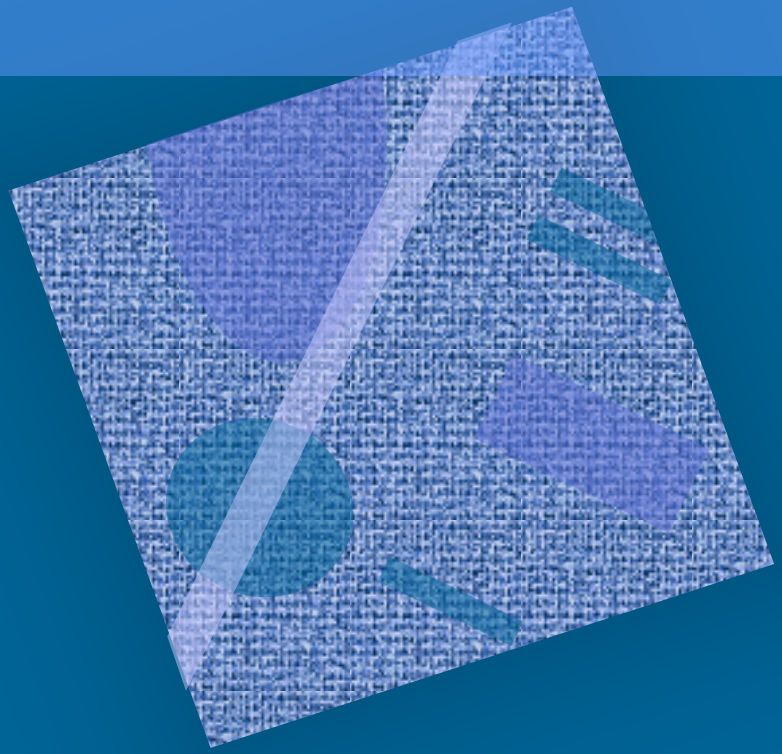
COMS 469: Interactive Media II



Agenda

- Project One
- PHP
- Preview

Project One

- Grading Methodology
- Return Project One & Evaluation Sheet

Project One

- Evaluation Methodology
  - Consider each project in and of itself
    - Look at how each project implements JavaScript and incorporates its features into its design.
    - Not a comparison of projects to each other or to some fictional project in my head
  - Motivation for the evaluation
    - Point out things that work and things that do not work in order to help you design a better web page
    - Not design hazing

COMS 469
Project #1 **Name:**

Content & Design (20 points) –
[ ] The project is turned in on time and in the proper format. (5)
[ ] Site correctly employs standard HTML code and is free from si
[ ] Text information is free from errors in spelling, grammar, and p
[ ] Layout and color combinations are appropriate for the content c
and accessible presentation of content, and provide the site with a p

DHTML (20 points) –
[ ] The site employs JavaScript and CSS to alter page element appe
rollovers, positioning and moving page elements, or other forms of
[ ] The JavaScript and CSS is written properly and is without error
[ ] The DHTML features function properly, do not exhibit errors,
results. (5)
[ ] The DHTML features provide for appropriate forms of user inte
overall site design, and complement the site's general appearance.

Windows & Frames (20 points) –
[ ] The site employs JavaScript to control the opening, closing, lay
windows and/or frames. (5)
[ ] The JavaScript is written properly and is without errors in synta
[ ] The window and/or frame features function properly, do not ex
or unstable results. (5)
[ ] The window and/or frame options are appropriate for the site's
useful feedback and navigational information. (5)

Forms & Cookies (20 points) –
[ ] The site employs JavaScript to validate or process form entry da
user's system. (5)
[ ] The JavaScript is written properly and is without errors in synta
[ ] The form and/or cookies application functions properly and doe
inconsistent results. (5)
[ ] The validation alerts and/or cookie-generated content provide th
feedback, and information. (5)

Dynamic Content (20 points) –
[ ] The site employs JavaScript to generate page content (i.e. custo
information, user customization, or other forms of dynamically gen
[ ] The JavaScript is written properly and is without errors in synta
[ ] The dynamically generated data function properly, do not exhib
unstable results. (5)
[ ] The incorporation of dynamic content is motivated by the proje
fits appropriately into the site's structure and approach. (5)

Total –

Rev. 10 February 2009

The evaluation of your project was guided by the evaluation form, which has been available on the course web site.

The form is organized into five categories with 4 specific goals in each category. Each goal is worth 5 points; each category is worth 20 points.

If you accomplished a stated goal, you received the 5 points. If there were errors or problems, you lost 5 points.

If you forgot to include something altogether, the most you could lose was 10 points.

Points for each category are recorded on the evaluation sheet and total points are indicated in the lower left-hand corner.

Project One

- Grading - grade is a function of 100 points

  90-100 = A

  80-89 = B

  70-79 = C

  60-69 = D

  0-59 = F

Project One

- Grade Distribution
  - A = 6
  - B = 6
  - C = 1
  - D = 4
- High Score = 100

Project One

- Return projects and evaluation sheet
- Questions
  - General = answer in class
  - Specific = during break or office hours
- Next Step
  - Fix errors now; don't wait
  - Learn from your mistakes and have the best version available for your portfolio
  - "Special Offer" for D's (65 and below)

PHP

- Brief History of PHP
- PHP Applications
- Writing PHP
- Using Variables and Operators

PHP History

- PHP was invented by Rasmus Lerdorf in 1995 to extend HTML
- Originally called PHP/FI = *Personal Home Page/Forms Interpreter*
- Open Source
  - PHP/FI 2.0 was released by Lerdorf in November of 1997 as Open Source
  - PHP is currently distributed and developed as open source software

PHP History

- PHP 3.0
  - Released in June 1998
  - Written by Andi Gutmans and Zeev Suraski, who found PHP/FI 2.0 severely under-powered to drive e-commerce applications
  - Called PHP = *PHP Hypertext Pre-processor*

PHP History

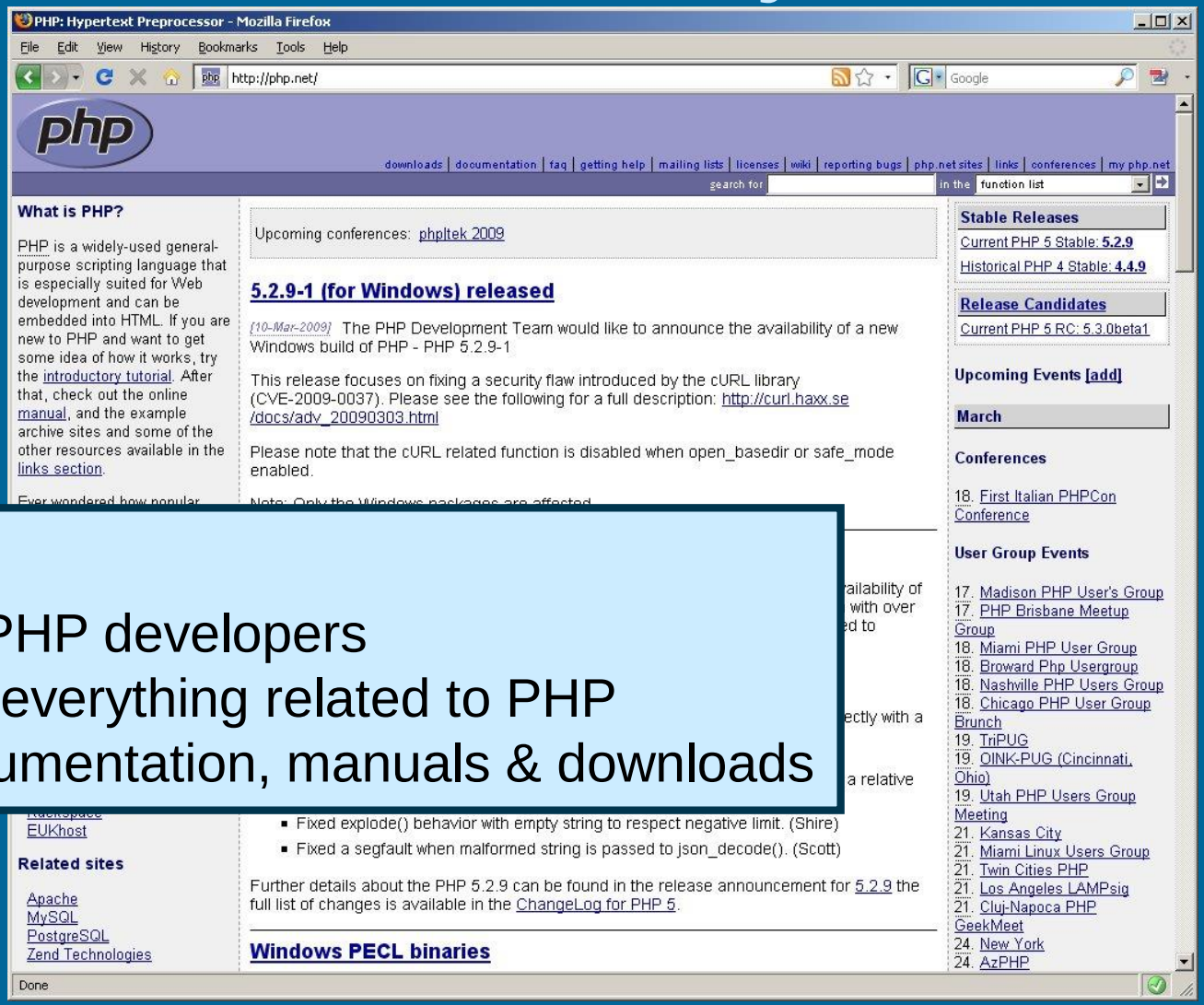
- PHP 4.0 (May 2000)
  - Gutmans and Suraski rewrite the PHP core in order to handle third party databases and APIs
  - PHP 4 is based on the **Zend Engine** (combination of the names Zeev and Andi)
  - Works with MySQL and other databases
  - Support for sessions and object oriented features

History

- PHP 5.0 (Sept. 2004)
  - Radical redesign of PHP 4.0
  - New Zend II Engine
  - New Features
    - New Object Model
    - Better Object Oriented syntax
    - XML and SOAP integration

NOTE:
- PHP 5.5.10 is the most recent release.
- Considered to be the most stable and robust version of the PHP language.

PHP History



php.net

- Web site for PHP developers
- Resource for everything related to PHP
- Provides documentation, manuals & downloads

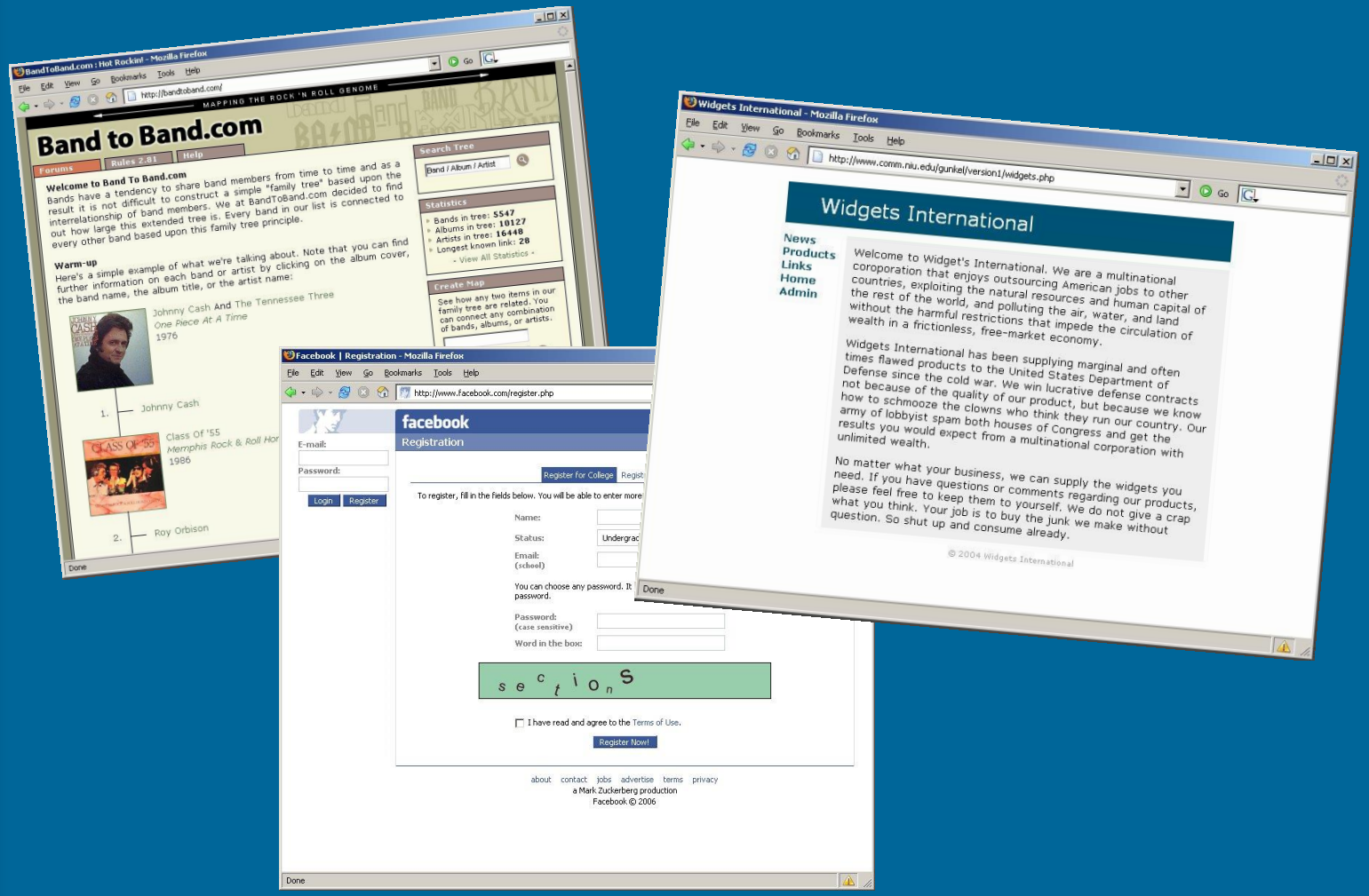
PHP Applications

- PHP programs perform three basic kinds of operations
  - Obtain data from users
  - Access and manipulate data stored in files and databases
  - Display data so that users can view it

