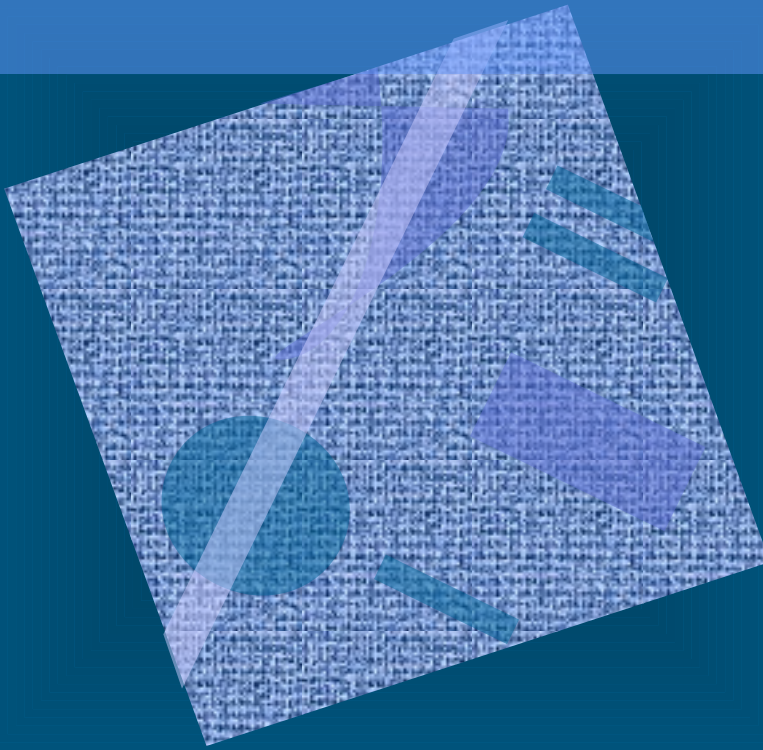


COMS 469: Interactive Media II



Agenda

- Review
- Conditional Statements
- String and Numeric Functions
- Arrays

Review

- PHP History
 - Rasmus Lerdorf 1995
 - Andi Gutmans & Zeev Suraski
 - Versions
 - 1998 – PHP 2.0
 - 2000 – PHP 4.0
 - 2004 – PHP5.0



Review

- PHP Applications



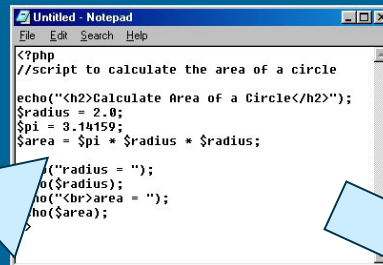
Content Management Site



Review

- Writing PHP – Development cycle

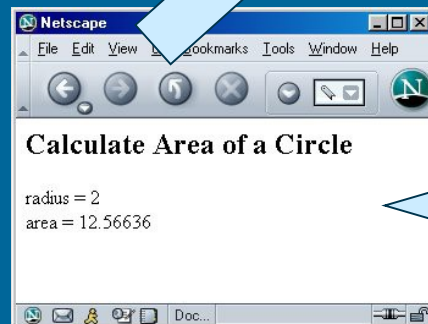
1. Write PHP script in NotePad or NotePad++



```
<?php
//script to calculate the area of a circle

echo("<h2>Calculate Area of a Circle</h2>");
$radius = 2.0;
$pi = 3.14159;
$area = $pi * $radius * $radius;

echo("radius = ");
echo($radius);
echo("<br>area = ");
echo($area);
>
```

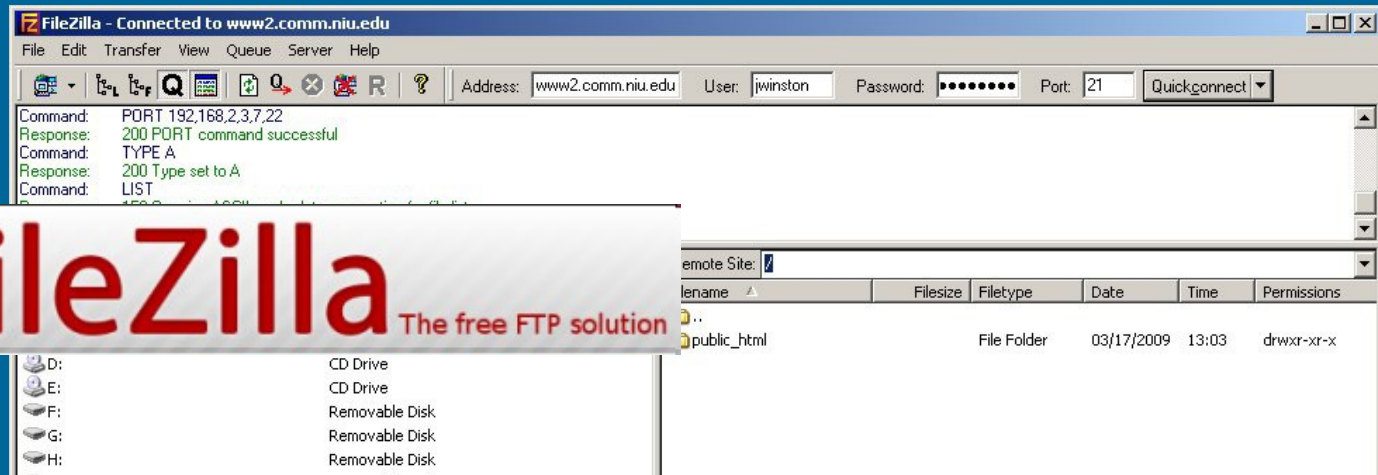


3. View file in browser



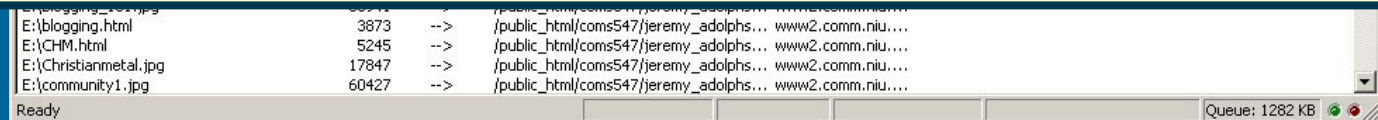
2. Upload to the Server

Review



FileZilla The free FTP solution

<i>Host</i>	10.159.9.155
<i>Username</i>	1 st initial + last name = jwinston
<i>Password</i>	8 digit zID = 01625661
<i>URL</i>	http://10.159.9.155/jwinston/page1.php



