



## CENTER FOR TEXTUAL STUDIES AND DIGITAL HUMANITIES

# DIGH 401 - Introduction to Computing

Fall Semester 2014

Week 13

## Today's Class

- Week 12 Exercise
- Basic web design
  - HTML
  - CSS
  - HTML 5

## Week 12 Exercise

Reimagine using Collections or Dictionaries

- a simple check-in and check-out system for books

Recreate a basic collection in PHP

## Week 12 Exercise

Reimagine using a collection pattern - a simple check-in / check-out...

[Example 1](#) | [Example 2](#)

[Code comparison](#)

## HTML

### Intro

- HyperText Markup Language (HTML)
- HTML relies on keywords or element tags
- HTML can also use attributes within opening element tags
- keywords follow a rigidly defined syntax
- HTML creates web pages that web browsers can view
- an error or bug may cause the page to not render or simply render incorrectly
- to understand the current core of web page designing you need to know at least the basics of HTML

## HTML - Elements and attributes

### Element syntax

- start with an opening element tag, and close with a closing tag
- content is everything between opening and closing element tags
- elements can contain empty content
- empty elements should be closed in the opening tag
- most elements permit attributes within the opening tag

### Attribute

- attributes provide additional information to the parent element
- always added to the opening tag
- standard syntax of name/value pairs, class="401"
- standard attributes include
  - class
  - id
  - style
  - title

## HTML

### Structure of HTML

- basic HTML tag defines the entire HTML document

```
<html>
```

```
</html>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

```
<html>
```

```
<head>
```

```
...
```

```
</head>
```

```
<body>
```

```
...
```

```
</body>
```

```
</html>
```

## HTML - working within the <body>

- to define the main body of the web page we use the <body> element
- headings can be created using variants of
  - <h1>, <h2>.....<h6>
- we can now add some simple text in a <p> element

<p>...</p>

- add a line break using the <br /> element
- <hr /> element adds a horizontal line
- comments can also be added through our HTML

<!-- comment... -->



## HTML - working within the <body>

### Formatting some text

- formatting can be considered relative to stylistic and semantic requirements
- formatting is also available for embedded 'code' viewing
- text formatting includes
  - bold <b>, emphasis <em>, italic <i>, strong <strong>, sub and superscripted <sub> <sup>, inserted & deleted <ins> <del>...
- computer 'code' formatting includes
  - code <code>, variables <var>, pre-formatted text <pre>...
- quotations, citations and definitions include
  - abbreviations <abbr>, acronyms <acronym>, citation <cite>, definition <dfn>...

## HTML - working within the <body>

### Linking in HTML

### [Example](#)

- linking is an inevitable part of web design and HTML usage
- can be considered within three different contexts
  - linking to an external site
  - linking to another page within the same site
  - linking different parts of the same page
- add links to text and images within the HTML
- <a> element for links plus required attributes
  - <a href="http://www.google.com/">Google</a> or  
<a href="mailto:name@email.com">Email</a>
  - <a href="/another\_page.html">another page</a>
  - <a name="anchor">Internal Anchor</a> or  
<a id="anchor">Anchor</a>
  - <a href="#anchor">Visit Internal Anchor</a> or  
<a href="/another\_page.html#anchor">Visit External Anchor</a>

## HTML - working within the <body>

### Linking in HTML - continued

- standard attributes supported by <a> element include
  - class, id, lang, style, title...
- optional attributes are available for <a> element including
  - target, href, name...
- target attribute specifies where the link will be opened relative to the current browser window. Possible attribute values include
  - \_blank
  - \_self
  - \_parent
  - \_top

## HTML - working within the <body>

### Working with images

- <img> allows us to embed an image within a web page
- <img> element requires a minimum 'src' attribute

```

```

- other optional attributes include
  - class, id, alt, title, width, height...
- use images as links
- image maps

```
<map name="textmap">  
  <area shape="rect" coords="..." alt="Quote 1" href="notes1.html" />  
</map>
```

## HTML - working within the <body>

### Adding a table

- organise data within a table starting with the <table> element
- three primary child elements include
  - <tr>, <th>, <td>

```
<table>
<tr>
<th>header 1</th>
</tr>
<tr>
<td>row 1, cell 1</td>
</tr>
</table>
```

- also add a <caption>, span multiple columns using the 'colspan' attribute, or span multiple rows using the 'rowspan' attribute

## HTML - working within the <body>

### Organising a list

- unordered list <ul>, ordered list <ol>, definition list <dl>
- <ul> and <ol> contains list items <li>

```
<ul>  
<li>...</li>  
</ul>
```

```
<ol>  
<li></li>  
</ol>
```

- definition list uses <dt> for the item, and <dd> for the definition

```
<dl>  
<dt>Game 1</dt>  
<dd>our definition</dd>  
</dl>
```

## HTML - working within the <body>

### Using forms

- used to capture data input by a user, which can then be processed by the server
- <form> element acts as the parent wrapper for a form
- <input> element for user input includes options using the 'type' attribute
  - text, password, radio, checkbox, submit

<form>

Text field: <input type="text" name="textfield" />

</form>

## HTML - working within the <body>

### Processing forms

```
<form action="process_form.php" method="post">  
Name: <input type="text" name="name" />  
Age: <input type="text" name="age" />  
<input type="submit" />  
</form>
```

```
<?php
```

```
$name = $_POST['name'];  
$age = $_POST['age'];
```

```
echo 'Name = '.$name.'<br />';  
echo 'Age = '.$age.'<br />';
```

```
?>
```

[Example](#)