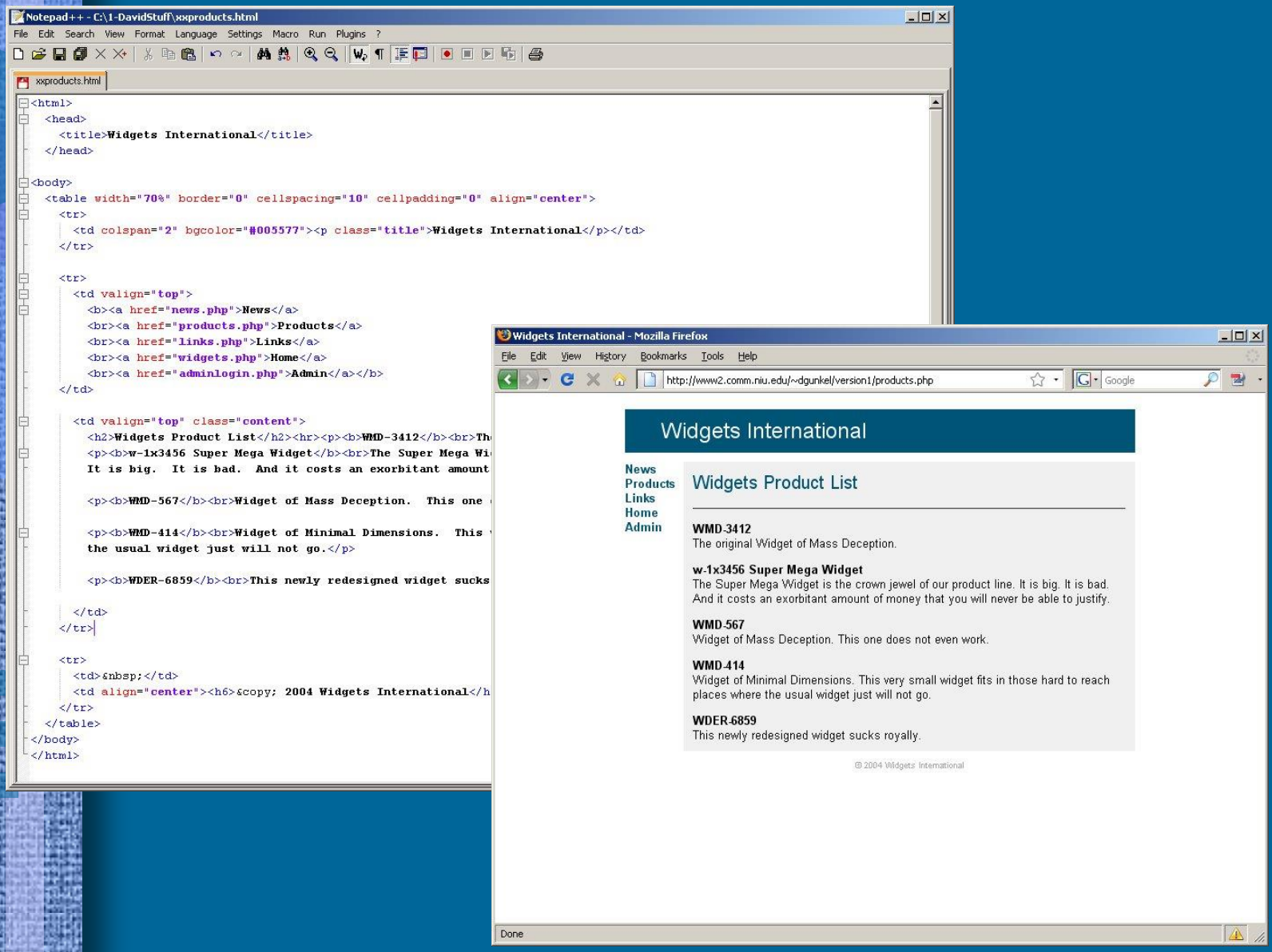
PHP Applications

- PHP, what is it good for? Absolutely everything.
  - Process form data
  - Present dynamic content
  - Communicate with databases
  - Content management
  - Password protection & User Registration
  - E-commerce: Shopping carts and Inventory

PHP Applications



Static HTML


Notepad++ - C:\1-DavidStuff\xxproducts.html
File Edit Search View Format Language Settings Macro Run Plugins ?
xxproducts.html
<html>
<head>
<title>Widgets International</title>
</head>
<body>
<table width="70%" border="0" cellspacing="10" cellpadding="0" align="center">
<tr>
<td colspan="2" bgcolor="#005577"> Widgets International </td>
</tr>
<tr>
<td valign="top">
News

Products

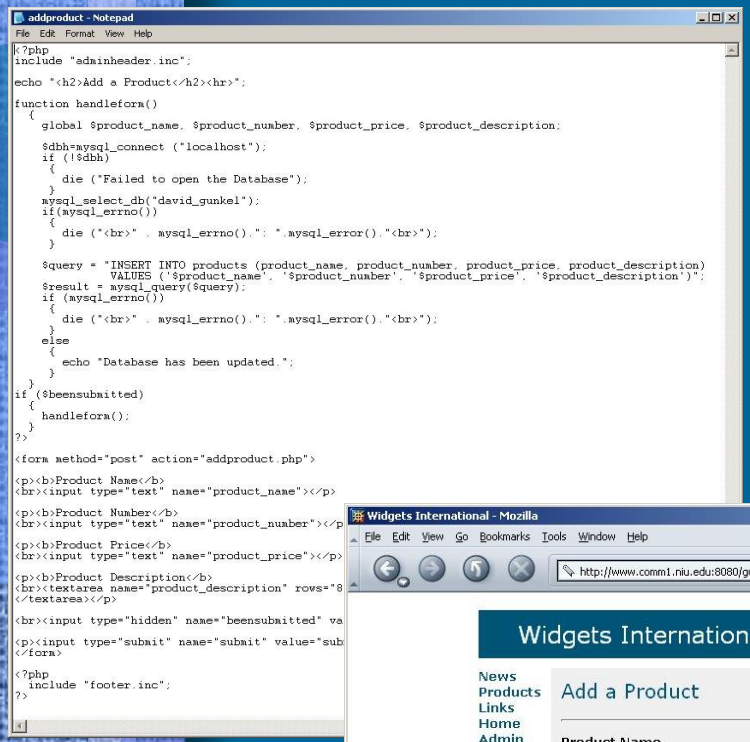
Links

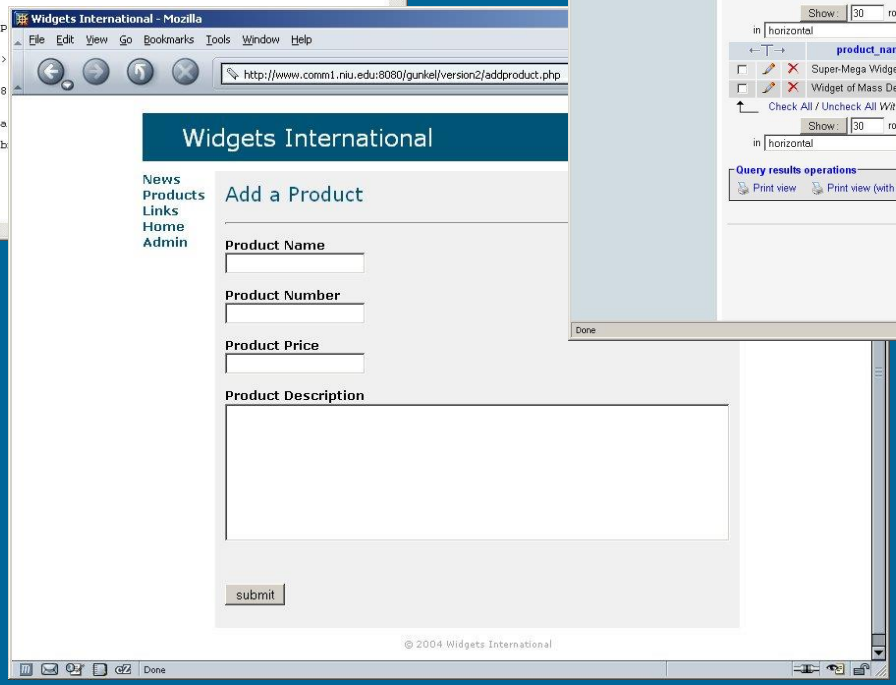
Home

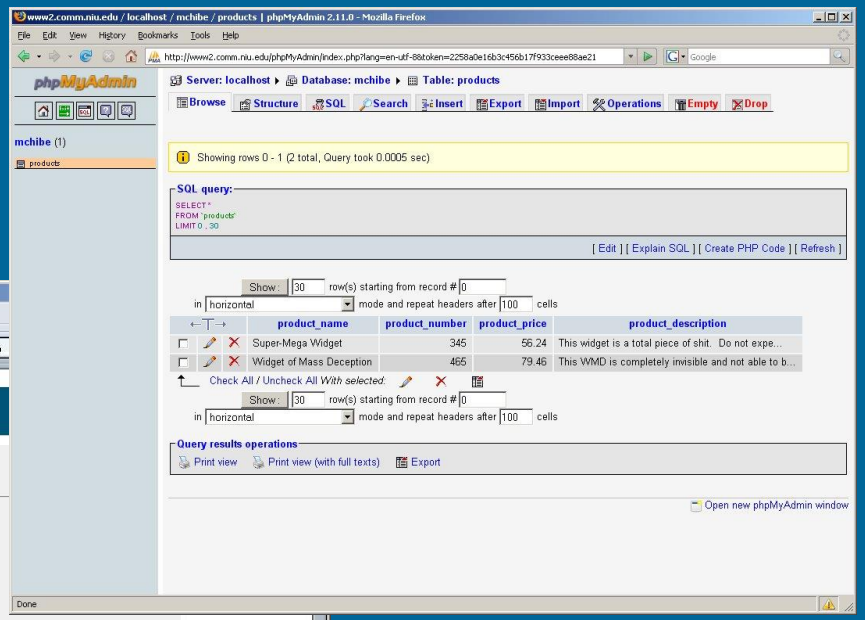
Admin
</td>
<td valign="top" class="content">
<h2>Widgets Product List</h2><hr> WMD-3412
Th
w-1x3456 Super Mega Widget
The Super Mega Wi
It is big. It is bad. And it costs an exorbitant amount
WMD-567
Widget of Mass Deception. This one
WMD-414
Widget of Minimal Dimensions. This
the usual widget just will not go.
WDER-6859
This newly redesigned widget sucks
</td>
</tr>
<tr>
<td> </td>
<td align="center"><h6>© 2004 Widgets International</h
</tr>
</table>
</body>
</html>
Widgets International - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www2.comm.niu.edu/~dgunkel/version1/products.php
Google
Widgets International
News
Products
Links
Home
Admin
Widgets Product List
WMD-3412
The original Widget of Mass Deception.
w-1x3456 Super Mega Widget
The Super Mega Widget is the crown jewel of our product line. It is big. It is bad. And it costs an exorbitant amount of money that you will never be able to justify.
WMD-567
Widget of Mass Deception. This one does not even work.
WMD-414
Widget of Minimal Dimensions. This very small widget fits in those hard to reach places where the usual widget just will not go.
WDER-6859
This newly redesigned widget sucks royally.
© 2004 Widgets International
Done