<http://filezilla-project.org/>

Review

Literals, variables & assignment statements

```
page2-1 - Notepad
File Edit Format View Help

<?php

//PHP literals, variables and assignment statements

$integer_value = 1;
$double_value = 1.56983;
$string_value = "You smell as fresh as a Greyhound bus.";

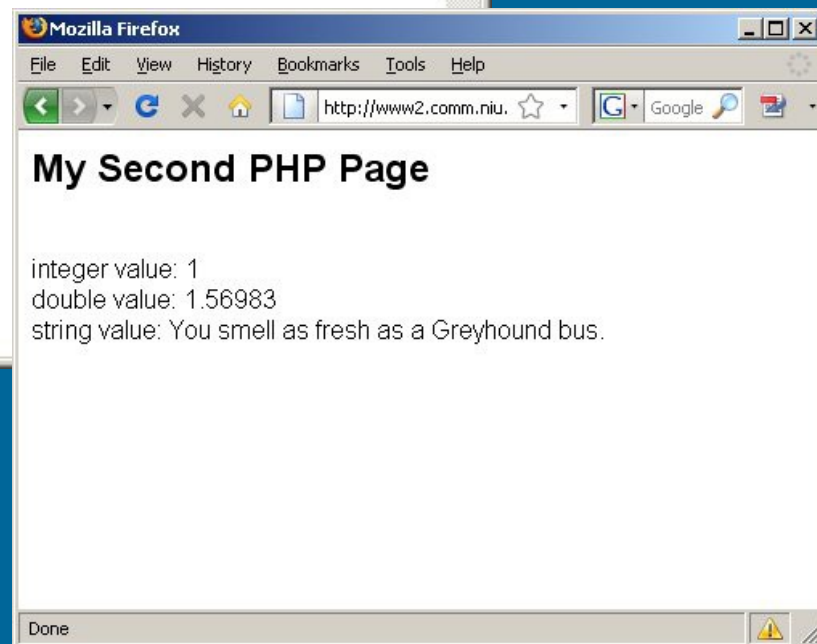
echo("<h2>My Second PHP Page</h2>");

echo("<br>integer value: ");
echo($integer_value);

echo("<br>double value: ");
echo($double_value);

echo("<br>string value: ");
echo($string_value);

?>
```



Review

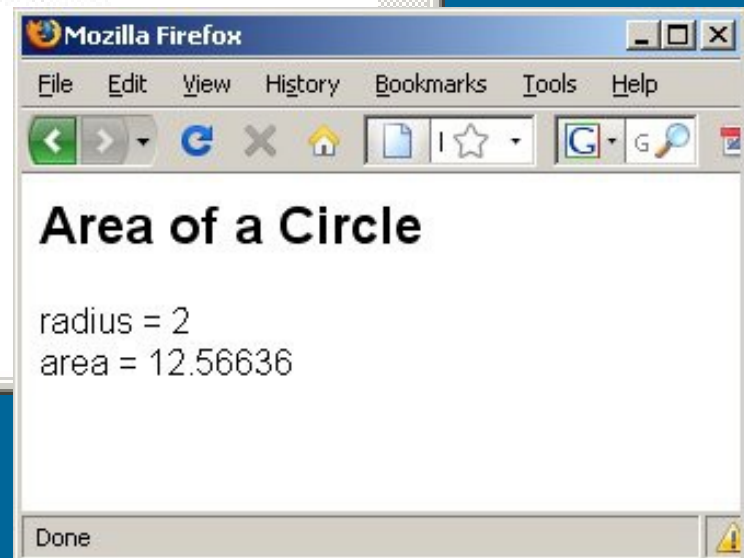
PHP Operators

```
page2-2 - Notepad
File Edit Format View Help

<?php
//Calculate the area of a circle
echo("<h2>Area of a Circle</h2>");
$radius = 2.0;
$pi = 3.14159;
$area = $pi * $radius * $radius;

echo("radius = ");
echo($radius);
echo("<br>area = ");
echo($area);

?>
```



Review

PHP & HTML forms

```
page4-1 - Notepad
File Edit Format View Help

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

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

<h2>Contact List</h2>

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

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

<p><b>ANIMAL TYPE</b><br><input type="radio" name="
<br><input type="radio" name="
<br><input type="radio" name="
<br><input type="radio" name="

<p><b>INFORMATION</b><br><textarea name="informati
Write your information here<br></textarea></p>

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

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

