Server-Side Scripting

• Server-side scripting
  – Scripts run on the server
  – Scripts return HTML to the client

• Apache
  – Open-source
  – Perl and PHP are well-supported

• Internet Information Services (IIS)
  – Microsoft
  – ASP is well-supported
Server-Side Scripting

(1) GET foo.php / HTTP 1.1

(2) Apache finds foo.php on disk

(3) Apache calls PHP to process foo.php

(4) PHP sends output back to Apache

(5) Apache sends PHP output as HTTP response to client
Web Scripting with PHP

• **Hello world script:** lect2/helloworld.php

```php
<?php
    echo "Hello world";
?>
```

• **Look at the HTML output**
  – Web browser, view source
PHP References

- [http://www.w3schools.com/php/default.asp](http://www.w3schools.com/php/default.asp)
PHP Variables

• Variables in PHP:
  – Start with a dollar sign:
    $a, $foo, $numParts
  – Have a data type:
    Integer, float, string, boolean, array, object
  – Are loosely (weakly) typed
    $ricky = 3;       // $ricky is an integer
    $lucy = 3.14;    // $lucy is a float
    $ethel = “friend”;  // $ethel is a string
Variable Scope

• Built-in superglobals are available everywhere
  `$GLOBALS,  $_GET,  $_POST,  $_COOKIE,  $_SESSION`

• Global vs local scope:

```php
<?php
$lucy = 5;  $ricky = 10;
$fred = $lucy+$ricky;

ethel();

function ethel () {
    $lucy=100;  $ricky=200;
    $fred = $lucy + $ricky;
    echo $fred;
    echo "\n";
}

echo $fred;
echo "\n";
?>
```

Note: run these programs from the command-line with:

```
php -qn foo.php
```

-q = quiet mode
-n = no php.ini
Variable Variables

• This is a little gross, but...

```php
<?php
$lucy = 5;
echo $lucy . "\n";
$ethel="lucy";
$$ethel=10;
echo $lucy . "\n";
?>
```

lect2/varvar.php

Note the $$
Constants

- Have scope everywhere (inside and outside fns)

```php
<?php

define('LUCY', 5);
define('ETHEL', 10);

echo LUCY . "\n";

function ricky() {
    echo ETHEL . "\n";
}

ricky();

?>
```
Operators

- Arithmetic operators
  - + - * / %
- String operator: concatenation
  - .
- Assignment operators
  - =
    - += -= *= /= %= .=
- Assignment returns a value
- Pre- and post-increment and decrement
  - ++ --

```php
<?php
    $a = 5;  $b = 10;
    $ricky = "Hello ";
    $lucy = "world";
    $c = $a * $b;
    $d = $b * ($f=$b);
    echo $c . "\n";
    echo $d . "\n";
    echo ++$d . "\n";
    $fred = $ricky . $lucy;
    echo $fred . "\n";
?>
```
More operators

• Comparison operators
  
  ==  !=  <>  <  >  <=  >=  ===  !==

  Caution: $a = $b evaluates to true for all values of $a and $b

• Logical operators
  
  &&  ||  !

• Bitwise operators
  
  &  |  ^  <<  >>
Strings

- Many string functions

```php
<?php

$a = 5;
$b = 10;

$ricky = "ricky";

echo strlen($ricky) . "\n";;
echo substr($ricky, 1, 4) . "\n";

?>
```
<?php
$fred=5;  $lucy=10;

if ($lucy>$fred) {
    echo "lucy is greater\n";
}

if ($lucy>$fred) {
    echo "lucy is greater or equal\n";
} else {
    echo "fred is greater\n";
}

if ($lucy>$fred) {
    echo "lucy is greater\n";
} elseif ($lucy==$fred) { // remember ==
    echo "both are equal\n";
} else {
    echo "fred is greater\n";
}

?>
<?php
$lucy=2;
$lucy='ethel';

switch ($lucy) {
    case 1:
        echo "lucy is 1\n";
        break;
    case 2:
        echo "lucy is 2\n";
        break;
    case 'ethel':
        echo "lucy is ethel\n";
        break;
    default:
        echo "default\n";
        break;
}
?>
While Loops

```php
<?php
$fred=1;
while ($fred<=10) {
    echo $fred . "\n";
    $fred++; // Increment
}
?>
```
For Loops

```php
<?php
    echo "Loop test<p>";
    for($i=0;$i<=9;$i++) {
        echo "i = $i<br>";
    }
?>
```

- **Iterator expression**: 
- **Starting expression**: 
- **End condition**: 

lect2/forloop.php
PHP Arrays

• Arrays in other languages:
  
  ```
  ```

• PHP arrays are more like pairings:
  
  ```
  force["luke"] = 3;
  force["ben"] = 5;
  force["yoda"] = 10;
  force["vader"] = -8;
  ```

