



Intro to: Final Project, HTML, PHP

A look at things to come



The Final Project

- A small web based database application
- Should have at least 4 entity tables
- Should have at least 4 relationships (including at least one many-to-many relationship)
- Can be done in a group of two
- Should use **PHP** and **HTML** to create the web interface



HTML

- **HTML** is a markup language
- Groups and defines data
- Defines some layout
- **CSS** is responsible for most styling



A basic HTML page

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<title>My first web page</title>
</head>
<body>
This is my first web page
</body>
</html>
```



Basic elements

- !DOCTYPE is not part of the page, lets browser know what you are going to do
- <tag>..</tag> everything in HTML goes in tags
- <html> contains HTML
- <head> is stuff that is not rendered on the page
- <body> is all the actual content of the page



Other important layout tags

- <p> creates a paragraph
-
 creates a line break (it opens and closes in a single <>)
- a link creates a link that looks like [a link](#) and goes to foo.com
- <hX> where X is a number 1-6 creates heading, h1 being the largest, h6 being the smallest



Lists

```


- <li>foo</li>
- <li>bar</li>
- .
- <li>baz</li>

```

This creates a list of items



Tables

```

<table>
  <tr>
    <td>row 1 column 1</td>
    <td>row 1 column 2</td>
  </tr>
  <tr>
    <td>row 2 column 1</td>
    <td>row 2 column 2</td>
  </tr>
</table>

```



Forms

- Forms will be your primary means of passing data to a database

```

<form action="processform.php" method="post">
  <fieldset>
    <legend>Text and Password</legend>
    <label for="username">Username:</label>
    <input name="username" id="username" value="Some Text" />
    <label for="password">Password:</label>
    <input type="password" name="password" id="password" value="Password" />
    <input type="submit" name="submit" />
  </fieldset>
</form>

```



Forms continued

- `<form>` all form stuff goes in here
- `<fieldset>`, `<legend>`, `<label>` fancy things to make the form look better
- `<input>` where data is pulled from when the form is sent also can be used to submit form



Forms, input

- Types of input (`<input type="X">`)
 - Text
 - Password
 - Checkbox
 - Radio
 - Submit
 - Reset
 - Hidden
 - Button



Forms, submit

- `<form action="foo.php" method="post">` means that when the form is submitted it will be *posted* to foo.php
- `<input type="submit" name="bar">` submits the form and will add a variable named "bar", more on this later



More information

- If you want more check out htmldog.com
- Also the developer tools in Google Chrome and Firebug in Mozilla Firefox are great for looking at existing websites



Intro to PHP

- PHP is a server side language
- Once the page has loaded, PHP does nothing
- PHP is interpreted by the server
- You will want to run it from your engineering account by visiting the page
- Syntax is somewhat similar to C or Java



An example hello world

```
<html>
<head>
<title>PHP Test</title>
</head>
<body>
<?php echo '<p>Hello World</p>'; ?>
</body>
</html>
```

- This will print “Hello World” inside of <p> tags



PHP basics

- All php code must be inside of `<?php .. ?>`
- Put this as the very first lines on all of you php pages


```
<?php
    ini_set('display_errors', 'On');
?>
```
- This will give some error reporting, but wont help much for syntax errors



More php basics

- Pages containing PHP should end in .php
- .php pages can also contain HTML outside of the `<?php ?>` tag
- Often times servers will process PHP in .html files as well



Variables

- All variables need to start with a \$
- PHP variables are not of a fixed type
- Type is determined by context
- Many type conversions are automatic



Strings

- Strings are easy to use in PHP
- “.” is the concatenation operator (sticks two strings together)
- \$foo = “test string”;
- \$bar = “ and more”;
- \$baz = \$foo . \$bar; // “foo” . “bar” produces “foobar”
- echo \$baz; //prints “test string and more”



\$_POST and Associative Arrays

- You will be using the \$_POST variable a lot
- Stores values from forms
- Same as a dictionary or map
- Key is the name of the form field
- Value is, unsurprisingly, the value of the form field
- If we had <input name=“name”> and we put in “Bart” \$_POST[“name”] would contain “Bart” after posting the form





More Information on PHP

- <http://php.net/manual/en/>