```
page4-1 - Notepad
File Edit Format View Help

<?php
//This script process the form on page4-1.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'];
?>
```



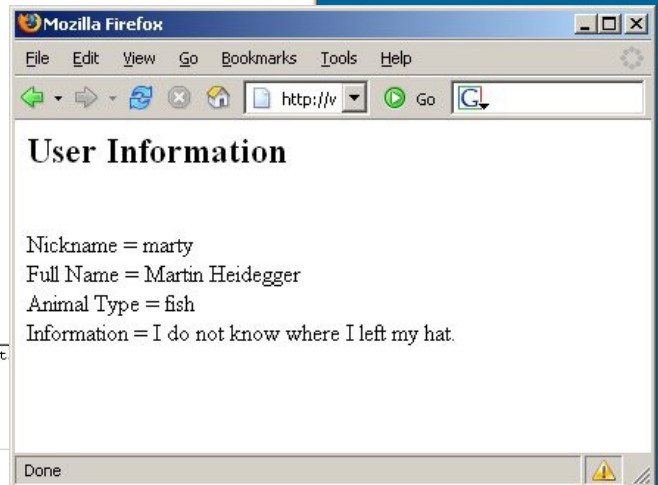
Contact List

NICKNAME

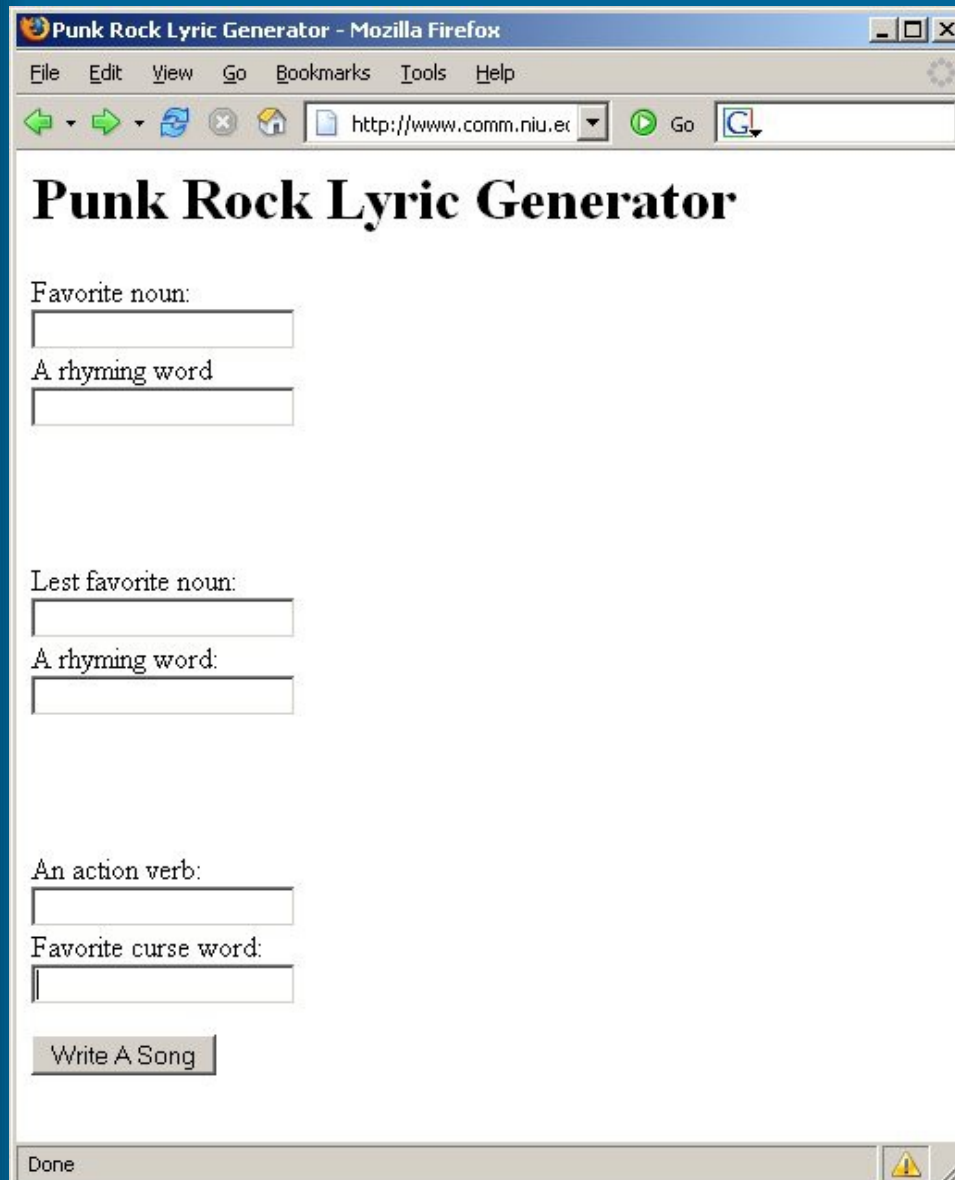
FULL NAME

ANIMAL TYPE
☐ Insect
☒ Fish
☐ Reptile
☐ Mammal

INFORMATION



Review



The screenshot shows a Mozilla Firefox browser window with the title 'Punk Rock Lyric Generator - Mozilla Firefox'. The address bar shows 'http://www.comm.niu.ec'. The page content is titled 'Punk Rock Lyric Generator' and contains three sets of input fields for generating lyrics. Each set includes a label, a text input field, and a 'Write A Song' button. The status bar at the bottom shows 'Done' and a warning icon.

Punk Rock Lyric Generator - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://www.comm.niu.ec Go

Punk Rock Lyric Generator

Favorite noun:

A rhyming word

Least favorite noun:

A rhyming word:

An action verb:

Favorite curse word:

Write A Song

Done

Review Exercise

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

```
<html>
<head>
<title>Punk Rock Lyric Generator</title>
</head>

<body>
<h1>Punk Rock Lyric Generator</h1>
<form method="post" action="lyric_generator.php">

<p>Favorite noun:
<br><input type="text" name="word1a">

<br>A rhyming word
<br><input type="text" name="word1b"></p>
<br>&nbsp;

<p>Lest favorite noun:
<br><input type="text" name="word2a">

<br>A rhyming word:
<br><input type="text" name="word2b"></p>
<br>&nbsp;

