the title of a work.
  - In HTML 4.01, the <cite> tag defines a citation.

- **`<q cite="">…</q>`** – the enclosed text is a short quotation from another source
  - Has a **`cite`** attribute for the URI of the source

HTML – Examples


```
<p>
  There is a computing "urban legend"
  that Bill Gates once said, <q
  cite="http://archive.wired.com/politics/law/news/1997/01/1484">
  640K of memory should be enough
  for anybody</q>
  (Jon Katz, <cite>Did Gates Really
  Say 640K is Enough For Anyone?</cite>,
  Wired News, 16 Jan 1997).
</p>
```

There is a computing "urban legend" that Bill Gates once said, "640K of memory should be enough for anybody" (Jon Katz, *Did Gates Really Say 640K is Enough For Anyone?* , Wired News, 16 Jan 1997).

HTML – Phrase Elements

- `<dfn>…</dfn>` – the enclosed text is a definition term
- `<var>…</var>` – the enclosed text is a variable

HTML – Inline Elements

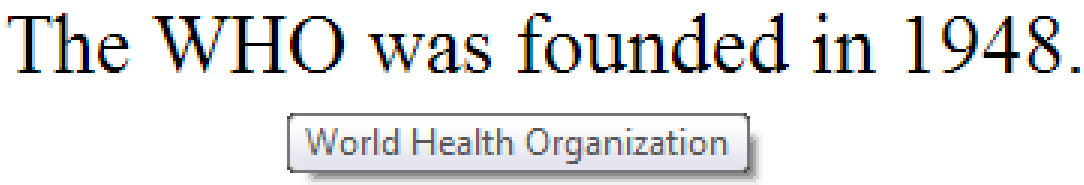
- **<abbr>…</abbr>** – tag indicates an abbreviation or an acronym, like "WWW" or "NATO".
  - By marking up abbreviations you can give useful information to browsers, spell checkers, translation systems and search-engine indexers.

```
The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.
```

- **<acronym>…</acronym>** – the enclosed text as an acronym
  - **<acronym title="World Wide Web"> WWW</acronym>**
  - Is not Supported in HTML5

HTML – Examples


```
<p>
  The <abbr title="World Health
  Organization">WHO</abbr> was
  founded in 1948.
</p>
```



HTML – Inline Elements

- **[…]** – the enclosed text is a mathematical superscript
  - Superscript text appears half a character above the normal line, and is sometimes rendered in a smaller font.
  - Example: ABC[XYZ]

- **_…** – the enclosed text is a mathematical subscript
  - Subscript text appears half a character below the normal line, and is sometimes rendered in a smaller font.
  - Example: H_2O

HTML – Examples


```
<p>
  <dfn>root<sub>1</sub> = [ -<var>b</var> +
    sqrt( <var>b</var><sup>2</sup> -
      4<var>a</var><var>c</var> ) ] /
    2<var>a</var>
  </dfn><br>
  <dfn>root<sub>2</sub> = [ -<var>b</var> -
    sqrt( <var>b</var><sup>2</sup> -
      4<var>a</var><var>c</var> ) ] /
    2<var>a</var>
  </dfn>
</p>
```

$$root_1 = [\ -b + sqrt(\ b^2 - 4ac\ )\ ]\ /\ 2a$$

$$root_2 = [\ -b - sqrt(\ b^2 - 4ac\ )\ ]\ /\ 2a$$

HTML – Inline Elements

- **`<span>…</span>`** – generic grouping of inline text – used to apply formatting
  - Does the same job as **`<div>`** but does not start a new block of text
  - Provides no visual change by itself.
  - Provides a way to add a hook to a part of a text or a part of a document.
    - When a text is hooked in a element, you can style it with CSS, or manipulate it with JavaScript.

HTML – Inline Tags Nesting Rules

- You should not (cannot) use inline tags as immediate children of **<body>**
- Should be nested within a block element, such as **<p>**.
- Inline tags can only contain other inline tags

HTML – List Elements

- Special block tags let you create lists
- **`<ol>…</ol>`** – an ordered list of items
- Attributes:
  - Attribute **`start`**=number - specifies the start value of an ordered list
  - Attribute **`type`**={1,A,a,I,i} - specifies the kind of marker used in the list
  - Attribute **`reversed`** - specifies that the list order should be descending [New in HTML5]
  - Attribute: **`compact`** not supported in HTML5

HTML – List Elements

- **`<ul>…</ul>`** – an unordered list of items
- Attributes:
  - Attribute: **`type`** not supported in HTML5
  - Attribute: **`compact`** not supported in HTML5

- **`<li>…</li>`** – an item in a list
  - Used both in <ul> and
  - Attributes:
    - Attribute: **`value`**=number - Specifies the value of a list item. The following list items will increment from that number (only for **`<ol>`** lists)
    - Attribute: **`type`** not supported in HTML5

HTML – List Elements

- **<dl>…</dl>** – a description list
  - Used in conjunction with **<dt>** (defines terms/names) and **<dd>** (describes each term/name).