Administrator Side

Database driven dynamic content

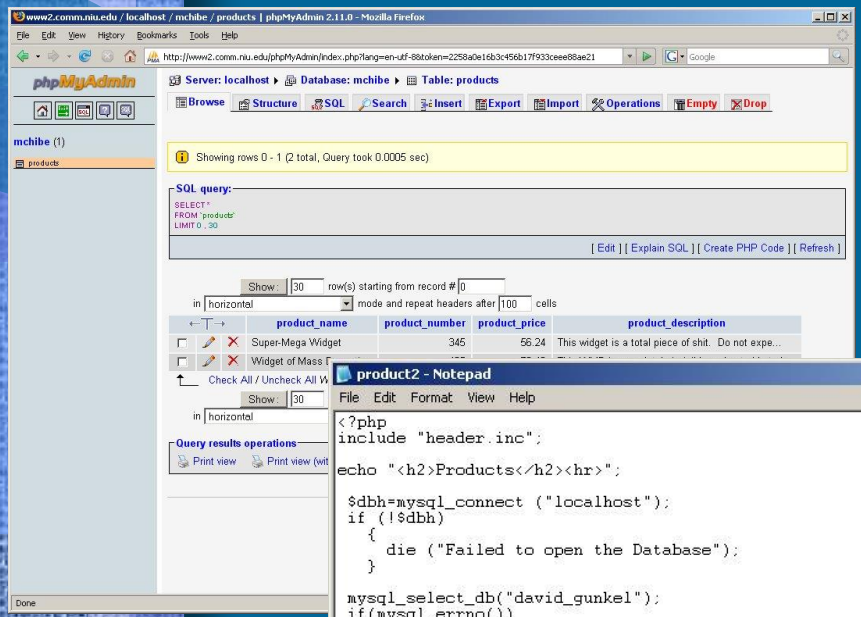


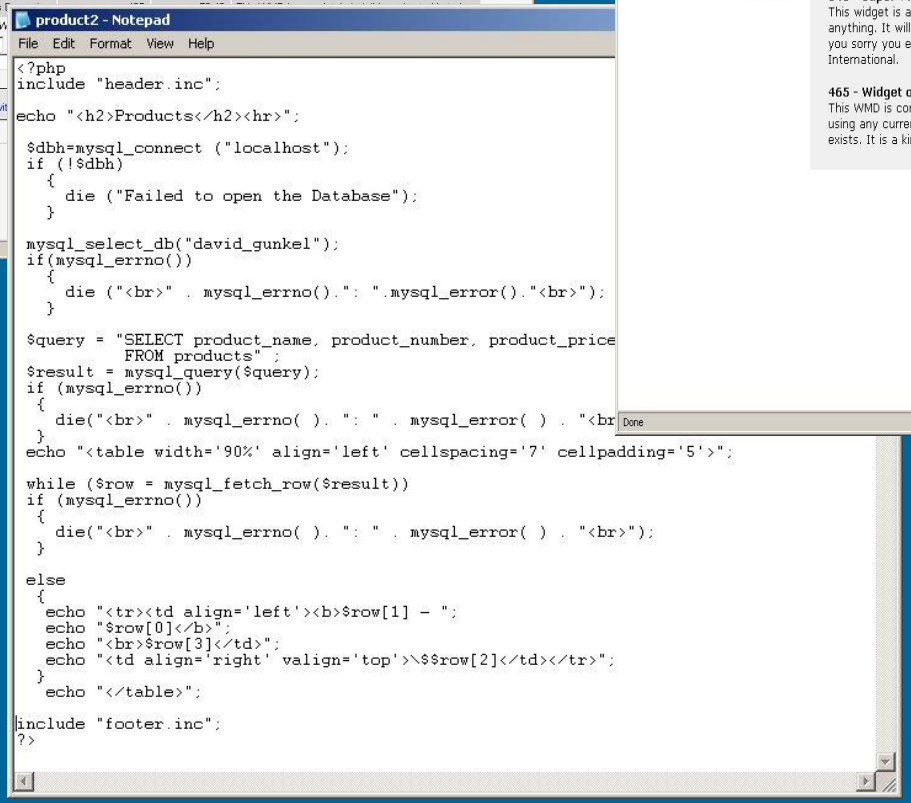


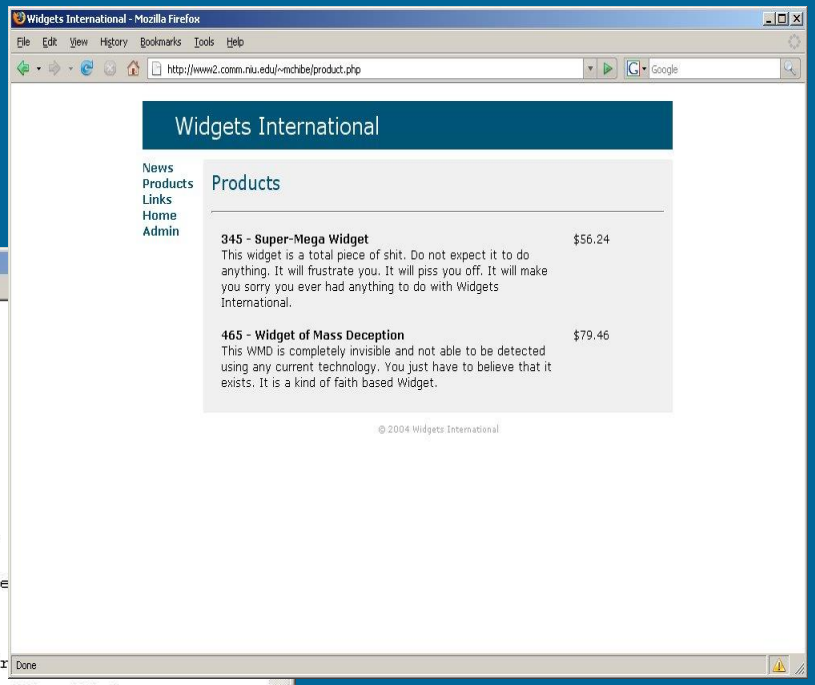


Database driven dynamic content

User Side







Content Management Site

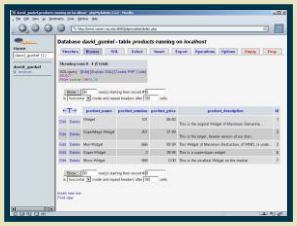

mainpage


admin login


authentication


header


news

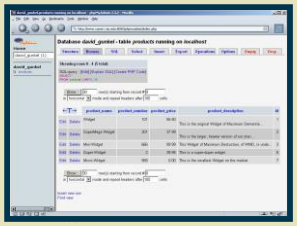



add news


admin header


footer

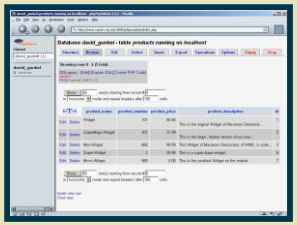

products




add product


admin footer


links




add link

PHP Applications

- PHP vs. other server side applications

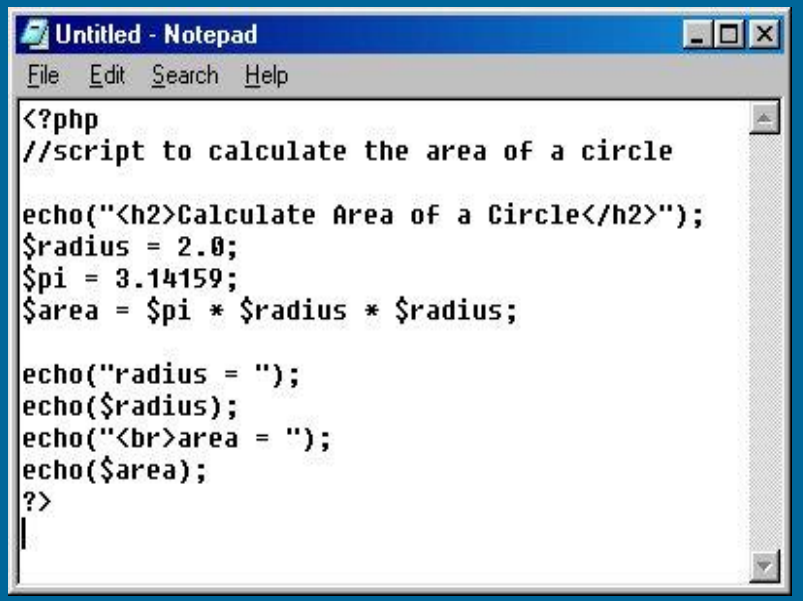
	Learning curve	Language Type	Server Requirements	Development Method
PHP	easy	Scripting (not compiled)	Any Server	Open Source
ASP	easy+	Scripting	Micro$oft	Proprietary (Micro$oft)
			ver	Open Source
			ver but Java nent	Proprietary (Sun)
		(compiled)	ver	International Standard ISO

Bottom Line

PHP is better, faster, and easier to learn than the alternatives. It provides excellent performance and interoperability across different platforms, tight integration with almost every database available, and a stable and secure web development environment. And it costs you nothing.

Writing PHP

- PHP is written in a text editor
  - Like HTML & JavaScript
  - Use text editor and not word processor
  - Saved file with .php extension – *file.php*



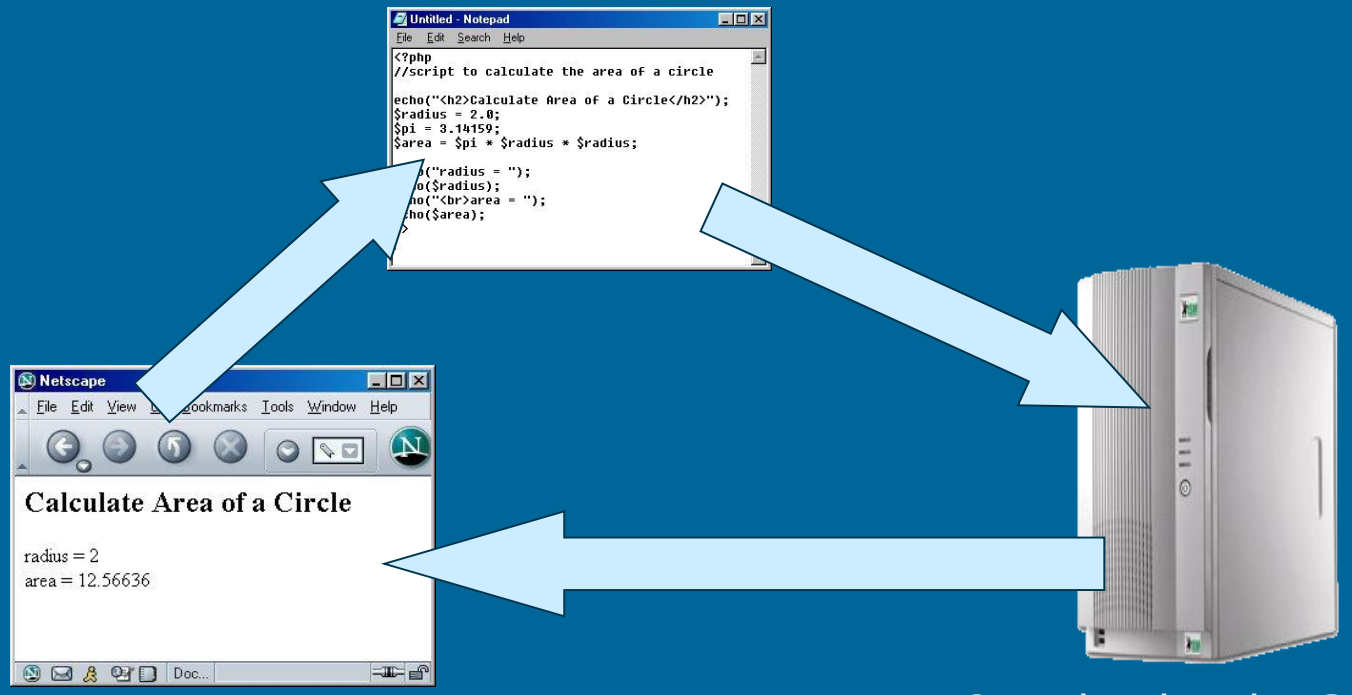
Writing PHP

- PHP runs on the server
  - Unlike HTML or JavaScript, PHP does not get interpreted on the client side in the browser environment
  - PHP is interpreted on the server using a pre-installed PHP environment. The results of the script are then sent to and displayed in the user's browser.

Writing PHP

- PHP Development cycle

1. Write PHP script in NotePad or NotePad++



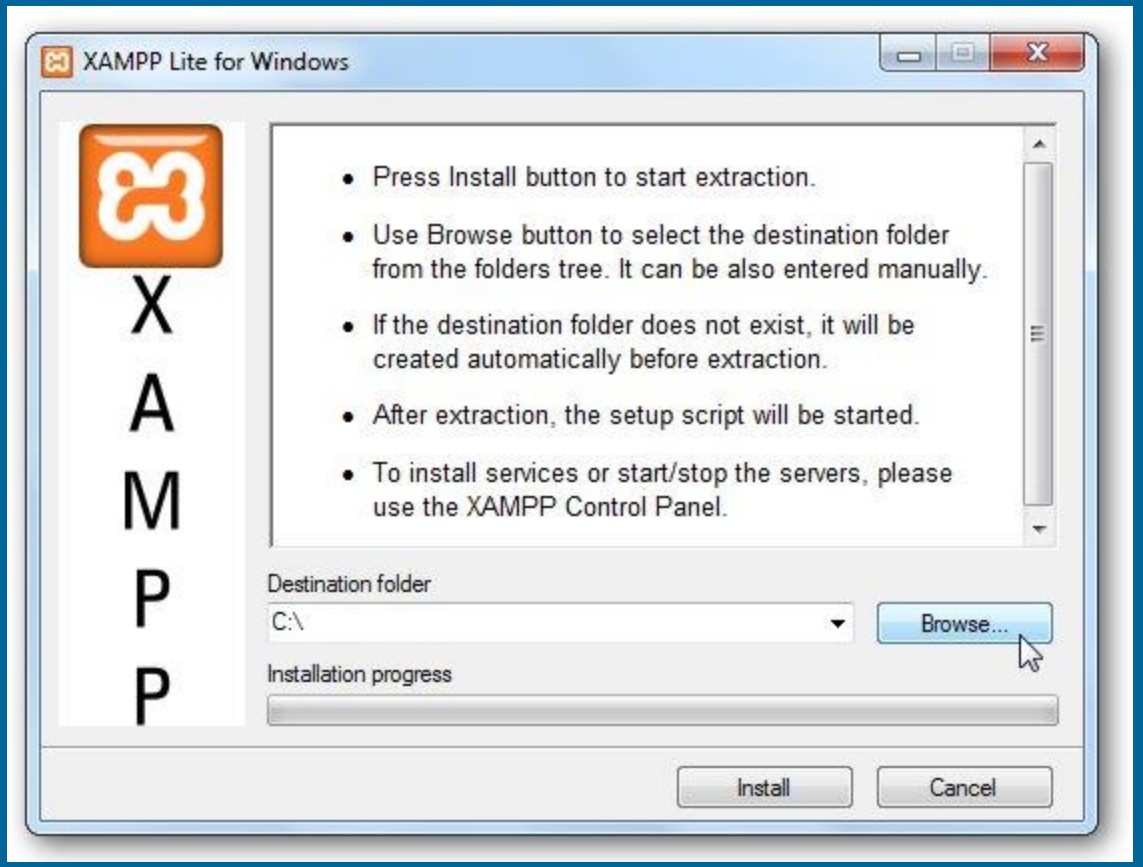
2. Upload to the Server

3. View file in browser

Writing PHP

- What this means for you?
  - Cannot develop PHP the same way you develop HTML and JavaScript
  - Need to have access to a server that supports PHP and MySQL
    - Local – Install server software (XAMPP Lite) on your local drive and run PHP in this environment
    - Server/Web Host
      - Set-up and administer your own server
      - Use a Web Host that supports PHP

Writing PHP



http://download.cnet.com/XAMPP-Lite/3000-10248_4-75157363.html

http://www.howtogeek.com/howto/14998/

Writing PHP

- Web Hosts
  - NIU Host
    - No cost; large file capacity
    - Supports PHP 5 & MySQL
  - Commercial Web Host
    - $40 for 3 months; large file capacity
    - Supports PHP & MySQL
    - Added bonus – create & use your own domain name

Writing PHP

- NIU Web Host
  - Utilizes COMS dept. server
  - Stipulations
    - Only good for the time that you are enrolled in the course
    - Cannot make commercial transactions on an NIU server
    - Does not permit full control over all server functions
    - No domain name registration

Writing PHP

- Commercial Web Host

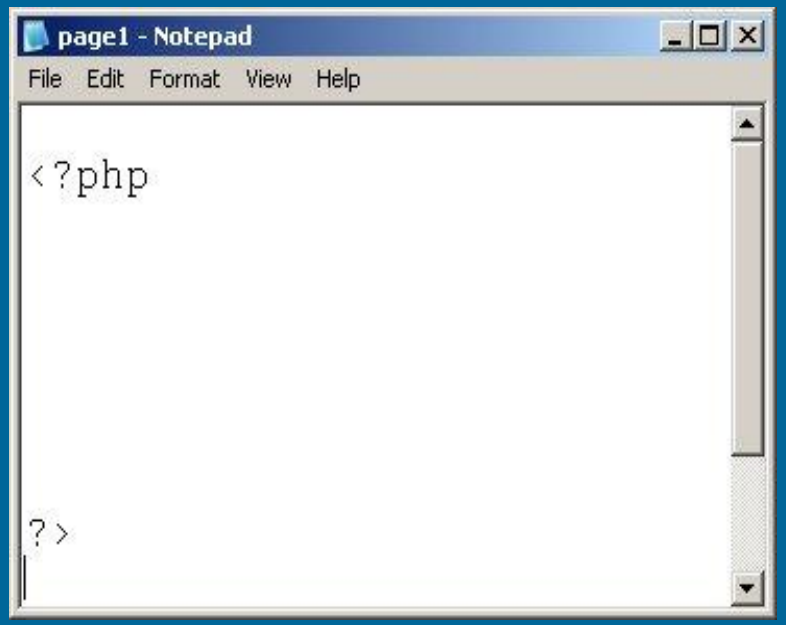
ipower.com

- Starter package = $3.95/month
- Unlimited Space and Bandwidth
- Quick set up and good support
- Domain name registration
- PHP, MySQL, Perl, SSL, etc.