<p>An action verb:
<br><input type="text" name="verb1">

<br>Favorite curse word:
<br><input type="text" name="verb2"></p>

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

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

HTML Form

- 1) Action calls the php file
- 2) Text Input with variable names


```

3 <head>
4 <title>Punk Rock Lyric Generator</title>
5 </head>
6
7 <body>
8 <blockquote>
9
10 <h1>I don't wanna</h1>
11
12 <?php
13     $word1a = $_POST['word1a'];
14     $word1b = $_POST['word1b'];
15     $word2a = $_POST['word2a'];
16     $word2b = $_POST['word2b'];
17     $verb1  = $_POST['verb1'];
18     $verb2  = $_POST['verb2'];
19 ?>
20
21 <br>I don't wanna <?php echo $word1a ?>
22 <br>I don't wanna <?php echo $word2a ?>
23 <br>I just wanna <?php echo $verb1 ?> around
24 <br>And <?php echo $verb2 ?> my neighbor's <?php echo $word2b ?>
25 <br>&nbsp;
26
27 <br>I don't wanna <?php echo $word2a ?>
28 <br>I don't wanna <?php echo $word1a ?>
29 <br>I just wanna <?php echo $verb2 ?> on you
30 <br>And <?php echo $verb1 ?> your stinkin' <?php echo $word1b ?>
31 <br>&nbsp;
32 <br>I don't wanna
33 <br>I don't wanna
34 <br>I don't wanna
35 <br>&nbsp;
36 <br>(repeat)
37
38 </body>
39 </html>
40

```

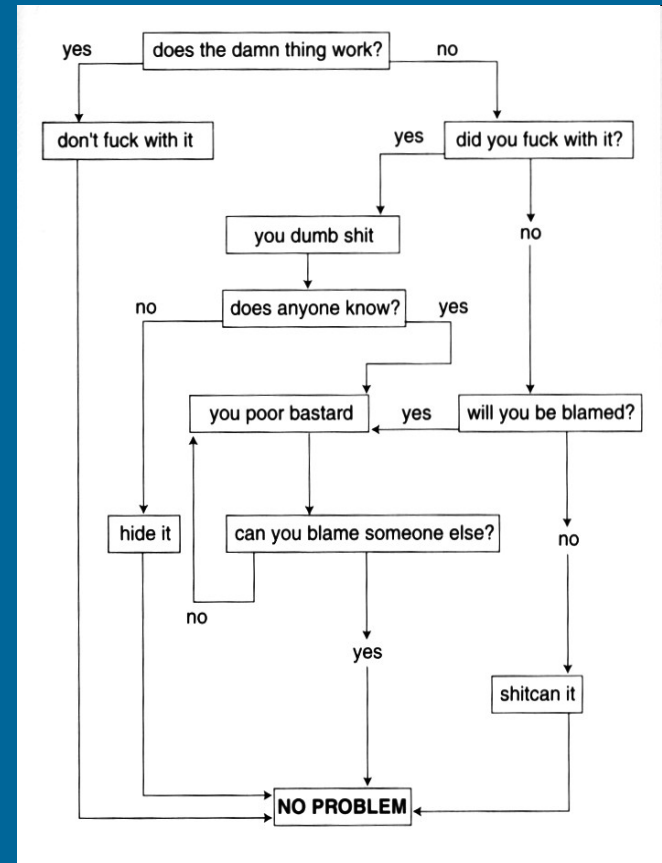
PHP File

- 1) Initialize variables
- 2) Use PHP **echo** to insert the value of the variables

Conditionals

- Introduction

- Computer programs can be designed to make *decisions*, sometimes doing one thing and sometimes doing something else based on circumstances
- PHP programs make decisions by evaluating *conditional expressions* and then executing statements based on the results of these evaluations



Conditionals

- Introduction
 - *Conditional expressions* have one of two values
 - TRUE = 1
 - FALSE = 0
 - Conditional expressions often employ relational operators
 - i.e. <, >, <=, >=, ==, !=, != =
 - Example `$a < 1`
 - Expression has the value *true* if and only if the variable “a” is less than one
 - In all other cases, the expression is *false*

Conditionals

- Introduction
 - Form complex conditional expressions by joining two conditional expressions with a *logical operator*
 - i.e. AND, &&, OR, ||, XOR, !x
 - Example
 - $\$a < 1 \text{ AND } \$b < 1$
 - This expression is true if both $\$a$ and $\$b$ have values less than one

Conditionals

- Conditional expressions are combined to write *conditional statements*
 - Conditional statements are statements that make decisions
 - Examples
 - if, else, elseif
 - switch, break
 - for
 - while, do while

Conditionals

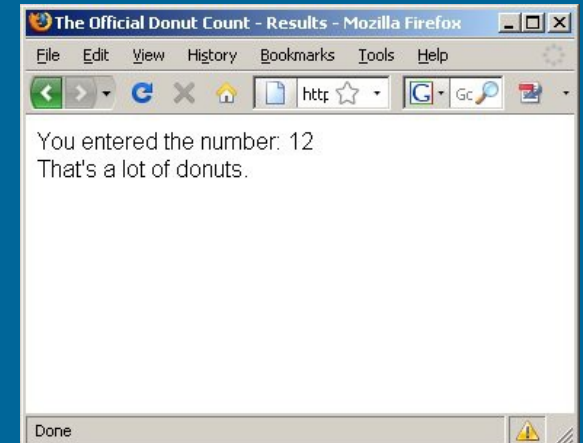
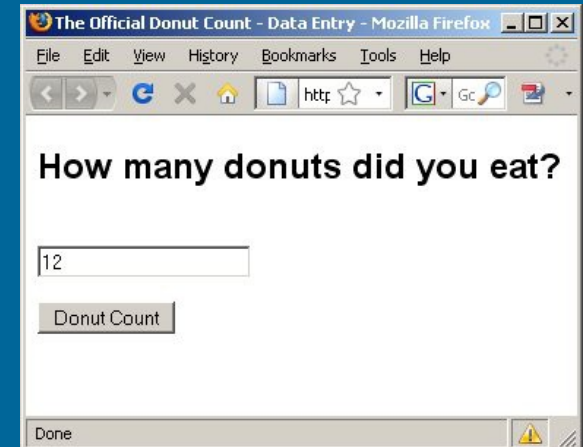
- Simple if Statements
 - Two operations
 - Evaluates a conditional expression
 - If the conditional expression is true, then execute some designated statement
 - Example

```
if ($number > 10)  
    echo "That's a lot of donuts";
```

- If the value of `$number` is greater than 10
- Then display the string "That's a lot of donuts"