- **<dt>…</dt>** – within a description list, a term/name to be defined

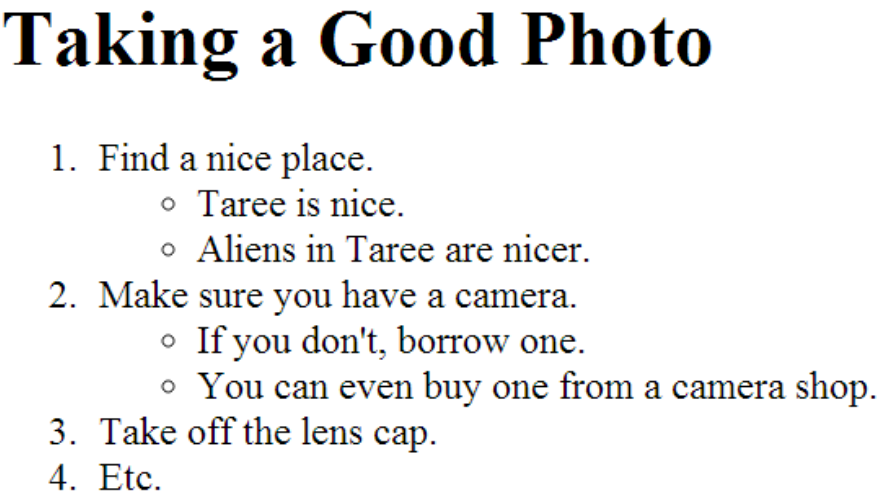
- **<dd>…</dd>** – within a description list, the describe a term

HTML – List Elements

- List tags should only contain the appropriate item tags as immediate children
- `<dt>` can only contain inline tags
- `<li>` and `<dd>` can contain inline and block tags – allows nested lists!

HTML – Examples


```
<body>
<h1>Taking a Good Photo</h1>
<ol>
<li>Find a nice place.
  <ul>
    <li>Taree is nice.</li>
    <li>Aliens in Taree are nicer.</li>
  </ul></li>
<li>Make sure you have a camera.
  <ul>
    <li>If you don't, borrow one.</li>
    <li>You can even buy one from a camera shop.</li>
  </ul></li>
<li>Take off the lens cap.</li>
<li>Etc.</li>
</ol>
</body>
```



HTML – Examples


```
<dl>
  <dt>HTML</dt>
    <dd>
      Hyper Text Markup Language
    </dd>
  <dt>SMTP</dt>
    <dd>
      Simple Mail Transfer Protocol
    </dd>
  <dt>FTP</dt>
    <dd>File Transfer Protocol</dd>
</dl>
```

