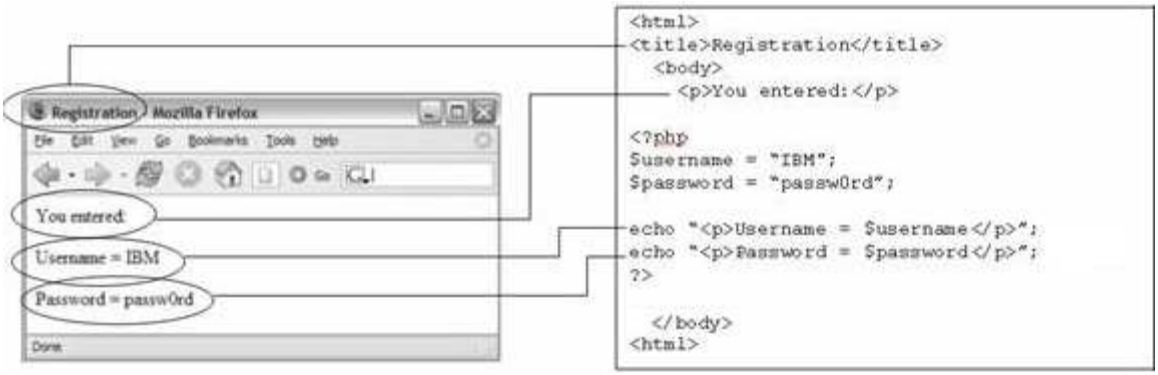
Embedded PHP

CSE 190 M (Web Programming), Spring 2008
University of Washington

Web services vs. embedded PHP

- the PHP programs we've written so far were **web services** (their output is plain text)
- most PHP programs actually produce HTML as their output
  - example: responses to HTML form submissions
- an **embedded PHP** program is a file that contains a mixture of HTML and PHP code



A bad way to produce HTML in PHP

```
<?php
print "<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.1//EN\"\n";
print " \"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd\">\n";
print "<html xmlns=\"http://www.w3.org/1999/xhtml\">\n";
print "  <head>\n";
print "    <title>My web page</title>\n";
...
?>
```

- printing HTML code with `print` statements is ugly and error-prone:
  - must quote the HTML and escape special characters, e.g. `\"`
  - must insert manual `\n` line breaks after each line
- don't `print` HTML; it's bad style!

Syntax for embedded PHP

```
html content

<?php
PHP code
?>

html content
```

- any contents of a `.php` file that are not between `<?php` and `?>` are output as pure HTML
- can switch back and forth between HTML and PHP "modes"

Embedded PHP example

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
 "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head><title>CSE 190 M: Embedded PHP</title></head>
  <body>
    <h1>Geneva's Counting Page</h1>
    <p>Watch how high I can count:
<?php
for ($i = 1; $i <= 10; $i++) {
  print "$i\n";
}
?>
    </p>
  </body>
</html>
```

PHP

- the above code would be saved into a file such as `count.php`
- How many lines of numbers will appear? (View Source!)

Embedded PHP as form response

```
...
  <body>
    <h1>New account created.</h1>
    <p>
<?php
$name = $_REQUEST["name"];
$email = $_REQUEST["email"];
...
print "\"Thank you\", $name, for creating an account with $email.\n";
?>
    </p>
  </body>
```

PHP

- users expect an HTML response page when they submit forms
- embedded PHP allows you run some server-side code and also send back an HTML response page

Embedded PHP + `print` = bad

```
...
  <body>
    <h1>Geneva's Counting Page</h1>
    <p>Watch how high I can count:
<?php
for ($i = 1; $i <= 10; $i++) {
  print "$i\n";
}
?>
    </p>
  </body>
```
PHP

- best PHP style is to use as few `print`/`echo` statements as possible in embedded PHP code
- but without `print`, how do we insert dynamic content into the page?

PHP expression blocks

```
<?= expression ?>
```
PHP

```
<h2>The answer is <?= 6 * 7 ?></h2>
```
PHP

The answer is 42

- **PHP expression block**: a small piece of PHP that evaluates and embeds an expression's value into HTML
  - `<?=` *expression* `?>` is equivalent to:

    ```
    <?php
    print expression;
    ?>
    ```

  - useful for embedding a small amount of PHP (a variable's or expression's value) in a large block of HTML without having to switch to "PHP-mode"

Expression block example 1

```php
<?php
$name = $_REQUEST["name"];
$email = $_REQUEST["emailaddress"];
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
 "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head><title>Account Creation</title></head>
  <body>
    <h1>New account created.</h1>
    <p>
      Thank you, "<?= $name ?>", for creating an
      account with <?= $email ?>.
    </p>
  </body>
</html>
```

- expression blocks get rid of `print` statement in previous example
- can often move PHP code up, out of the middle of the HTML

Expression block example 2

```php
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
 "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head><title>CSE 190 M: Embedded PHP</title></head>
  <body>
<?php
for ($i = 99; $i >= 1; $i--) {
?>
    <p>
      <?= $i ?> bottles of beer on the wall, <br />
      <?= $i ?> bottles of beer. <br />
      Take one down, pass it around, <br />
      <?= $i - 1 ?> bottles of beer on the wall.</p>
<?php
}
?>
  </body>
</html>
```

- this code could go into a file named `beer.php`

Common error: unclosed braces

```
...
  <body>
    <p>Watch how high I can count:
<?php
for ($i = 1; $i <= 10; $i++) {
?>
      <?= $i ?>
    </p>
  </body>
</html>
```
PHP

- if you open a `{` brace, you must have a matching `}` brace later (in the same PHP-mode block or a later one)
  - `</body>` and `</html>` above are inside the `for` loop, which is never closed
- if you forget to close your braces, you'll see an error about 'unexpected `$end`'

Common error fixed

```
...
  <body>
    <p>Watch how high I can count:
<?php
for ($i = 1; $i <= 10; $i++) {     # PHP mode
?>
      <?= $i ?>                 <!-- HTML mode -->
<?php
}                                  # PHP mode
?>
    </p>
  </body>
</html>
```
PHP

Common error: Missing = sign

```
...
  <body>
    <p>Watch how high I can count:
<?php
for ($i = 1; $i <= 10; $i++) {
?>
      <? $i ?>
<?php
}
?>
    </p>
  </body>
</html>
```
PHP

- a block between <? ... ?> is often interpreted as the same as one between <?php ... ?>
- PHP evaluates the code, but $i does not produce any output

Complex expression blocks

```
...
  <body>
<?php
for ($i = 1; $i <= 3; $i++) {
?>
    <h<?= $i ?>>This is a level <?= $i ?> heading.</h<?= $i ?>>
<?php
}
?>
  </body>
```
PHP

```
  <body>
    <h1>This is a level 1 heading.</h1>
    <h2>This is a level 2 heading.</h2>
    <h3>This is a level 3 heading.</h3>
  </body>
```
HTML

- expression blocks can even go inside HTML tags and attributes