Conditionals

- Exercise
 - Donut evaluator
 - Work with Conditionals
 - Write a simple if statement
 - Process
 - Write and save an html file donuts.html
 - Write and save a php file donuts.php



Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\donuts.html

File Edit Search View Format Language Settings Macro Run Plugins ?

donuts.html donuts.php

```
<html>
<head>
  <title>Official Donut Count - Data Entry</title>
</head>
<body>
  <form method="post" action="donuts.php">

    <h2>How many donuts did you eat?</h2>
    <br><input type="text" name="number">
    <p><input type="submit" value="Donut Count"></p>

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

donuts.html

Create an HTML form

The form has one entry field, and it uses **action=donuts.php** to process the data entered on the form.

Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\donuts.php

File Edit Search View Format Language Settings Macro Run Plugins ?

donuts.html donuts.php

```
<html>
<head>
  <title>Official
</head>

<body>

  <?php
    $number = $_POST['number'];
    echo "You entered the number: $number";

    if ($number > 5)
      echo "<br>That's a lot of donuts.";
  ?>

</body>
</html>
```

PHP file to process the data entered on the form.

- Initialize the value of the variable `$number`
- `echo` the value of `$number`.
- Conditional statement. If the value of `$number` is greater than 5, then write out a statement.
- Notice how php allows you to mix string literals with variable names; no concatenation needed.

Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\donuts.php

File Edit Search View Format Language Settings Macro Run Plugins ?

donuts.html donuts.php

donuts.php

```
<html>
<head>
    <title>Official Donut Count - Results</title>
</head>

<body>

    <?php
        $number = $_POST['number'];
        echo "You entered the number: $number";

        if ($number > 5)
            echo "<br>That's a lot of donuts.";
    ?>

</body>
</html>
```


Conditionals

- More Sophisticated if Statements
 - Statement group
 - More than one outcome
 - Contained by curly brackets

```
if ($number > 5)
{
    echo "<br>That's a lot of donuts.";
    echo "<br>You must have been hungry.";
}
```

Conditionals

- More Sophisticated if Statements
 - if...else
 - Two different results
 - One is executed when the conditional expression is true and another (**else**) when it evaluates false

```
if ($number > 5)
    echo "<br>That's a lot of donuts.";
else
    echo "<br>Next time, try harder.";
```

Conditionals

- More Sophisticated if Statements
 - Nesting conditional statements
 - Put a conditional statement inside a conditional
 - i.e. an if...else statement inside an if...else statement

```
if ($number > 5)
  if ($number > 10)
    echo "<br>That's really too many donuts.";
  else
    echo "<br>That's a lot of donuts.";
else
  echo "<br>Next time, try harder.";
```

Conditionals

- More Sophisticated if Statements
 - elseif statements
 - Simplify nesting of if statements
 - Combination of **else** and **if**

```
if ($number > 10)
    echo "<br>That's really too many donuts.";
elseif ($number > 5)
    echo "<br>That's a lot of donuts.";
else
    echo "<br>Next time, try harder.";
```


Conditionals

- Switch-case Statements

- Operate like **elseif** statements, but selects from alternatives represented by numbers instead of conditional expressions
- A **switch-case** statement includes a number of **cases**, which it tries to match, and a **default**, which is executed if there is no match

```
switch ($brain_size)
{
    case "small":
        echo "Your brain is the size of a pea.";
        break;
    case "medium":
        echo "Your brain is the size of a peanut.";
        break;
    default:
        echo "Your brain is the size of an eggplant.";
        break;
}
```

Conditionals

- Exercise
 - Validating User Input
 - Use conditional statements
 - Validate form data with PHP
 - Process
 - Create an HTML form
user_registration.html
 - Write PHP to validate input
user_registration.php



User Registration - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Novell GroupWise (David ... User Registration

www2.comm.niu.edu

Search

User Registration

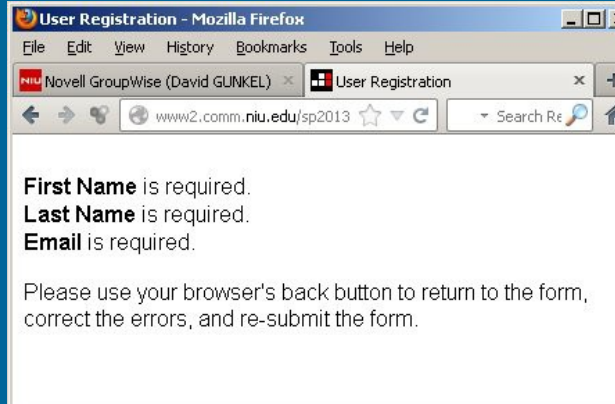
First Name:

Last Name:

Email Address:

Send

Reset



User Registration - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Novell GroupWise (David GUNKEL) User Registration

www2.comm.niu.edu/sp2013

Search

First Name is required.
Last Name is required.
Email is required.

Please use your browser's back button to return to the form, correct the errors, and re-submit the form.

user_registration.html user_registration.php

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

<body>

  <h1>User Registration</h1>
  <form method="post" action="user_registration.php">

    <p><b>First Name:</b>
    <br><input type="text" name="firstname"></p>

    <p><b>Last Name:</b>
    <br><input type="text" name="lastname"></p>

    <p><b>Email Address:</b>
    <br><input type="text" name="email"></p>

    <p><input type="submit" value="Send"></p>
    <p><input type="reset" value="Reset"></p>

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

HTML form with three type=text input fields.

The form's method should be post and the action should call **user_registration.php**, which you will create next.

user_registration.html user_registration.php

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

<body>

  <h1>User Registration</h1>
  <form method="post" action="user_registration.php">

    <p><b>First Name:</b>
    <br><input type="text" name="firstname"></p>

    <p><b>Last Name:</b>
    <br><input type="text" name="lastname"></p>

    <p><b>Email Address:</b>
    <br><input type="text" name="email"></p>

    <p><input type="submit" value="Send"></p>
    <p><input type="reset" value="Reset"></p>

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

user_registration.html