Writing PHP

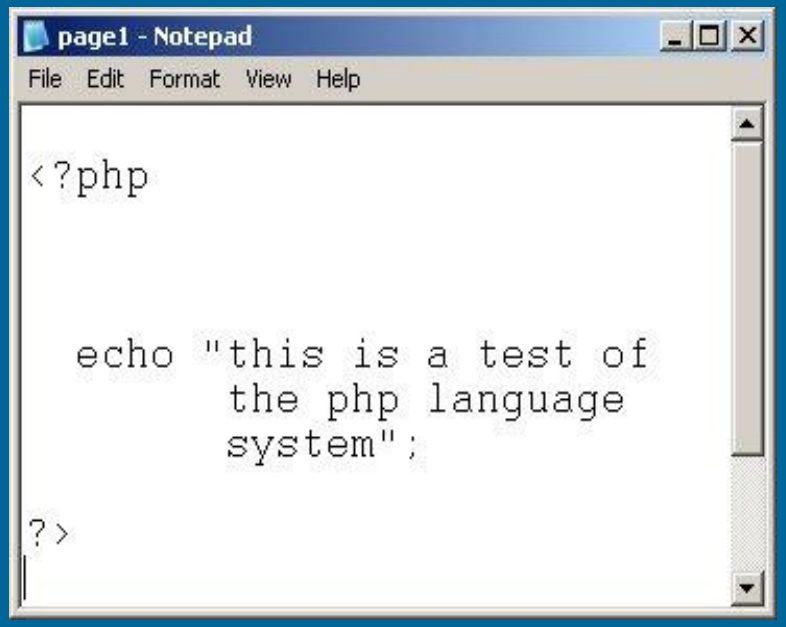
- PHP document
  - Basic PHP container tags

    <?php … ?>
  - Tell the server to treat anything in between the tags as PHP commands
  - Interpret code as PHP commands rather than sending it as is to the browser as HTML



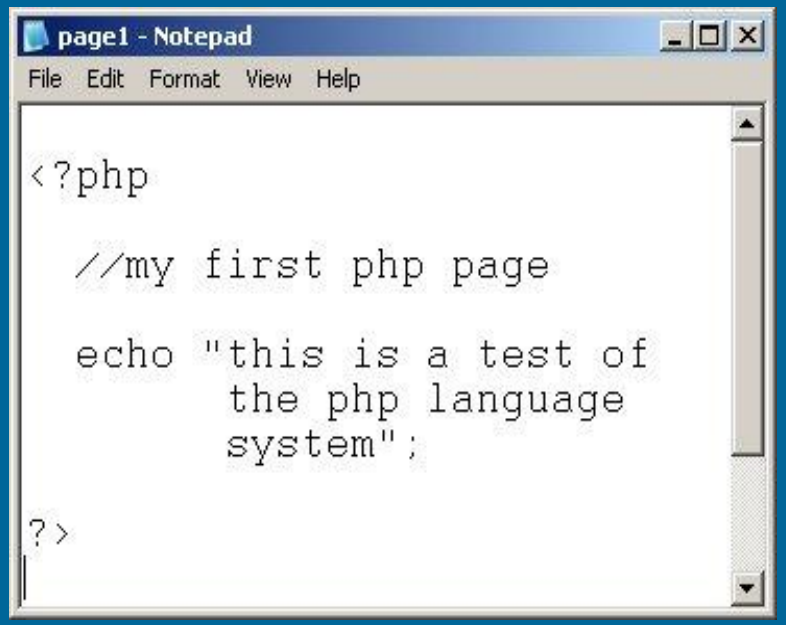
Writing PHP

- First PHP Page
  - Write output to the browser
  - Use echo command
  - Syntax
    - Place the echo command within the <?php container
    - Put a text string between quotation marks
    - End line with semi-colon

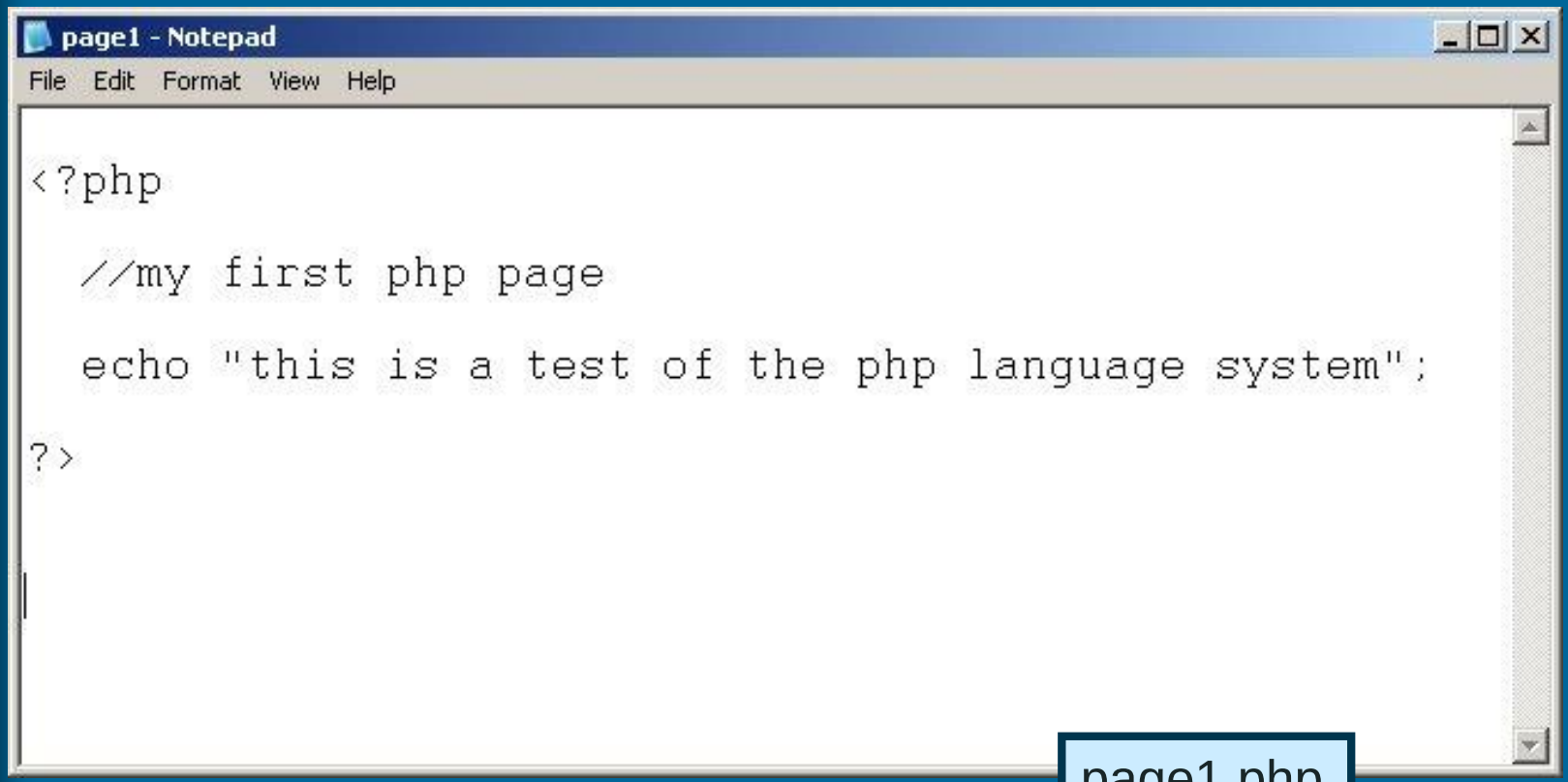


Writing PHP

- Documenting PHP
  - PHP comments are proceeded by //
  - Use comments to document script and to “comment-out” lines of code for debugging
  - Multi-line comments can be included by using the comment containers /*…. */

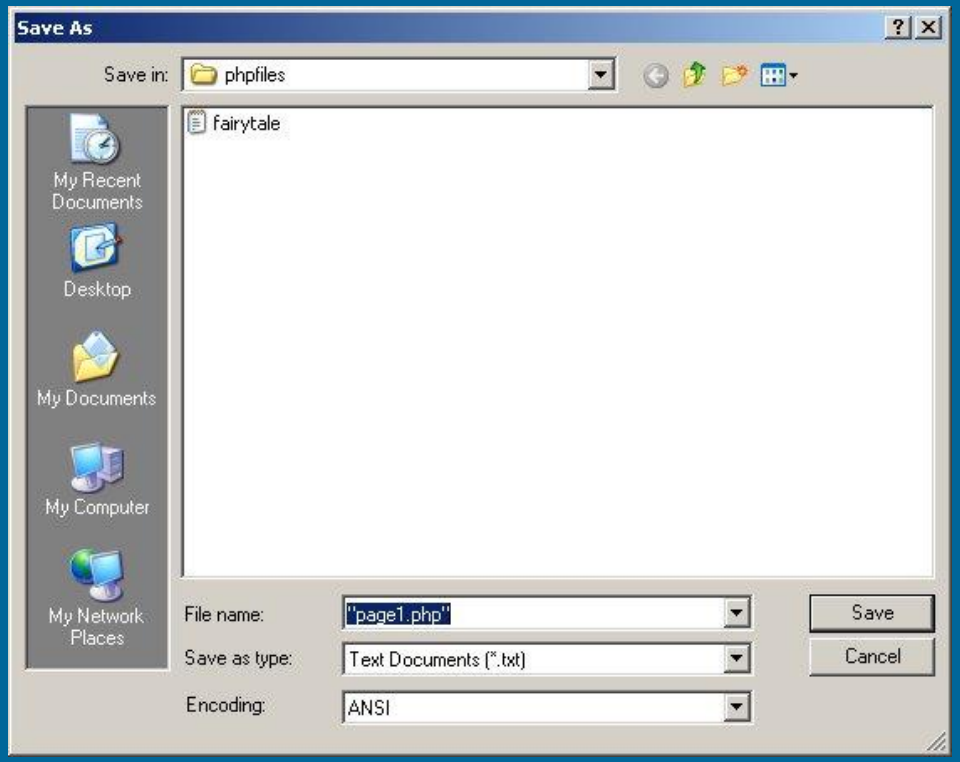


Writing PHP


page1 - Notepad
File Edit Format View Help
<?php
//my first php page
echo "this is a test of the php language system";
?>
page1.php

Writing PHP

- Save PHP file
  - Select Save-as
  - *page1.php*
  - Save with .php extension



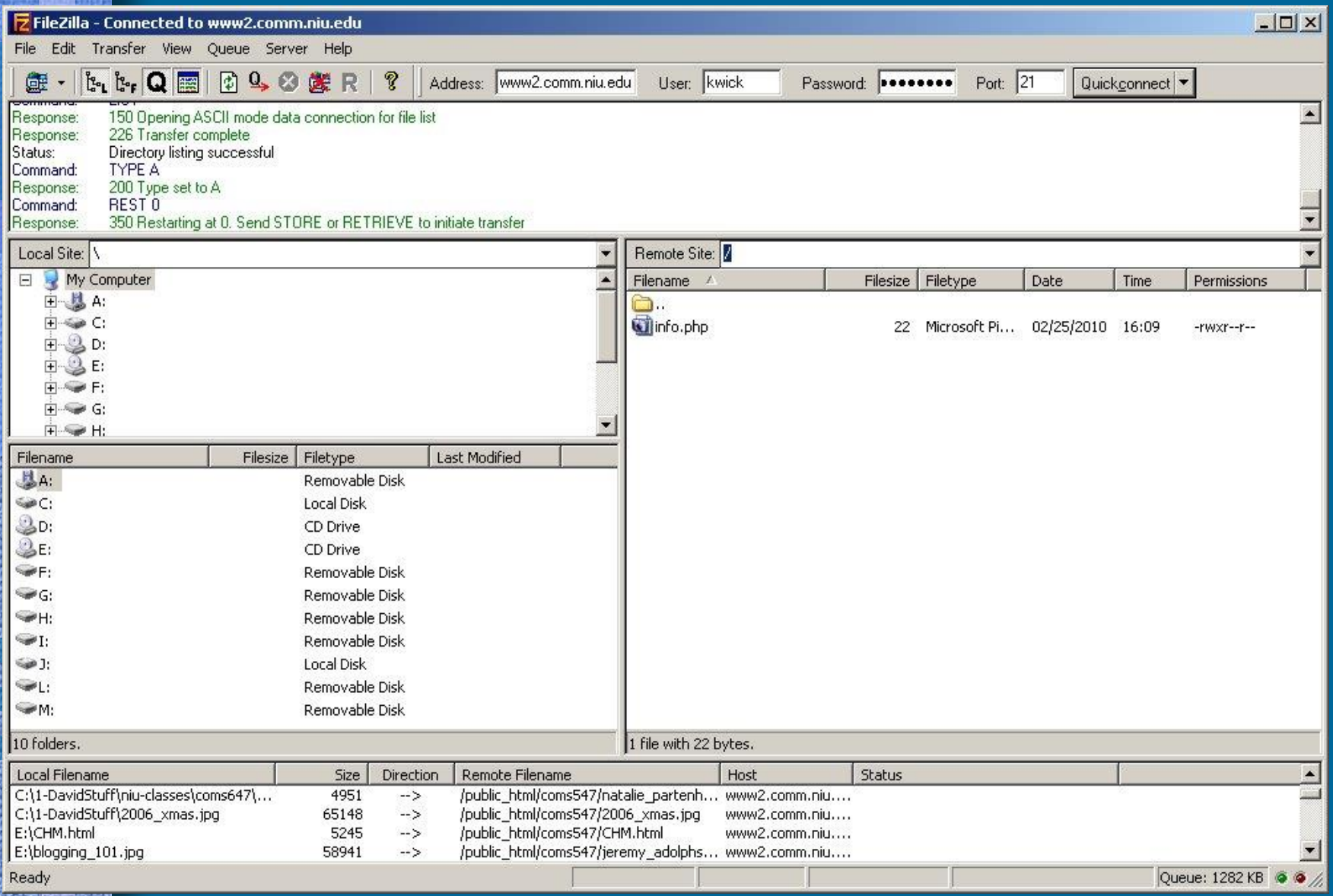
Writing PHP

- Upload the PHP file to the web server
  - FTP software

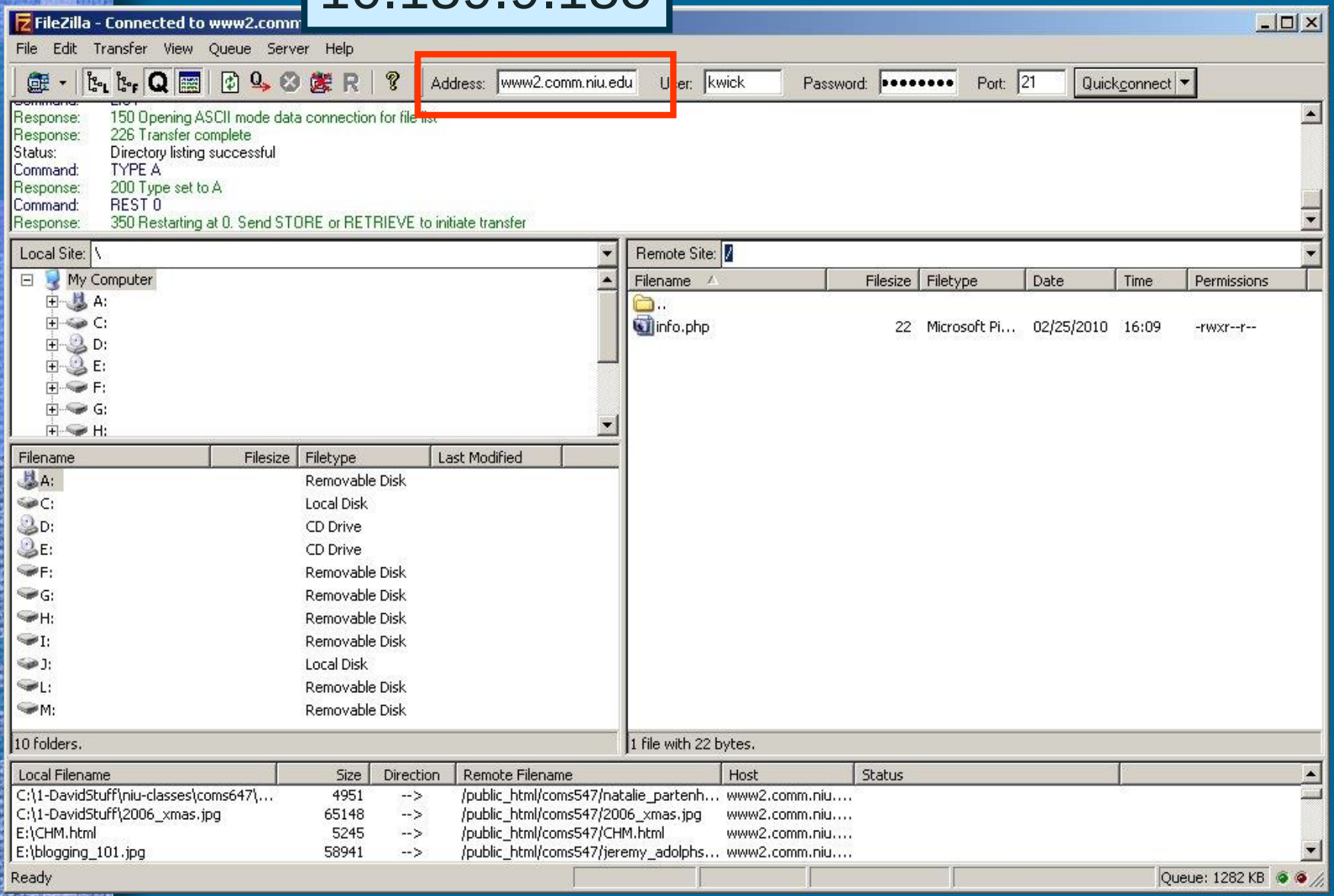
  - Login Instructions

Host	10.159.9.155
Username	1st initial + last name = jwinston
Password	8 digit zID = 01625661