HTML

Hyper Text Markup Language

SMTP

Simple Mail Transfer Protocol

FTP

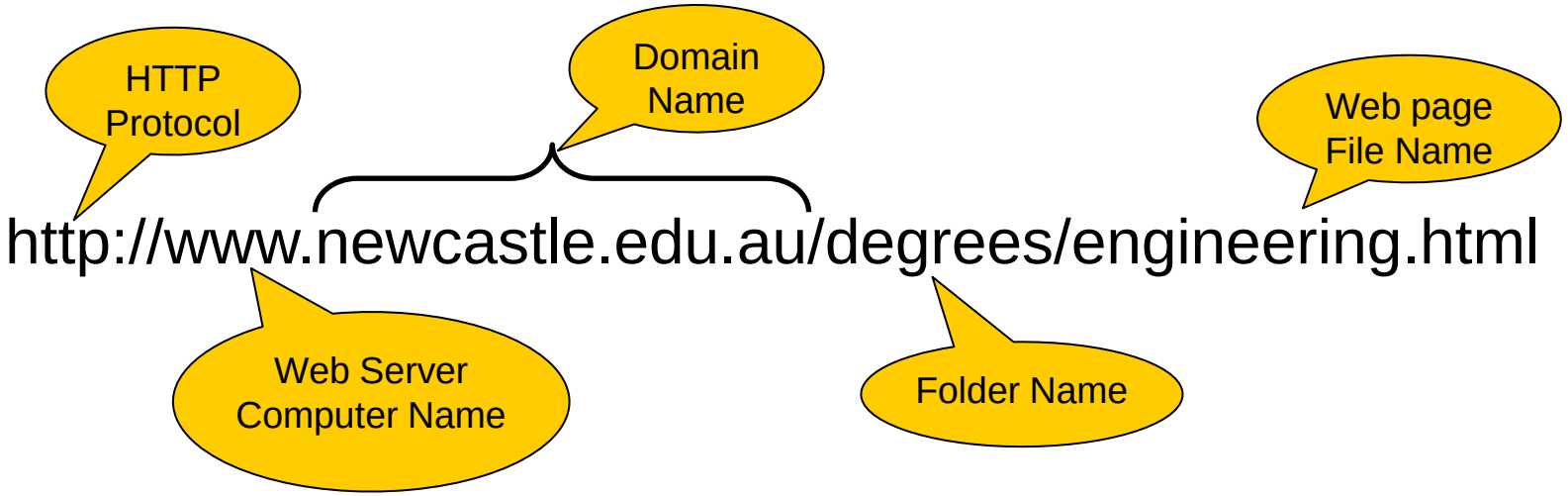
File Transfer Protocol

HTML – Linking

- **…** – create a hyperlink from the enclosed content to the document named in the URI
  - Called an anchor
  - When the link is activated in the browser, the linked document is requested and (all going well) displayed in the browser
  - **<a>** is an inline tag

HTML – Linking

- URL = Uniform Resource Locator
  - The physical address for a resource
  - Much more commonly used



HTML – Linking

- Absolute URL
  - Enough information to access the document from any other document
  - At least protocol, host and path are specified

- Relative URL
  - Only enough information if the context of the current document is used

HTML – Linking

- Relative URL
  - Protocol/host default to same as current document
  - Port defaults to number appropriate for protocol (http = 80, ftp = 21) – host can “override” this
  - Path defaults to the same as the current document
  - Paths without leading / are relative to current path
  - Name default determined by host (usually index.html)
  - Id defaults to nothing (top of document)
  - Query defaults to nothing

HTML – Linking

- Relative URL Example
  - I am currently adding a link to (editing the file)

```
http://compseng.newcastle.edu.au/abc123/index.html
```

  - I want put in a link to my research html file

```
http://compseng.newcastle.edu.au/abc123/research.html
```

  - I could use the relative URL

```
<a href=“research.html">Current Research</a>
```

  - or the absolute URL

```
<a href="http://compseng.newcastle.edu.au/abc123/
  research.html">Current Research</a>
```

HTML – Linking

- Another Relative URL Example
  - If I want to add a link to my main page from

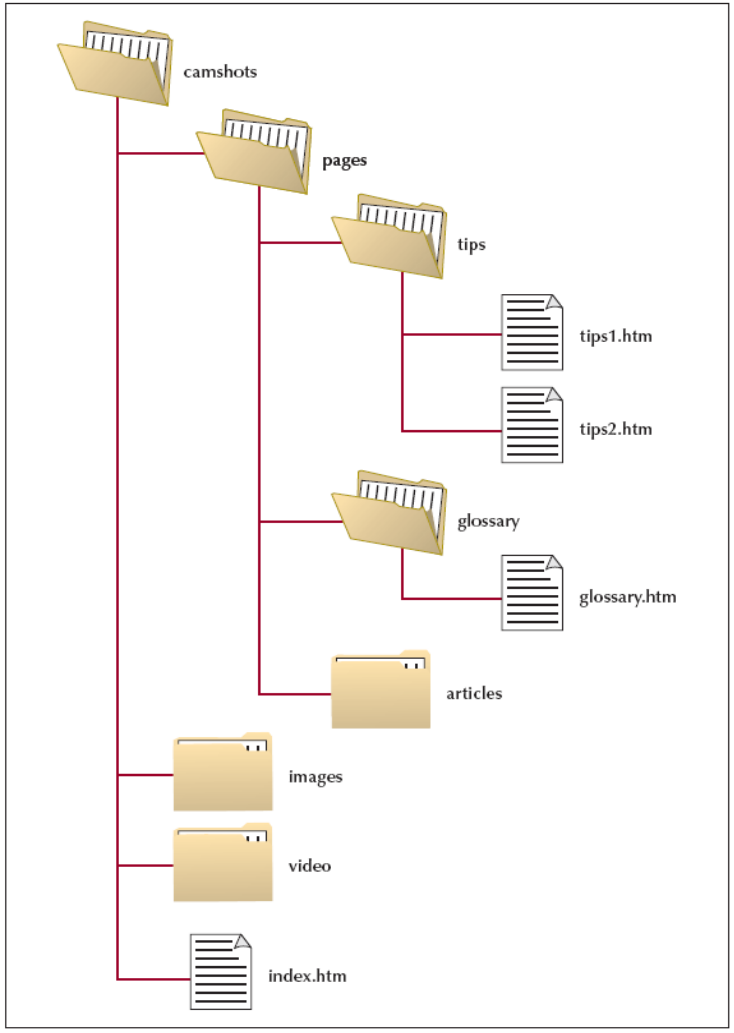
    `http://compseng.newcastle.edu.au/abc123/lab/research.html`
  - I can use the relative URL

    `<a href="../index.html">back to the lab home page</a>`
  - I could also use the relative URL

    `<a href="..">back to my lab home page</a>`

    as **`index.html`** is the default name on the **`compseng.newcastle.edu.au`** Web server.