```

<html>
<head>
  <title>User Registration</title>
</head>
<body>
<?php
$firstname = $_POST['firstname'];
$lastname = $_POST['lastname'];
$email = $_POST['email'];
$errors=0;
if (!$firstname)
{
  echo "<br><b>First Name</b> is required.";
  $errors++;
}
if (!$lastname)
{
  echo "<br><b>Last Name</b> is required.";
  $errors++;
}
if (!$email)
{
  echo "<br><b>Email</b> is required.";
  $errors++;
}
if ($errors > 0)
{
  echo "<p>Please use your browser's back button to return to the form, correct ";
  if ($errors == 1)
    echo "the error, and re-submit the form.";
  else
    echo "the errors, and re-submit the form.";
}
else
  echo "Thank you, $firstname $lastname <br>We will contact you at $email";
?>
</body>
</html>

```

1) Initialize the three form variables. Initialize the \$errors variable to 0. This says, in effect, begin with the assumption that there are no errors.

```

<html>
<head>
  <title>User Registration</title>
</head>
<body>
<?php
  $firstname = $_POST['firstname'];
  $lastname = $_POST['lastname'];
  $email = $_POST['email'];
  $errors=0;
  if (!$firstname)
  {
    echo "<br><b>First Name</b> is required.";
    $errors++;
  }
  if (!$lastname)
  {
    echo "<br><b>Last Name</b> is required.";
    $errors++;
  }
  if (!$email)
  {
    echo "<br><b>Email</b> is required.";
    $errors++;
  }
  if ($errors > 0)
  {
    echo "<p>Please fill out the required fields.";
    if ($errors == 1)
      echo "the " . $errors . " field is required.";
    else
      echo "the " . $errors . " fields are required.";
  }
  else
    echo "Thank you for registering!";
?>
</body>
</html>

```

2) Three if statements that check whether the form input fields for first name, last name, and email are empty. Here's how these if statements work:

- If the user does not enter data on the form, the variable in question (i.e. `$firstname`) is not created; it does not exist.
- If the variable does not exist, then the body of the statement is executed—an `echo` statement is printed out and the variable `$errors` is incremented. If it does exist, the body of the statement is not executed.

NOTE: We could also use `if (empty($firstname))`

```

<html>
<head>
  <title>User Registration</title>
</head>
<body>
<?php
  $firstname = $_POST['firstname'];
  $lastname = $_POST['lastname'];
  $email = $_POST['email'];
  $errors=0;
  if (!$firstname)
  {
    echo "<br><b>First Name is required.</b>";
    $errors++;
  }
  if (!$lastname)
  {
    echo "<br><b>Last Name is required.</b>";
    $errors++;
  }
  if (!$email)
  {
    echo "<br><b>Email</b> is required.";
    $errors++;
  }

  if ($errors > 0)
  {
    echo "<p>Please use your browser's back button to return to the form, correct ";
    if ($errors == 1)
      echo "the error, and re-submit the form.";
    else
      echo "the errors, and re-submit the form.";
  }
  else
    echo "Thank you, $firstname $lastname <br>We will contact you at $email";
  ?>
</body>
</html>

```

3) Use nested conditional statements to produce variable results.

a) If the value of the `$errors` variable is greater than the initial value of 0 (meaning that at least one of the data entry fields is empty), then **echo** the first part of the instruction followed by two possibilities:

If the value of `$errors` is equal to 1, then **echo** "the error."

If the value of `$errors` is something else, then **echo** "the errors."

b) If the value of the errors is something else (i.e. not greater than 0), then **echo** a thank you message.

user_registration.php

```
<html>
<head>
  <title>User Registration</title>
</head>
<body>
<?php
  $firstname = $_POST['firstname'];
  $lastname = $_POST['lastname'];
  $email = $_POST['email'];
  $errors=0;
  if (!$firstname)
  {
    echo "<br><b>First Name</b> is required.";
    $errors++;
  }
  if (!$lastname)
  {
    echo "<br><b>Last Name</b> is required.";
    $errors++;
  }
  if (!$email)
  {
    echo "<br><b>Email</b> is required.";
    $errors++;
  }
  if ($errors > 0)
  {
    echo "<p>Please use your browser's back button to return to the form, correct ";
    if ($errors == 1)
      echo "the error, and re-submit the form.";
    else
      echo "the errors, and re-submit the form.";
  }
  else
    echo "Thank you, $firstname $lastname <br>We will contact you at $email";
?>
</body>
</html>
```


Strings and Numeric Functions

- Introduction
 - PHP has built-in functions for manipulating strings and numbers
 - Over 75 functions for working with strings and 50+ for working with numbers
 - Not an exhaustive account; only look at the most useful

Common PHP string Functions

FUNCTION	DESCRIPTION
empty()	Tests if a string is empty
strlen()	Calculates the number of characters in a string
strrev()	Reverses a string
str_repeat()	Repeats a string
substr()	Retrieves a section of a string
strcmp()	Compares two strings
str_word_count()	Calculates the number of words in a string
str_replace	Replaces parts of a string
trim()	Removes leading and trailing whitespace from a string
strtolower()	Lowercases as string
strtoupper()	Uppercases a string
ucfirst()	Uppercases the first character of a string
ucwords()	Uppercases the first character of every word in a string
addslashes()	Escapes special characters in a string with backslashes
stripslashes()	Removes backslashes from a string
htmlentities()	Encodes HTML within a string
htmlspecialchars()	Encodes special HTML characters within a string
nl2br()	Replaces line breaks in a string with elements
html_entity_decode()	Decodes HTML entities within a string
ereg()	Performs a regular expression match on a string
strip_tags()	Removes PHP and HTML coding from a string

String Functions

The image shows a Notepad++ editor window on the left and a Mozilla Firefox browser window on the right. The Notepad++ window is editing a file named `string_functions.php`. The code in the file is as follows:

```
<?php
$string = "There's a sucker born every

echo $string;
echo "<br>" . strrev($string);
echo "<br>" . str_repeat($string, 2);
echo "<br>" . substr($string, 10, 6);
echo "<br>" . strtolower($string);
echo "<br>" . strtoupper($string);
echo "<br>" . ucwords($string);
echo "<br>" . ucfirst($string);
echo "<br>" . strlen($string);
?>
```

The Mozilla Firefox browser window is displaying the output of the PHP script at the URL `http://www2.comm.niu.edu/sp2013/jarada/s...ing_functions.php`. The output is as follows:

There's a sucker born every minute
etunim yreve nrob rekcus a s'erehT
There's a sucker born every minuteThere's a sucker born every minute
sucker
there's a sucker born every minute
THERE'S A SUCKER BORN EVERY MINUTE
There's A Sucker Born Every Minute
There's a sucker born every minute
34

Common PHP numeric Functions

FUNCTION	DESCRIPTION
ceil()	Rounds a number up
floor()	Rounds a number down
abs()	Finds the absolute value of a number
pow()	Raises on number to the power of another
log()	Finds the logarithm of a number
exp()	Finds the exponent of a number
rand()	Generates a random number
bindec()	Converts a number from binary to decimal
decbin()	Converts a number from decimal to binary
decoct()	Converts a number from decimal to octal
dechex()	Converts a number from decimal to hexadecimal
hexdec()	Converts a number from hexadecimal to decimal
octdec()	Converts a number from octal to decimal
number_format()	Formats a number with grouped thousands and decimals Useful for working with money – 2 decimal places
printf()	Formats a number using a custom specification

Numeric Functions

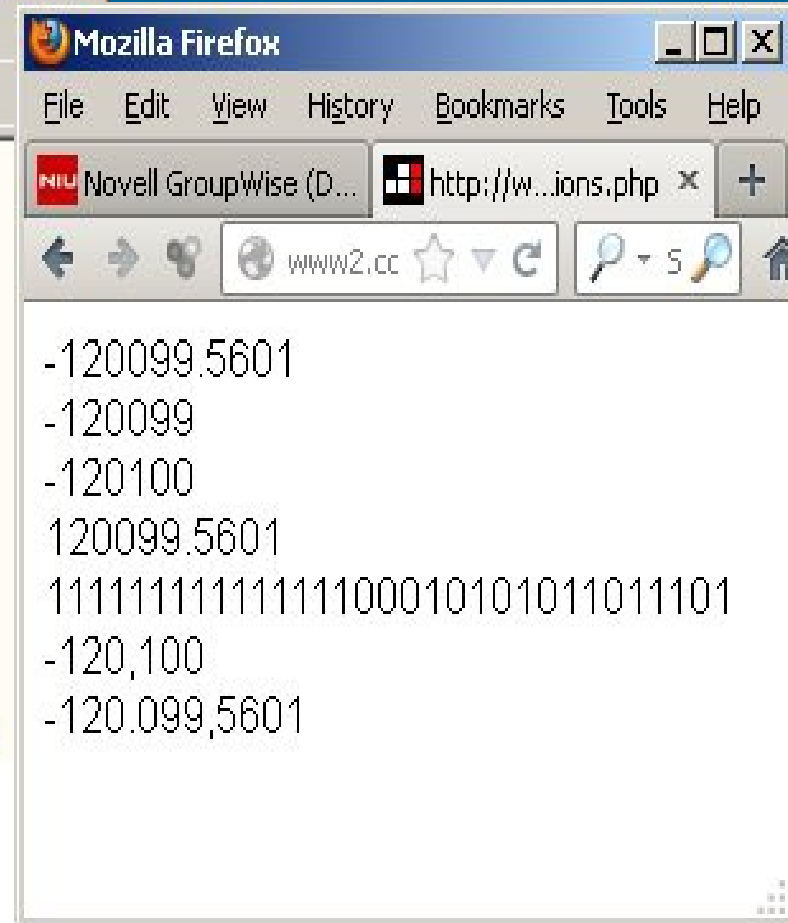
Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\nu...

File Edit Search View Format Language Settings Macro Run Plugins ?

number_functions.php

```
<?php
    $number = -120099.5601;

    echo $number;
    echo "<br>" . ceil($number);
    echo "<br>" . floor($number);
    echo "<br>" . abs($number);
    echo "<br>" . decbin($number);
    echo "<br>" . number_format($number);
    echo "<br>" . number_format($number, 4, ',', '.');
    ?>
```



String and Numeric Functions

- Exercise – Modify User Registration Form
 - Validate email address
 - Use the `ereg()` function
 - Regular expression from <http://regexlib.com/>
 - Provide detailed feedback
 - Use string functions to format username and email
 - Use `rand()` function to generate a random number password

The image displays two screenshots of a web browser window titled "User Registration - Mozilla Firefox". The browser's address bar shows the URL "www2.comm.niu.ec".

The top screenshot shows the "User Registration" form with the following fields and values:

- First Name:** Victor
- Last Name:** Frankenstein
- Email Address:** vfrank@gmail

Below the form are two buttons: "Send" and "Reset".

The bottom screenshot shows the "Login Credentials" section, which displays the results of the registration:

- Name:** VFRANKENSTEIN
- Password:** 6787
- Email:** vfrank{at}gmail.com

```

<html>
<head>
  <title>User Registration</title>
</head>
<body>
<?php
  $firstname = $_POST['firstname'];
  $lastname = $_POST['lastname'];
  $email = $_POST['email'];
  $errors=0;
  if (!$firstname)

```

Modify the email validation line
by replacing the original
if (!\$email) with
if (!ereg (pattern, subject))

```

if(!ereg ("^([a-zA-Z0-9_\-\.]+)@([a-zA-Z0-9_\-\.]+\.[a-zA-Z]{2,5})$", stripslashes(trim($email))))
{
  echo "<br><b>Valid Email</b> is required.";
  $errors++;
}

```