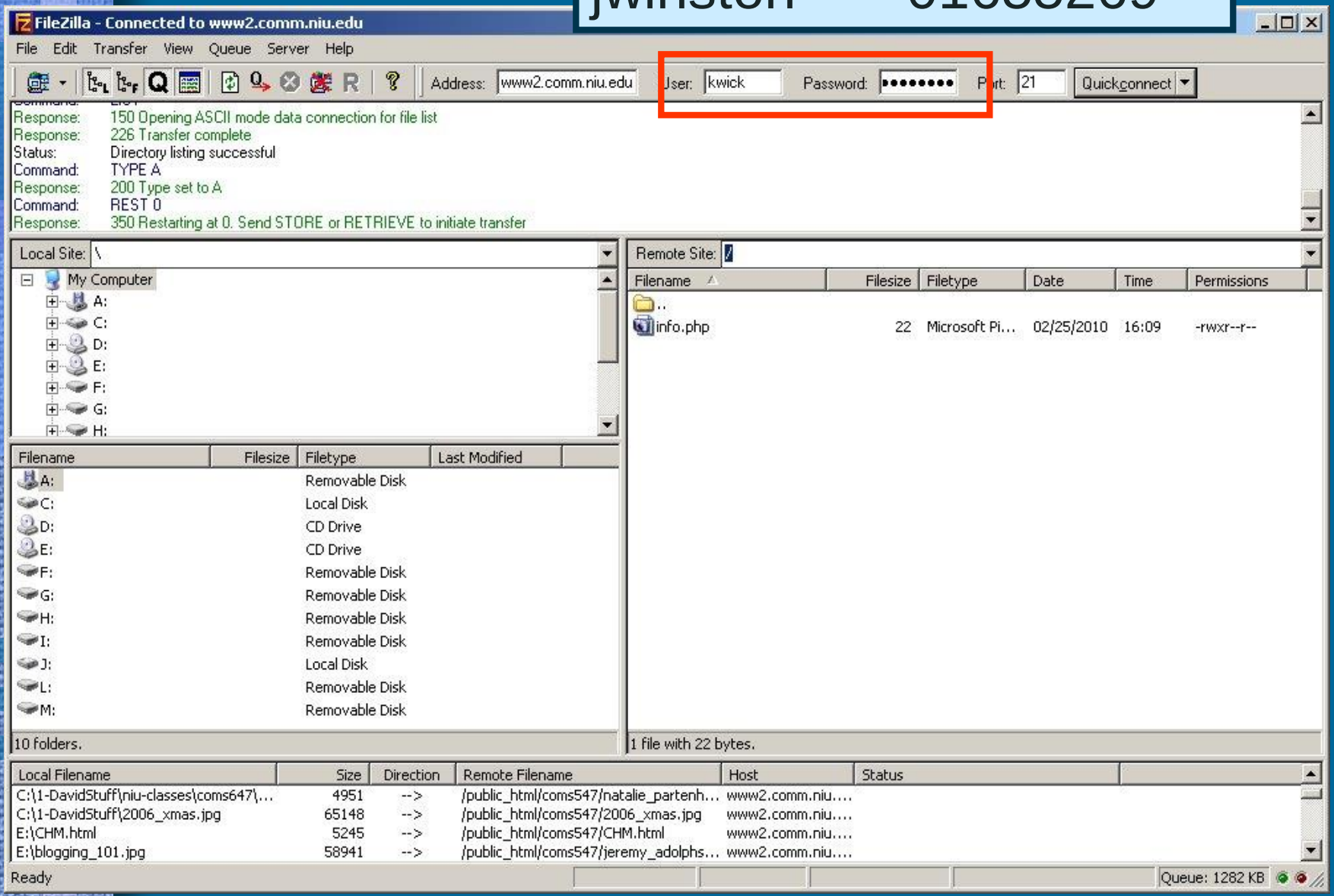
  - **NOTE:** Off campus login requires VPN
    - https://secure.niu.edu


FileZilla - Connected to www2.comm.niu.edu
File Edit Transfer View Queue Server Help
Address: www2.comm.niu.edu User: kwick Password: •••••••• Port: 21 Quickconnect
Response: 150 Opening ASCII mode data connection for file list
Response: 226 Transfer complete
Status: Directory listing successful
Command: TYPE A
Response: 200 Type set to A
Command: REST 0
Response: 350 Restarting at 0. Send STORE or RETRIEVE to initiate transfer
Local Site: \
My Computer
A:
C:
D:
E:
F:
G:
H:
Filename Filesize Filetype Last Modified
A: Removable Disk
C: Local Disk
D: CD Drive
E: CD Drive
F: Removable Disk
G: Removable Disk
H: Removable Disk
I: Removable Disk
J: Local Disk
L: Removable Disk
M: Removable Disk
10 folders.
Remote Site: /
Filename Filesize Filetype Date Time Permissions
..
info.php 22 Microsoft Pi... 02/25/2010 16:09 -rwxr--r--
1 file with 22 bytes.
Local Filename Size Direction Remote Filename Host Status
C:\1-DavidStuff\niu-classes\coms647\... 4951 --> /public_html/coms547/natalie_partenh... www2.comm.niu....
C:\1-DavidStuff\2006_xmas.jpg 65148 --> /public_html/coms547/2006_xmas.jpg www2.comm.niu....
E:\CHM.html 5245 --> /public_html/coms547/CHM.html www2.comm.niu....
E:\blogging_101.jpg 58941 --> /public_html/coms547/jeremy_adolphs... www2.comm.niu....
Ready
Queue: 1282 KB

10.159.9.155

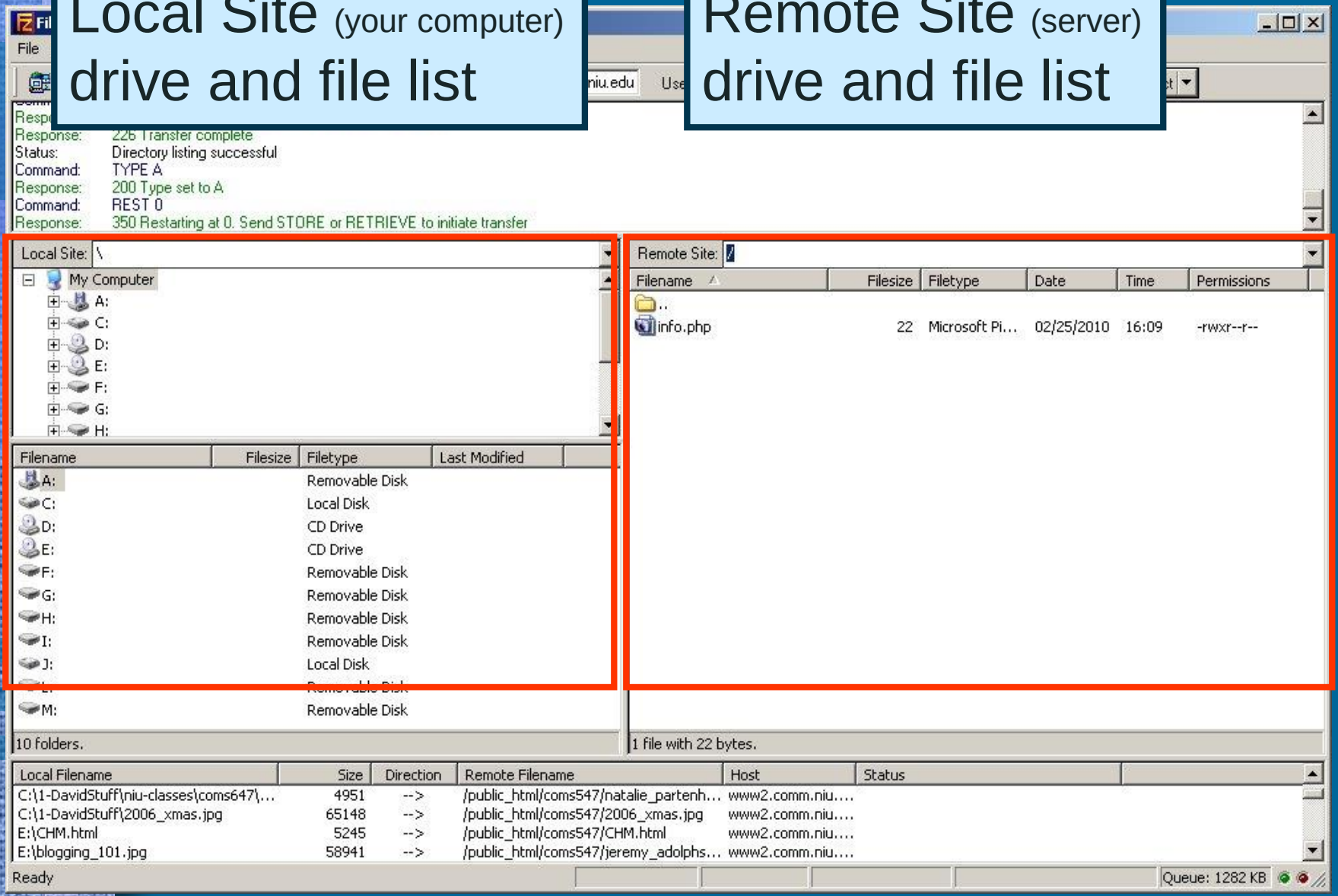

FileZilla - Connected to www2.comm
File Edit Transfer View Queue Server Help
Address: www2.comm.niu.edu User: kwick Password: •••••••• Port: 21 Quickconnect
Response: 150 Opening ASCII mode data connection for file list
Response: 226 Transfer complete
Status: Directory listing successful
Command: TYPE A
Response: 200 Type set to A
Command: REST 0
Response: 350 Restarting at 0. Send STORE or RETRIEVE to initiate transfer
Local Site: \
My Computer
A:
C:
D:
E:
F:
G:
H:
Filename Filesize Filetype Last Modified
A: Removable Disk
C: Local Disk
D: CD Drive
E: CD Drive
F: Removable Disk
G: Removable Disk
H: Removable Disk
I: Removable Disk
J: Local Disk
L: Removable Disk
M: Removable Disk
10 folders.
Remote Site: /
Filename Filesize Filetype Date Time Permissions
..
info.php 22 Microsoft Pi... 02/25/2010 16:09 -rwxr--r--
1 file with 22 bytes.
Local Filename Size Direction Remote Filename Host Status
C:\1-DavidStuff\niu-classes\coms647\... 4951 --> /public_html/coms547/natalie_partenh... www2.comm.niu....
C:\1-DavidStuff\2006_xmas.jpg 65148 --> /public_html/coms547/2006_xmas.jpg www2.comm.niu....
E:\CHM.html 5245 --> /public_html/coms547/CHM.html www2.comm.niu....
E:\blogging_101.jpg 58941 --> /public_html/coms547/jeremy_adolphs... www2.comm.niu....
Ready
Queue: 1282 KB

jwinston 01655269



Local Site (your computer) drive and file list

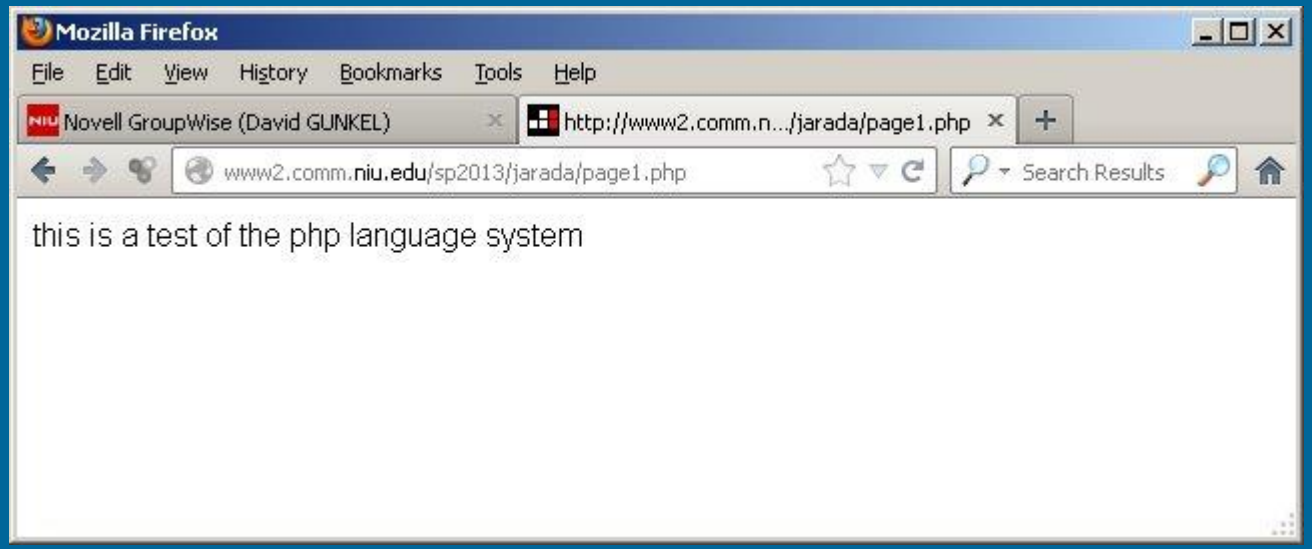
Remote Site (server) drive and file list