• PHP arrays are mappings of *keys* to *values*. 
PHP Arrays

• Mapping of keys to values.
• Keys – can be integers or strings

lect2/array1.php

```php
$fred = array(2 => 3,
              3 => 5,
              4 => 7);
```

lect2/array2.php

```php
$fred[2] = 3;
$fred[3] = 5;
$fred[4] = 7;
```

These two methods generate the same array.
Check with:

```php
print_r($fred);
```
PHP Arrays

- Values – can be any data type, including another array!
- Can create multi-dimensional arrays.

```
$fred = array(2 => 3,
              "life" => 42,
              "inls760isfun" => true,
              14 => array( 1 => 3, 2 => 5));
print_r($fred);
```

lect2/array3.php
PHP Arrays

• When using arrays that have string keys...
• Always use quotes:

```php
$force["luke"] = 3;   // OKAY

$force['luke'] = 3;    // OKAY

$force[luke] = 3;      // WRONG!!!
```
<?php

$fred = array (
    "760" => "WebDB",
    "523" => "DB",
    "780" => "ResMeth",
    "582" => "SysAnal",
    "500" => "HII"
);

print_r($fred);

asort($fred);

print_r($fred);

ksort($fred);

print_r($fred);

?>
Foreach Loops

```php
<?php
    $fred[2] = 3;
    $fred[3] = 5;
    $fred[4] = 7;
    foreach($fred as $ethel) {
        print "$ethel\n";
    }
    foreach($fred as $lucy => $ethel) {
        print "$lucy = $ethel\n";
    }
?>
```
HTML Forms and PHP

- **HTML:** `lect2/form1.html`

```html
<head>
  <title>HTML Forms and PHP Test 1</title>
</head>
<body>
  <h1>HTML Forms and PHP Test 1</h1>
  <form method="post" action="form1.php">
    Name: &nbsp;&nbsp;&nbsp;&nbsp;<input type="text" name="name">
    <p>
    Course: &nbsp;<input type="text" name="course">
    <p>
    <input type="submit" value="Send">
  </form>
</body>
```
$_GET and $_POST

- PHP *superglobal* associative arrays
- Hold information sent from an HTML form
- PHP automagically places each form variable value into:
  
  ```
  $_GET[formvariablename]    or
  $_POST[formvariablename]
  ```
HTML Forms and PHP

- **PHP:** lect2/form1.php

```php
<?php
    echo "Hello, " . htmlspecialchars(strip_tags($_GET['name']));
    echo "<br>";
    echo "You are in the course: " . htmlspecialchars(strip_tags($_GET['course']));
?>
```
Dropping in and out of PHP

- **PHP:** lect2/form2.php

```php
<h1>Form 2 test</h1>
Hello,
<?php htmlspecialchars(strip_tags($_GET['name'])); ?>
<br>
You are in the course:
<?php htmlspecialchars(strip_tags($_GET['course'])); ?>
```

Two changes from form1.php:
1) GET instead of POST
2) PHP mixed with HTML
PHP Functions

- **PHP functions:** lect2/func1.php

```php
<?php
function print_course_title()
{
    echo "<h1>INLS 760 &ndash; Web Databases</h1>";
}

print_course_title();

?>
```
PHP Functions

• **PHP functions:** lect2/func2.php

```php
<?php
function print_course_title()
{
    ?>

    <h1>INLS 760 &ndash; Web Databases</h1>

    <?php
    }
    print_course_title();
    ?>
```
PHP Functions

- **PHP functions:** lect2/func3.php

```php
<?php
function print_square($num)
{
    echo $num*$num;
}
?>
```
Code reuse with require

- PHP require:  lect2/require1.php

```php
<?php
require "func3.php";
print_square(1); echo "<p>";
print_square(2); echo "<p>";
print_square(3); echo "<p>";
print_square(4); echo "<p>";
?>
```
Site design fun with functions

- **PHP site design:** lect2/sitefunc.php

```php
<?php
function page_template()
{
?>
<head>
    <title>INLS 760</title>
</head>
<body>
    <h1>INLS 760</h1>
    <?php page_content(); ?>
</body>

<?php
}
?>
```
Site design fun with functions

• PHP site design: lect2/sitepage1.php

```php
<?php
    require "sitefunc.php";

    function page_content()
    {
        ?>
        This is where the <b>content</b> for this page would go.
    }
<?php
?

<?php
    page_template();
?>```
Reading from a file

- lect2/readfile1.php

```php
<?php
    $lines = file("data1.txt");
    foreach ($lines as $line)
    {
        $fields = explode(",",$line);
        foreach ($fields as $field)
        {
            echo "$field"
            echo "\n";
        }
    }
?>
```
Reading from a file

- **PHP site design:** lect2/readfile1.php

```php
<h1>Reading a file test 1</h1>
<table>
  <?php
    $lines = file("data1.txt");
    foreach ($lines as $line) {
      echo "<tr>";
      $fields = explode(",",$line);
      foreach ($fields as $field) {
        echo "<td>";
        echo "$field";
        echo "</td>";
      }
      echo "</tr>";
    }
  ?>
</table>
```

data1.txt:
- luke,skywalker,3
- ben,kenobe,5
- yoda,yoda,10
- darth,vader,-8