Specifying a folder path



Specifying a folder path

Absolute Path	Interpretation
/camshots/pages/tips/tips1.htm	The tips1.htm file located in the pages/tips subfolder
/camshots/pages/tips/tips2.htm	The tips2.htm file located in the pages/tips subfolder
/camshots/pages/glossary/glossary.htm	The glossary.htm file located in the pages/glossary subfolder
/camshots/index.htm	The index.htm file located in the camshots folder

Relative Path from the /camshots/pages/tips Subfolder	Interpretation
tips1.htm	The tips1.htm file located in the current folder
tips2.htm	The tips2.htm file located in the current folder
../glossary/glossary.htm	The glossary.htm file located in the sibling glossary folder
../../index.htm	The index.htm file located in the parent camshots folder

HTML – Linking

- If the document you are linking to is part of your Web site, then USE RELATIVE URLS!
- Only if the document is part of a different Web site (on a different server, or on the same server but controlled by someone else) should you use absolute URLs
- IMPORTANT: In this way, you can “zip up” your entire Web site, move it to another Web server, and it will still work!

HTML – Linking

- Linking to Identifiers
  - The URL **`/jfr457/research/index.html#gd`** requires there to be something in the requested document that says “identifier **`gd`** points to this spot”
  - **`<`*tag*` id="gd">…</`*tag*`>`** or **`<`*tag*` id="gd" />`**
  - Practically every tag allows the **`id`** attribute
  - If there is no matching **`id`** attribute, then the browser will point to the bottom of the document … or the top … or where you were last time you loaded this document – can cause confusion

HTML – Linking

- To check whether your links are correct (correct URL structure and point to existing documents), you can
  - Manually try every link to see if it takes you to the correct document, or
  - Use an automatic link checking tool, such as http://validator.w3.org/checklink

HTML – Special Characters

- HTML can display the full-range of Unicode characters independent of the **<?xml?>** encoding
  - ***&#number;*** – displays character with given Unicode number
  - ***&name;*** – displays a "named" character
    - **&** = **&** (ampersand)
    - **<** = **<** (less than)
    - **>** = **>** (greater than)
    - **é** = **é** (**e** with acute accent)
  - **Must** use **& < >** – not **& < >**

HTML – Images

- **`<img src="`*uri*`" alt="`*text*`" height="42" width="42" />`**
  - Loads the image file from ***uri*** and inserts it into the document at this point
  - The image is displayed with 42 pixel height X 42 pixel width
  - If the image cannot be loaded or displayed, then display the alternative ***text*** instead
  - **`<img>`** is an inline tag
  - **`src`** and **`alt`** are required attributes
  - JPEG, GIF and PNG image formats are supported by nearly all browsers – avoid system-specific formats like BMP, XBM, PICT

HTML – Comments

- **<!-- *comment text* -->**
  - Any text between **<!--** and **-->** is ignored by the browser
  - Use this to embed notes for yourself and other Web developers in your Web pages
  - Just like any other programming language, XHTML should be commented wherever the intent of the author is not clear from the code
  - Commenting can also be used to embed metadata, such as authorship and keywords – but there are other ways to do this…

HTML – Validation

- If there is a mistake in your HTML
  - Some browsers will refuse to display the page
  - Some browsers will report that the page is invalid but try to display it anyway
  - Some browsers will try to guess what it thought you really wanted and display that!

Just because it looks right on your browser, doesn't mean it is right!

HTML – Validation

- For all the gory details consult
  - http://www.w3.org/TR/html5/
  - http://www.w3.org/TR/xhtml1/
  - http://www.w3.org/TR/html4/

- Learn to use an XML/HTML validator
  - Checks your pages to ensure they conform to the standards
  - http://validator.w3.org/
  - http://www.htmlhelp.com/tools/validator/upload.html

Visual Formatting

- Almost all tags imply suggestions to the browser about how to draw their content
  - Headings change the font size and weight
  - `<blockquote>` indents the margins
- These are only suggestions!
  - A browser can ignore them
  - It may not even support them – e.g., a mobile phone

Summary

- History of hypertext
- Learnt about the basic elements of HTML
- Types and attributes of HTML elements
- Difference between block and Inline elements
- Importance of choosing proper tags so that you spiders can identify the importance of your page
- Phrase elements and other inline elements
- Different types of list elements
- How to link different HTML documents

References

- **Web Development and Design Foundations with HTML5 (6e)**

  By Terry Felke-Morris

  - *Chapters 1 and 2*

- http://www.w3schools.com/html/default.asp

- http://www.w3.org/ – the first place to look for all things related to Web standards