Writing PHP

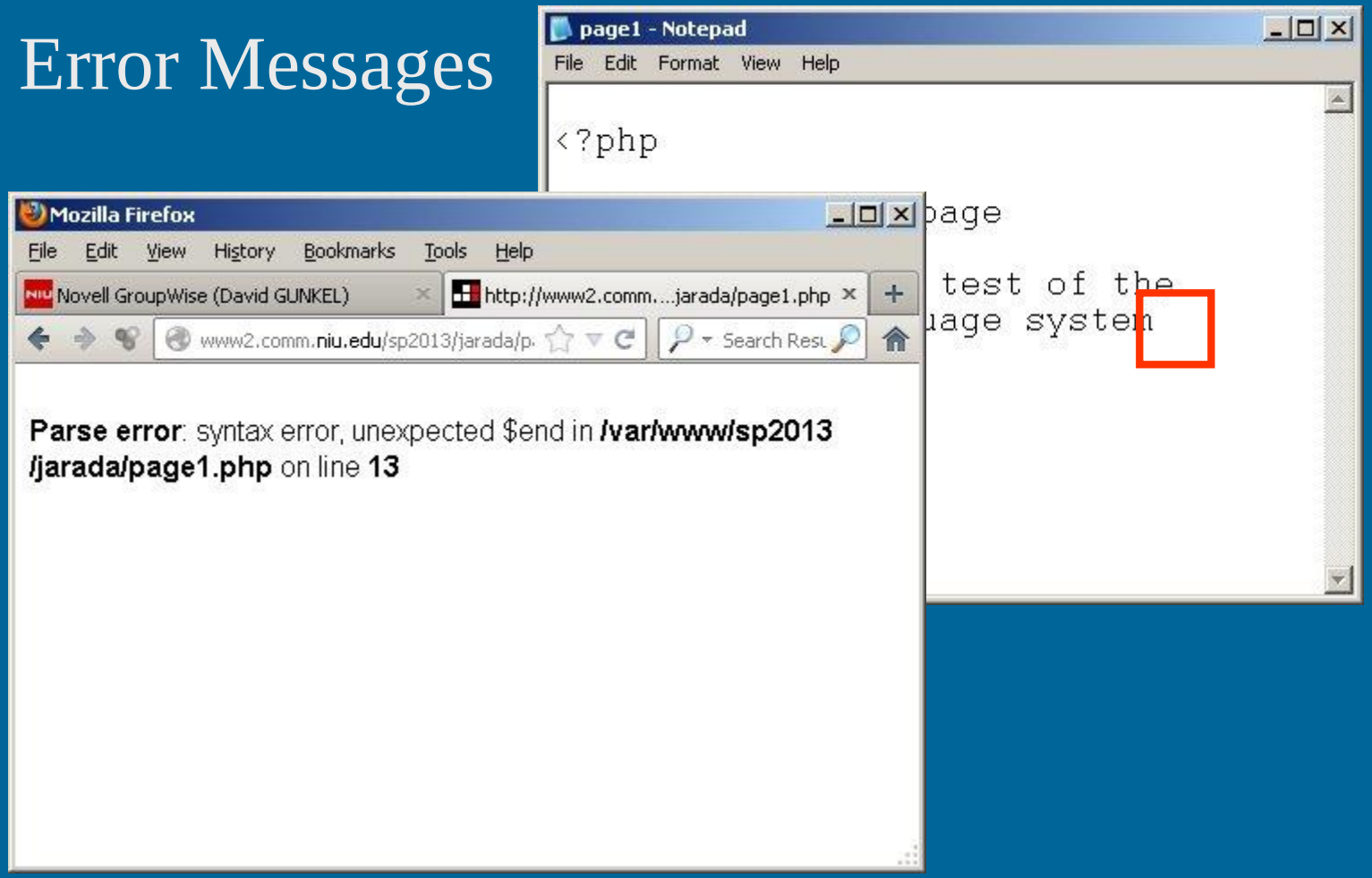
- Execute your PHP file
  - Open the browser of your choice
  - Execute the php file by typing its location in the address block

http://10.159.9.155/jwinston/page1.php



Writing PHP

- Error Messages



Variables and Operators

Variables and Operators

- Forms of PHP Data
  - *Literals*
    - Data with a fixed value
    - i.e. 1.5 or "the dog stinks like rotten pumpkins"
  - *Variables*
    - A form of data that can take a number of different values; a container that can accept different values
    - i.e. $some_variable
      - Begin with a dollar sign ($)
      - Follow the dollar sign with a name. The name cannot have a space in it; use the underline for a space.

Variables and Operators

- Assignment Statement
  - The way you associate a value with a variable
    - Use the equals sign between the variable and value
    - End the assignment statement with a semi-colon
  - Examples
    - Assigning a literal to a variable

      $some_variable = 6.559;

      $some_variable = “my dog has no nose”;
    - Assigning a variable to a variable

      $some_variable = $weight

Variables and Operators

- Destroying Variables
  - Empty a variable of its assigned value
  - Two methods
    1. unset( ) function

       unset($name);
    2. Assign null value

       $name = null;

PHP vs. Javascript

$name = null is not the same as $name = " ", which assigns an empty string value to a variable. Null is nothing. An empty string is a string value.

Variables and Operators

- PHP programs use & manipulate four data types
  - *Booleans* – binary data: 1 (true) or 0 (false)
  - *Numbers* – Numeric values that can be manipulated with mathematical operators
  - *Strings* – Non-numeric information (text) that are usually displayed for human users to read
  - *NULL* – Empty or no data

Variables and Operators

- Numbers
  - *Integers*
    - Whole numbers {1, 2, 3, 4....}
    - 0 and negative numbers {-1, -2, -3, -4…}
    - Stored as 32-bit numbers (±2.1 billion)
  - *Doubles* or *Floating-Point*
    - Any fractional number {2.5, 0.4, -3.6…}
    - Stored as 64-bit numbers with 14 decimal digits; able to store values as large as 1.8×10^{308}

Variables and Operators

- Strings
  - Can contain any character
  - “String are contained by quotation marks”
  - Escape sequence characters
    - Special characters that can be included in strings
    - Write characters that have special php significance; avoid having the character interpreted as a php character

\n	linefeed
\r	carriage return
\t	horizontal tab
\\	backslash
\$	dollar sign
\”	quotation mark

Variables and Operators

- Exercise
  - Write and execute a PHP file that demonstrates the use of literals, variables, and assignment statements
  - Process
    - Write the PHP file
    - Save it as page2-1.php
    - Upload the file to the sever
    - Execute the file

page2-1 - Notepad

File Edit Format View Help

```
<?php

  //PHP literals, variables and assignments

  $integer_value = 1;
  $double_value = 1.56986;
  $string_value = "You smell as fresh as a Greyhound bus";

  echo "<h2>My Second PHP page</h2>";

  echo "<br>Integer Value - ";
  echo $integer_value;

  echo "<br>Double Value - ";
  echo $double_value;

  echo "<br>String Value -";
  echo $string_value;

?>
```

page2-1 - Notepad

File Edit Format View Help

```
<?php

  //PHP literals, variables and assignments

  $integer_value = 1;
  $double_value = 1.56986;
  $string_value = "You smell as fresh as a Greyhound bus";

  echo "<h2>My Se

  echo "<br>Intege
  echo $integer_va

  echo "<br>Double
  echo $double_val

  echo "<br>String Value -";
  echo $string_value;

?>
```

1. Declare three variables and use assignment statements to assign literal values to them. The first variable is assigned an *integer*, the second is assigned a *double*, and the third variable is assigned a *string value*.

page2-1 - Notepad

File Edit Format View Help

```
<?php

  //PHP literals, va

  $integer_value = 1
  $double_value = 1.
  $string_value = "
```

```
  echo "<h2>My Second PHP page</h2>";

  echo "<br>Integer Value - ";
  echo $integer_value;

  echo "<br>Double Value - ";
  echo $double_value;

  echo "<br>String Value -";
  echo $string_value;
```

```
?>
```

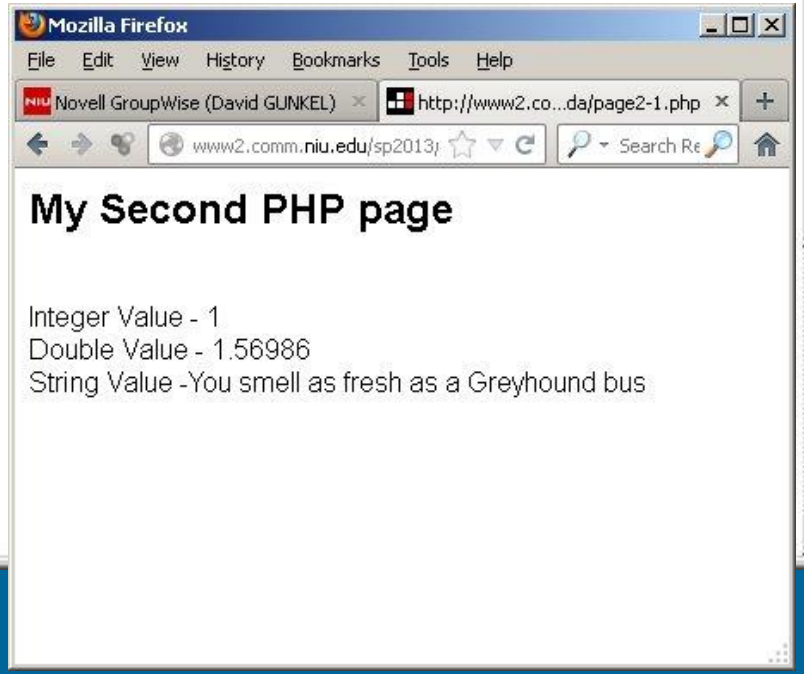
2. Use echo statements to print out information for the user. The first echo statement is filled with a *string literal*. The second echo statement is filled with the *variable*, the value of which is established above.

page2-1 - Notepad

File Edit Format View Help

page2-1.php

```
<?php

  //PHP literals, variables and assignments

  $integer_value = 1;
  $double_value = 1.56986;
  $string_value = "You smell as fresh as a Greyhound bus";

  echo "<h2>My Second PHP page</h2>";

  echo "<br>Integer Value - ";
  echo $integer_value;

  echo "<br>Double Value - ";
  echo $double_value;

  echo "<br>String Value -";
  echo $string_value;

?>
```



Variables and Operators

- Operators
  - Standard Arithmetic operators

```
+  Addition
-   Subtraction
*   Multiplication
/   Division
```

  - Examples

```
$profit = $sales - $expenses
$area = $height * $width
$tax_rate = $tax / $income
```

Variables and Operators

- Operators
  - Other (less common) operators

%	modulus	Like division, but returns remainder instead of the quotient
++	increment	Adds 1 to the value of the variable
--	decrement	Subtracts 1 from the value of the variable
.	concatenation	Used with strings; put string elements together

  - Examples

```
$x = 10 % 3;    Assign the value of 1 to $x
++$x;           Increase the value of $x by 1
--$x;           Decrease the value of $x by 1

$first = “Joe”;
$last = “Arada”;                    Concatenate two variables
$name = $first . “ ” . $last;
```

Variables and Operators

- Operators for Comparing Variables

==	Equals to	$p == $n
!=	Not equals to	$p != $n
>	Greater than	$p > $n
<	Less than	$p < $n
>=	Greater than or Equals to	$p >= $n
<=	Less than or Equals to	$p <= $n
===	Equals to and same type	$p===$n

- Logical Operators

&&	AND	$price > 5 && $price < 10
||	OR	$price = 10 || $price < 10
!	NOT	!($price > 10)

Variables and Operators

- Other Operators

+=	Add and assign
-=	Subtract and assign
*=	Multiply and assign
/=	Divide and assign
%=	Divide and assign modulus
.=	Concatenate and assign

These operators are commonly used when writing loop counters

Variables and Operators

- Exercise
  - Write and execute a PHP file that demonstrates the use of operators
  - Process
    - Write the PHP file
    - Save it as "page2-2.php"
    - Upload the file to the sever
    - Execute the file

Variables and Operators

page2-2 - Notepad

File Edit Format View Help

```
<?php

  //Calculate the area of a circle

  $radius = 2.0;
  $pi = 3.14159;
  $area = $pi * $radius * $radius;

  echo "<h1>Area of a Circle</h1>";
  echo "<br>Radius = " . $radius;
  echo "<br>Area = " . $area;

?>
```

Variables and Operators

1. Use two assignment statements to associate double numeric values to two variables.
2. Use a second assignment to calculate the area of a circle by applying the multiplication operator to the two variables.

```
ircle

    $radius = 2.0;
    $pi = 3.14159;
    $area = $pi * $radius * $radius;

    echo "<h1>Area of a Circle</h1>";
    echo "<br>Radius = " . $radius;
    echo "<br>Area = " . $area;

?>
```

Variables and Operators

page2-2 - Notepad

File Edit Format View Help

```
<?php

  //Calcula

  $radius =
  $pi = 3.1
  $area = $pi * $radius * $radius;

  echo "<h1>Area of a Circle</h1>";
  echo "<br>Radius = " . $radius;
  echo "<br>Area = " . $area;

?>
```

3. Use echo statements to display the value of the radius and area in the browser window
4. Use the concatenation operator to concatenate string values with the value of the variables

Variables and Operators

page2-2 - Notepad

File Edit Format View Help

page2-1.php

```
<?php

  //Calculate the area of a circle

  $radius = 2.0;
  $pi = 3.14159;
  $area = $pi * $radius * $radius;

  echo "<h1>Area of a Circle</h1>";
  echo "<br>Radius = " . $radius;
  echo "<br>Area = " . $area;

?>
```

Handling Form Input

- Next step
  - Apply what we know of php literals, variables scalars, arrays, operators, and functions in order to work with form entry data
  - One of the most popular applications of PHP
- Two components
  - HTML page with a form
  - PHP script to handle the form

Handling Form Input

- Exercise
  - Write an HTML form
    - form1.html
    - Upload to the server
  - Write a PHP file to process the form
    - page2-3.php
    - Upload to the server

File Edit Search View Format Language Settings Macro Run Plugins ?

form1.html | page2-3.php

```
<html>
<head>
  <title>User Information</title>
</head>

<body>
<form method="post" action="page2-3.php">

<h2>Contact List</h2>

<p><b>NICKNAME</b>
<br><input type="text"

<p><b>FULL NAME</b>
<br><input type="text" name="fullname"></p>

<p><b>ANIMAL TYPE</b>
<br><input type="radio" name="animal" value="insect">Insect
<br><input type="radio" name="animal" value="fish">Fish
<br><input type="radio" name="animal" value="reptile">Reptile
<br><input type="radio" name="animal" value="mammal">Mammal</p>

<p><b>INFORMATION</b>
<br><textarea name="information" rows="5" cols="40" wrap="virtual">
Write your information here
</textarea></p>

<p><input type="submit" value="Send Form"></p>
<p><input type="reset" value="Erase Stuff"></p>

</form>
</body>
</html>
```

Inside the opening form tag, set the method to post and the action to page2-3.php.

File Edit Search View Format Language Settings Macro Run Plugins ?

form1.html | page2-3.php

```
<html>
<head>
  <title>User Information</title
</head>

<body>
<form method="post" action="page

<h2>Contact List</h2>

<p><b>NICKNAME</b>
<br><input type="text" name="nickname"></p>

<p><b>FULL NAME</b>
<br><input type="text" name="fullname"></p>

<p><b>ANIMAL TYPE</b>
<br><input type="radio" name="animal" value="insect">Insect
<br><input type="radio" name="animal" value="fish">Fish
<br><input type="radio" name="animal" value="reptile">Reptile
<br><input type="radio" name="animal" value="mammal">Mammal</p>

<p><b>INFORMATION</b>
<br><textarea name="information" rows="5" cols="40" wrap="virtual">
Write your information here
</textarea></p>

<p><input type="submit" value="Send Form"></p>
<p><input type="reset" value="Erase Stuff"></p>

</form>
</body>
</html>
```

The form has four data entry fields. Two type=text, one type=radio and one type=textarea.

File Edit Search View Format Language Settings Macro Run Plugins ?

form1.html | page2-3.php

```
<html>
<head>
  <title>User Information</title>
</head>

<body>
<form method="post" action="page2-3.php">

<h2>Contact List</h2>

<p><b>NICKNAME</b>
<br><input type="text" name="nickname"></p>

<p><b>FULL NAME</b>
<br><input type="text" name="fullname"></p>

<p><b>ANIMAL TYPE</b>
<br><input type="radio" name="animal" value="insect">Insect
<br><input type="radio" name="animal" value="fish">Fish
<br><input type="radio" name="animal" value="reptile">Reptile
<br><input type="radio" name="animal" value="mammal">Mammal</p>

<p><b>INFORMATION</b>
<br><textarea name="information" rows="5" cols="40" wrap="virtual">
Write your information here
</textarea></p>

<p><input type="submit" value="Send Form"></p>
<p><input type="reset" value="Erase Stuff"></p>

</form>
</body>
</html>
```

form1.html

PHP Script to handle the form

Notepad++ - N:\niu-classes\coms469\spring2006\ppt2013\class0...

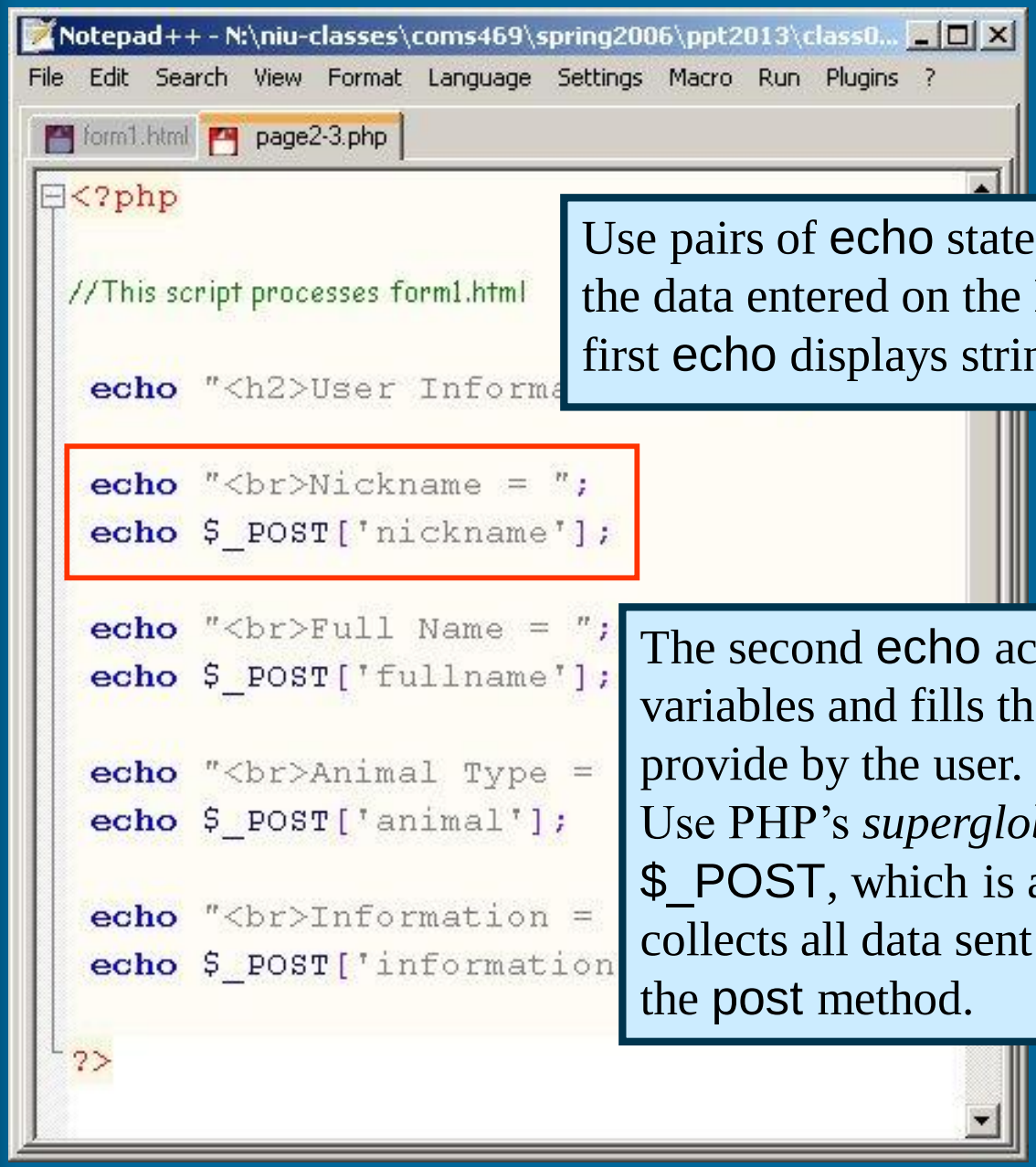
File Edit Search View Format Language Settings Macro Run Plugins ?

form1.html page2-3.php

```php
<?php

//This script processes form1.html

 echo "<h2>User Information</h2>";

 echo "<br>Nickname = ";
 echo $_POST['nickname'];

 echo "<br>Full Name = ";
 echo $_POST['fullname'];

 echo "<br>Animal Type = ";
 echo $_POST['animal'];

 echo "<br>Information = ";
 echo $_POST['information'];

?>
```

Use an echo statement to put a heading on the page.

PHP Script to handle the form



Use pairs of echo statements to present the data entered on the HTML form. The first echo displays string data.

The second echo accesses the form variables and fills them with the values provide by the user.
Use PHP's *superglobal variable* $_POST, which is an array that collects all data sent to the server by the post method.

PHP Script to handle the form

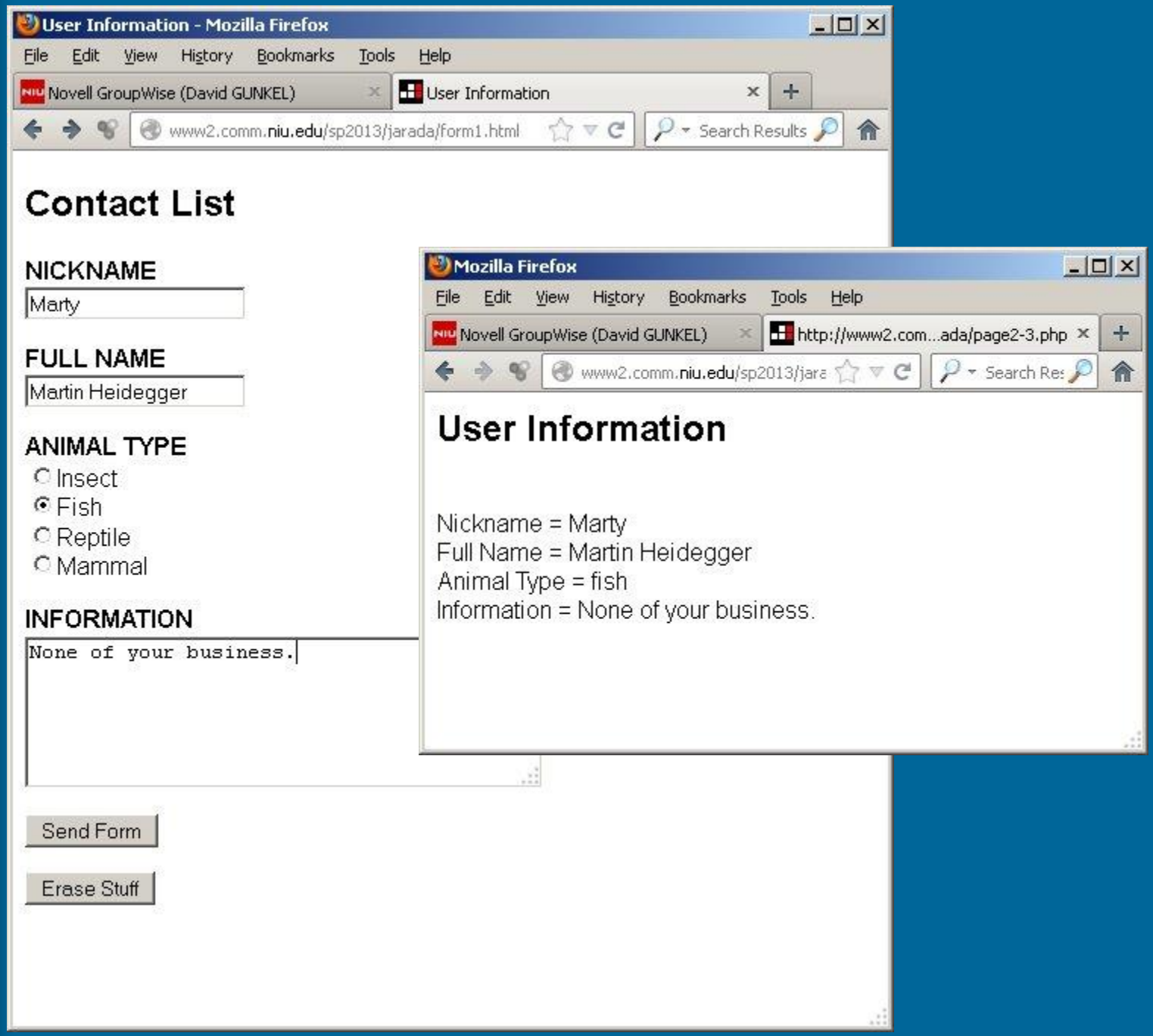
Notepad++ - N:\niu-classes\coms469\spring2006\ppt2013\class0...

File Edit Search View Format Language Settings Macro Run Plugins ?

form1.html | page2-3.php

page2-3.php

```php
<?php

//This script processes form1.html

 echo "<h2>User Information</h2>";

 echo "<br>Nickname = ";
 echo $_POST['nickname'];

 echo "<br>Full Name = ";
 echo $_POST['fullname'];

 echo "<br>Animal Type = ";
 echo $_POST['animal'];

 echo "<br>Information = ";
 echo $_POST['information'];

?>
```


User Information - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Novell GroupWise (David GUNKEL)
User Information
www2.comm.niu.edu/sp2013/jarada/form1.html
Search Results
Contact List
NICKNAME
Marty
FULL NAME
Martin Heidegger
ANIMAL TYPE
Insect
Fish
Reptile
Mammal
INFORMATION
None of your business.
Send Form
Erase Stuff
Mozilla Firefox
File Edit View History Bookmarks Tools Help
Novell GroupWise (David GUNKEL)
http://www2.com...ada/page2-3.php
www2.comm.niu.edu/sp2013/jara
Search Res
User Information
Nickname = Marty
Full Name = Martin Heidegger
Animal Type = fish
Information = None of your business.

Handling Form Input

- Variable Types
  - Variables come in different types
    - Integer
    - Double
    - String
  - At times, a program must change a string to an integer and vice versa
    - Take user input and perform a calculation
    - What the user enters is a string variable; what PHP needs to do the calculation is either an integer or a double number

Handling Form Input

- Variable Types
  - PHP is a dynamically typed programming language
  - Normally you do not have to specify types; PHP determines the type of variable based on the last value assigned to it
  - PHP does what is called type juggling
    - i.e. if one variable in an expression is a double, PHP treats every variable as a double
    - if all the variables are integers, PHP treats everything like an integer

Handling Form Input

- Variable Types
  - You can specify types manually; this is called *type casting* or *casting*

```
$x = 1;
$y = 2.5;
$z = $x + (integer) $y;
echo $z
```

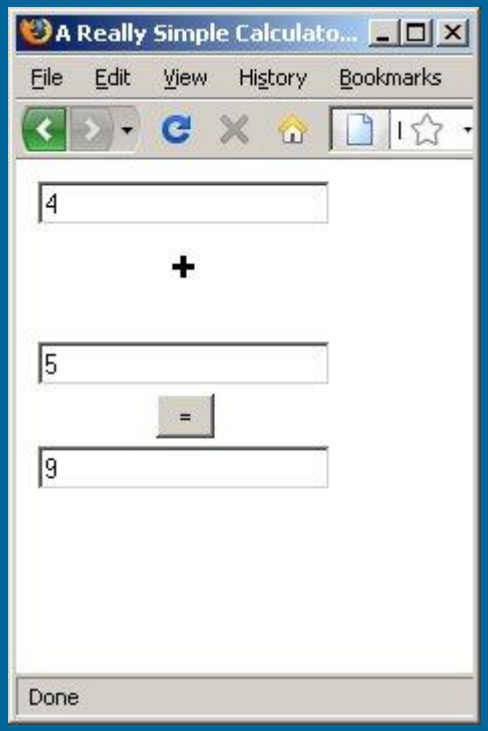
  - In most cases this is unnecessary. But you should know that it is an option.

Handling Form Input

- PHP & HTML
  - PHP is “an HTML embedded scripting language”
    - This means that PHP commands can be placed within HTML documents and HTML code can be placed inside PHP programs.
    - This makes developing PHP programs for delivering web content much faster and easier
    - This integration is unique to PHP; it does not exist in the other server-side applications

Handling Form Input

- Exercise
  - Really simple calculator
    - Demonstrate PHP's dynamic typing
    - Show how PHP commands can be intermingled with HTML
  - Process
    - Write one document that uses both HTML and PHP code; save as page2-4.php
    - Upload the file to the server and open it in the browser



page2-4.php

```
1  <html>
2  <head>
3    <title>A Really Simple Calculator
4  </head>
5
6  <body>
7    <?php
8        $first = $_POST['first'];
9        $second = $_POST['second'];
10     ?>
11 <form method="post" action="page2-4
12
13 <p><input type="text" name="first" value="<?php echo $first ?>"></p>
14
15 <p>     <b>+</b></p>
16
17 <p><input type="text" name="second" value="<?php echo $second ?>"></p>
18
19 <p>    <input type="submit" value="="></p>
20
21 <p><input type="text" name="result" value="<?php echo $first + $second ?>"></p>
22
23 </form>
24 </body>
25 </html>
26
```

Initialize the values of the two variables. Assign the value of each variable to the data entered on the form. Use the $_POST[ ] superglobal to access this data.

PHP Hypertext Preprocessor file | length : 568 lines : 29 | Ln : 28 Col : 1 Sel : 0 | 0 | Dos\Windows | ANSI as UTF-8 | INS

page2-4.php

```
1  <html>
2  <head>
3    <title>A Really Simp
4  </head>
5
6  <body>
7    <?php
8        $first = $_POST['
9        $second = $_POST['second'];
10    ?>
11 <form method="post" action="page2-4.php" width="300">
12
13 <p><input type="text" name="first" value="<?php echo $first ?>"></p>
14
15 <p>     <b>+</b></p>
16
17 <p><input type="text" name="second" value="<?php echo $second ?>"></p>
18
19 <p>    <input type="submit" value="="></p>
20
21 <p><input type="text" name="result" value="<?php echo $first + $second ?>"></p>
22
23 </form>
24 </body>
25 </html>
26
```

1. Because the PHP commands are in the HTML document, the action attribute of the form tag refers to itself. When the form is submitted, it simply calls itself.

PHP Hypertext Preprocessor file | length : 568 lines : 29 | Ln : 28 Col : 1 Sel : 0 | 0 | Dos\Windows | ANSI as UTF-8 | INS

page2-4.php

```
1  <html>
2  <head>
3    <title>A Re
4  </head>
5
6  <body>
7    <?php
8        $first =
9        $second
10     ?>
11 <form method="post" action="page2-4.php" width="300">
12
13 <p><input type="text" name="first" value="<?php echo $first ?>"></p>
14
15 <p>     <b>+</b></p>
16
17 <p><input type="text" name="second" value="<?php echo $second ?>"></p>
18
19 <p>    <input type="submit" value="="></p>
20
21 <p><input type="text" name="result" value="<?php echo $first + $second ?>"></p>
22
23 </form>
24 </body>
25 </html>
26
```

2. Two text input fields are created for the user to enter two numbers. The name of these input fields will be first and second. The value of these variables will be set to the numeric data entered by the user.

page2-4.php

```
1  <html>
2  <head>
3    <title>A Really Simple Calculator</title>
4  </head>
5
6  <body>
7    <?php
8        $first =
9        $second =
10     ?>
11 <form method="
12
13 <p><input type=
14
15 <p>    
16
17 <p><input type="text" name="second" value="<?php echo $second ?>"></p>
18
19 <p>    <input type="submit" value="="></p>
20
21 <p><input type="text" name="result" value="<?php echo $first + $second ?>"></p>
22
23 </form>
24 </body>
25 </html>
26
```

3. The results of the form are communicated on the form. This is done by providing another text input field named result. The value of this field is determined by applying the addition operator to the two variables and displaying this sum in the form input field by using an echo.

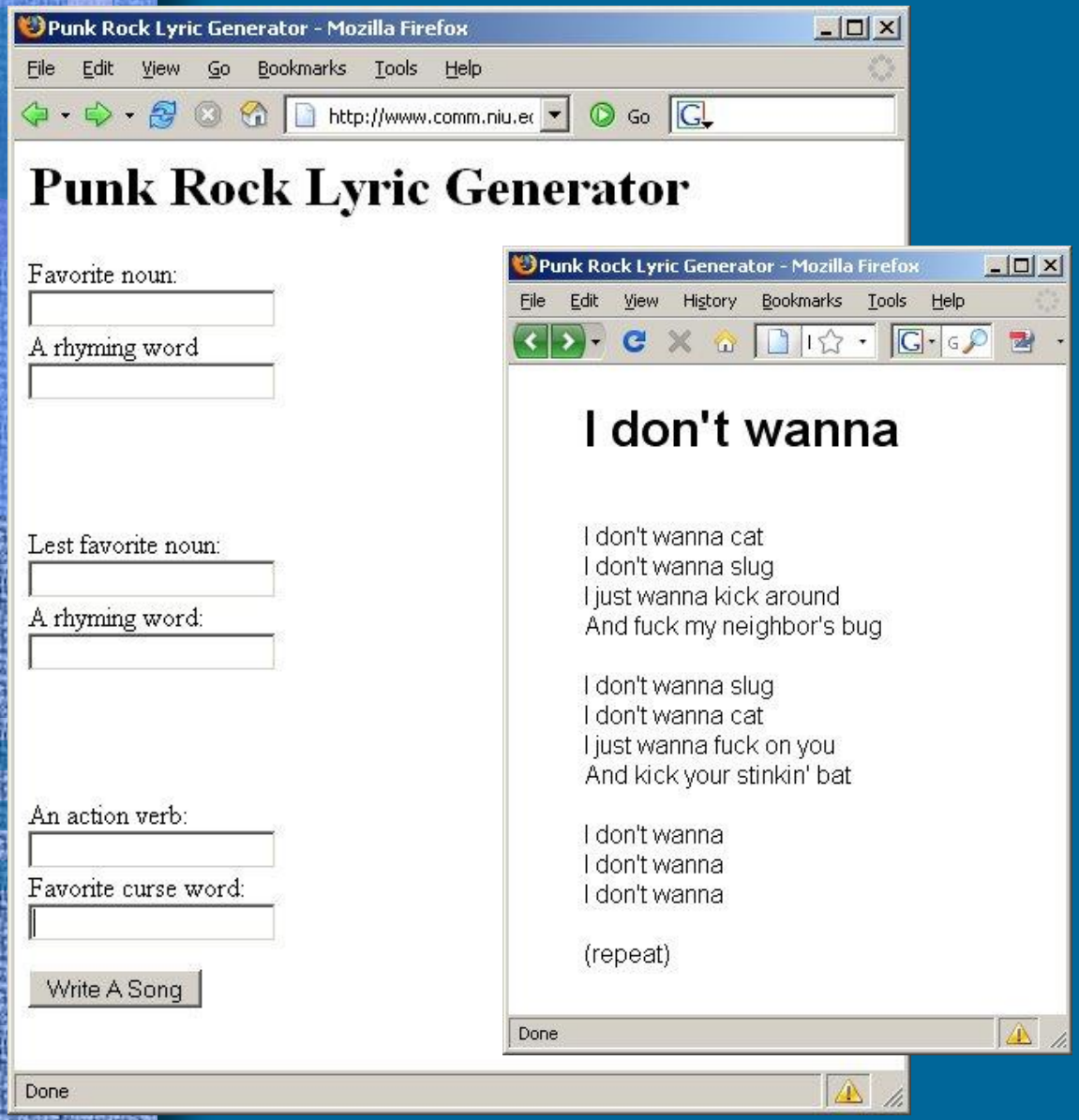
PHP Hypertext Preprocessor file | length : 568 lines : 29 | Ln : 28 Col : 1 Sel : 0 | 0 | Dos\Windows | ANSI as UTF-8 | INS

page2-4.php

```
<html>
<head>
  <title>A Really Simple Calculator</title>
</head>

<body>
  <?php
      $first = $_POST['first'];
      $second = $_POST['second'];
   ?>
<form method="post" action="page2-4.php" width="300">

<p><input type="text" name="first" value="<?php echo $first ?>"></p>

<p>     <b>+</b></p>

<p><input type="text" name="second" value="<?php echo $second ?>"></p>

<p>    <input type="submit" value="="></p>

<p><input type="text" name="result" value="<?php echo $first + $second ?>"></p>

</form>
</body>
</html>

```

PHP Hypertext Preprocessor file | length : 568 lines : 29 | Ln : 28 Col : 1 Sel : 0 | 0 | Dos\Windows | ANSI as UTF-8 | INS

Preview

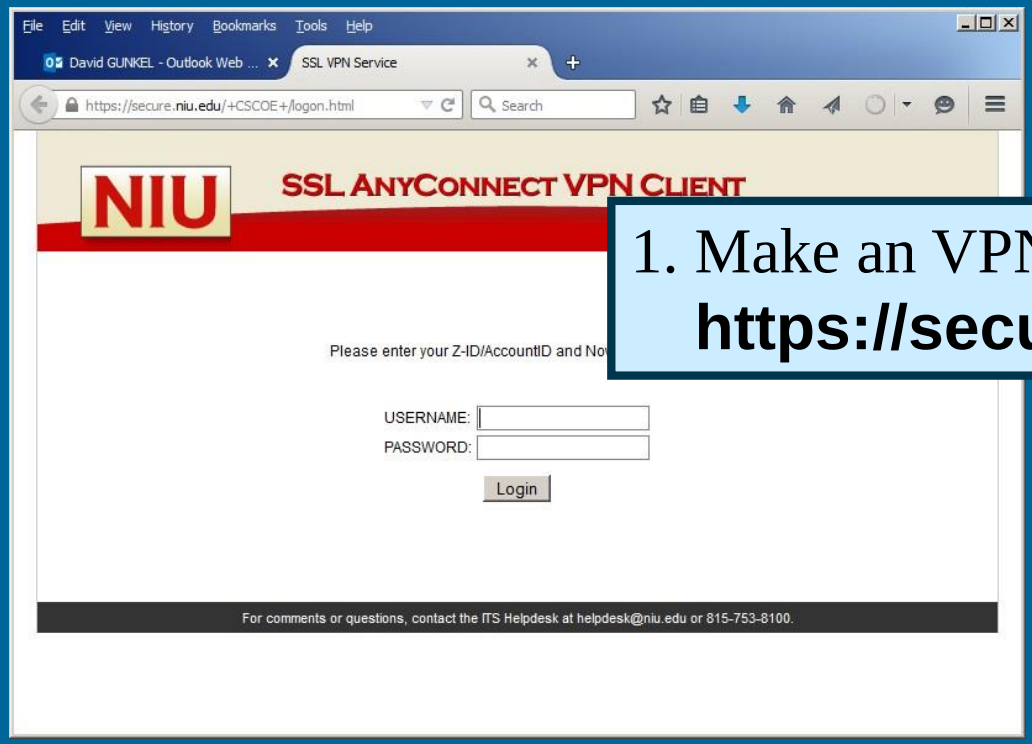


Take Home Exercise

1) Write HTML form
2) Write PHP file to take the form data and write an output page

Preview

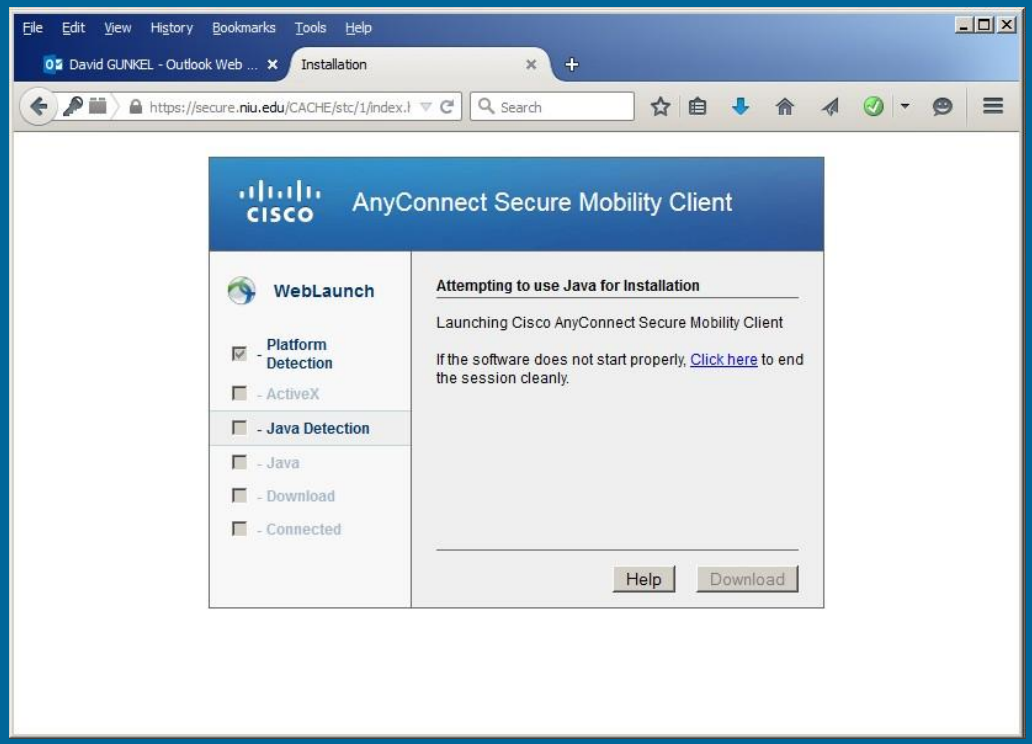
- Using PHP at home



1. Make an VPN connection at **https://secure.niu.edu**

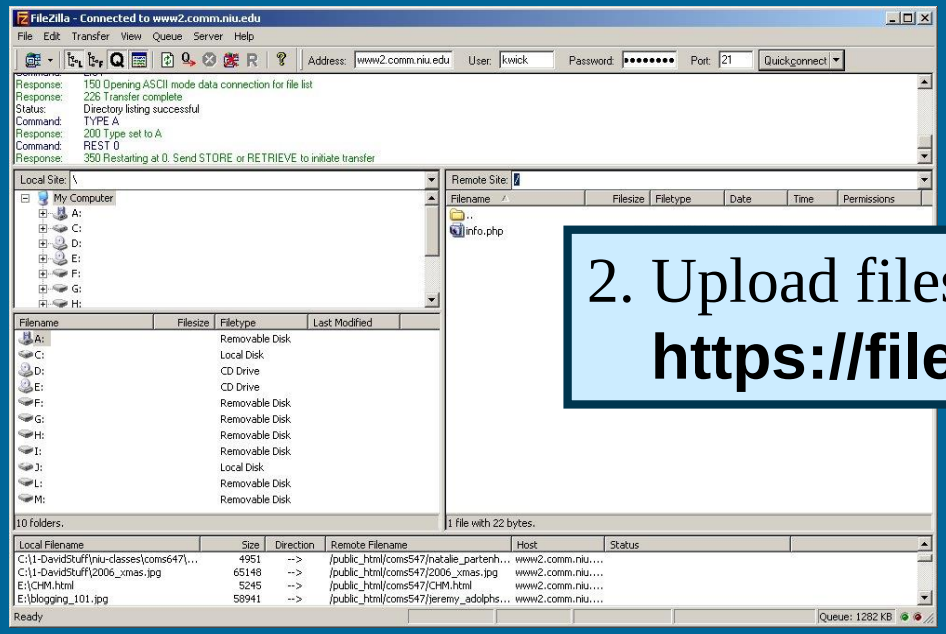
Preview

- Using PHP at home



Writing PHP

- Using PHP at home



2. Upload files using FileZilla
   https://filezilla-project.org/

Preview

- Using PHP at home

3. Open the file from the server
http://10.159.9.155/jstalin/page1.php

Preview

- Ch 3 Controlling Program Flow (pp. 49-84)
  - Conditional Statements
  - Repeating Actions - Loops
- Ch 4 Working with Arrays (pp. 85-120)
  - Storing Data in Arrays
  - Iterating over an Array
  - Date and Timers