```

if (!$email)

```

```

{
  echo "<br><b>Email</b> is required.";
  $errors++;
}

```

```

if ($errors > 0)

```

```

{
  echo "<p>Please use

```

```

  if ($errors == 1)

```

```

    echo "the error, a

```

```

  else

```

```

    echo "the errors,

```

```

  }
  else

```

```

    echo "Thank you, $firstname $lastname <br>We will contact you at $email";

```

```

?>

```

```

</body>

```

```

</html>

```

The new line uses the **ereg()** function to test whether the string value of \$email does *not* match the pattern of the regular expression.

The **stripslashes()** function is used to remove slashes that sometimes get added to form variables and the **trim()** function trims whitespaces from the data.

user_registration.php

```
<html>
<head>
  <title>User Registration</title>
</head>
<body>
<?php
  $firstname = $_POST['firstname'];
  $lastname = $_POST['lastname'];
  $email = $_POST['email'];
  $errors=0;
  if (!$firstname)
```

```
if(!ereg ("^([a-zA-Z0-9_\-\.]+)@([a-zA-Z0-9_\-\.]+\.[a-zA-Z]{2,5})$",stripslashes(trim($email))))
{
  echo "<br><b>Valid Email</b> is required.";
  $errors++;
}
```

```
if (!$email)
{
  echo "<br><b>Email</b> is required.";
  $errors++;
}
```

```
if ($errors > 0)
{
  echo "<p>Please use your browser's back button to return to the form, correct ";
  if ($errors == 1)
    echo "the error, and re-submit the form.";
  else
    echo "the errors, and re-submit the form.";
}
else
  echo "Thank you, $firstname $lastname <br>We will contact you at $email";
```

```
?>
</body>
</html>
```

<http://regexlib.com/>


```
<html>
<head>
  <title>User Registration</title>
</head>
<body>
<?php
  $firstname = $_POST['firstname'];
  $lastname = $_POST['lastname'];
  $email = $_POST['email'];
  $errors=0;
  if (!$firstname)
  {
    echo "<br><b>First Name</b> is required.";
    $errors++;
  }
else
{
  $initial = substr($firstname, 0, 1);
  $username = strtoupper($initial . $lastname);
  $password = rand(1000,9999);
  $useremail = str_replace('@', '(at)', $email);
  echo "<h2>Login Credentials</h2>";
  echo "Name: $username <br>Password: $password <br>Email: $useremail ";
}
?>
</body>
</html>
```

```
    echo "the errors, and re-submit the form.";
  }
else
  echo "Thank you, $firstname $lastname <br>We will contact you at $email";
?>
</body>
</html>
```

user_registration.php

```
<html>
<head>
  <title>User Registration</title>
</head>
<body>
<?php
  $firstname = $_POST['firstname'];
  $lastname = $_POST['lastname'];
  $email = $_POST['email'];
  $errors=0;
  if (!$firstname)
  {
    echo "<br><b>First Name</b> is required.";
    $errors++;
  }
else
{
  $initial = substr($firstname, 0, 1);
  $username = strtoupper($initial . $lastname);
  $password = rand(1000,9999);
  $useremail = str_replace('@', '(at)', $email);
  echo "<h2>Login Credentials</h2>";
  echo "Name: $username <br>Password: $password <br>Email: $useremail ";
}
?>
</body>
</html>

echo "the errors, and re-submit the form.";
}
else
  echo "Thank you, $firstname $lastname <br>We will contact you at $email";
?>
</body>
</html>
```

Arrays

- Introduction

- Arrays are multi-element variables

- Groups or sets of things
 - Can manipulate the whole group or one or more elements in the group

- Terminology

- *Name* – identifies an array
 - *Key* – identifies each element in an array
 - *Element* – individual values in the array



`$beer[0] => pabst`

Name Key Element

Arrays

- Two ways to create an Array

- 1) Array() function

```
$beer = array("pabst", "schlitz", "blatz");
```

- Names inside the parenthesis are the names of the array elements. They are contained in quotation marks and separated from each other by commas.
- If you do not specify a key, PHP will assign consecutive key values beginning at 0
- If you want to associate a particular key with a value, use the => operator

```
$beer = array(1=>"pabst", 2=>"schlitz", 3=>"blatz");
```


Arrays

- Two ways to create an Array

2) Assignment Statement

```
$beer[0] = "pabst";  
$beer[1] = "schlitz";  
$beer[2] = "blatz";
```

- Keys do not have to be consecutive or integers

```
$beer[10] = "pabst";  
$beer[2] = "schlitz";  
$beer[35] = "blatz";
```

```
$ beer["good"] = "pabst";  
$beer["bad"] = "schlitz";  
$beer["ugly"] = "blatz";
```

Arrays

- Accessing and modifying array values
 - Access the value of an element in an array by indicating the array name and key
- Change the value associated with a key by specifying the array name, key, and new value

```
$favorite = $beer[0];
```

```
$beer[1] = "bud light";
```

- Remove an element from an array

```
unset ($beer[1]);
```

Arrays

- Loops & Iterators
 - Accessing/manipulating more than one key or value in an array
 - Terminology
 - *Iterating* – Moving through the elements of an array and accessing each element
 - *Sequential Array* – An array whose keys are consecutive integers 0, 1, 2...
 - *Non-sequential Array* – An array whose keys are not consecutive integers
 - 10, 21, 50...
 - good, bad, ugly

Arrays

- Looping Methods
 - `for()`
 - `foreach()`
 - `while()`
 - `do while()`

Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\beer_array...

File Edit Search View Format Language Settings Macro Run Plugins ?

beer_array.php

for()

```
<?php
```

```
$beer = array("pabst", "schlitz", "blatz");
```

```
for ($i=0; $i<count($beer); $i++)
```

```
{
```

```
    echo $beer[$i] . "<br>";
```

```
}
```

```
?>
```

array function

Declare and initialize an array called \$beer

Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\beer_array...

File Edit Search View Format Language Settings Macro Run Plugins ?

beer_array.php

```
<?php

$beer = array("pabst", "schlitz", "blatz");
for ($i=0; $i<count($beer); $i++)
{
    echo $beer[$i] . "<br>";
}

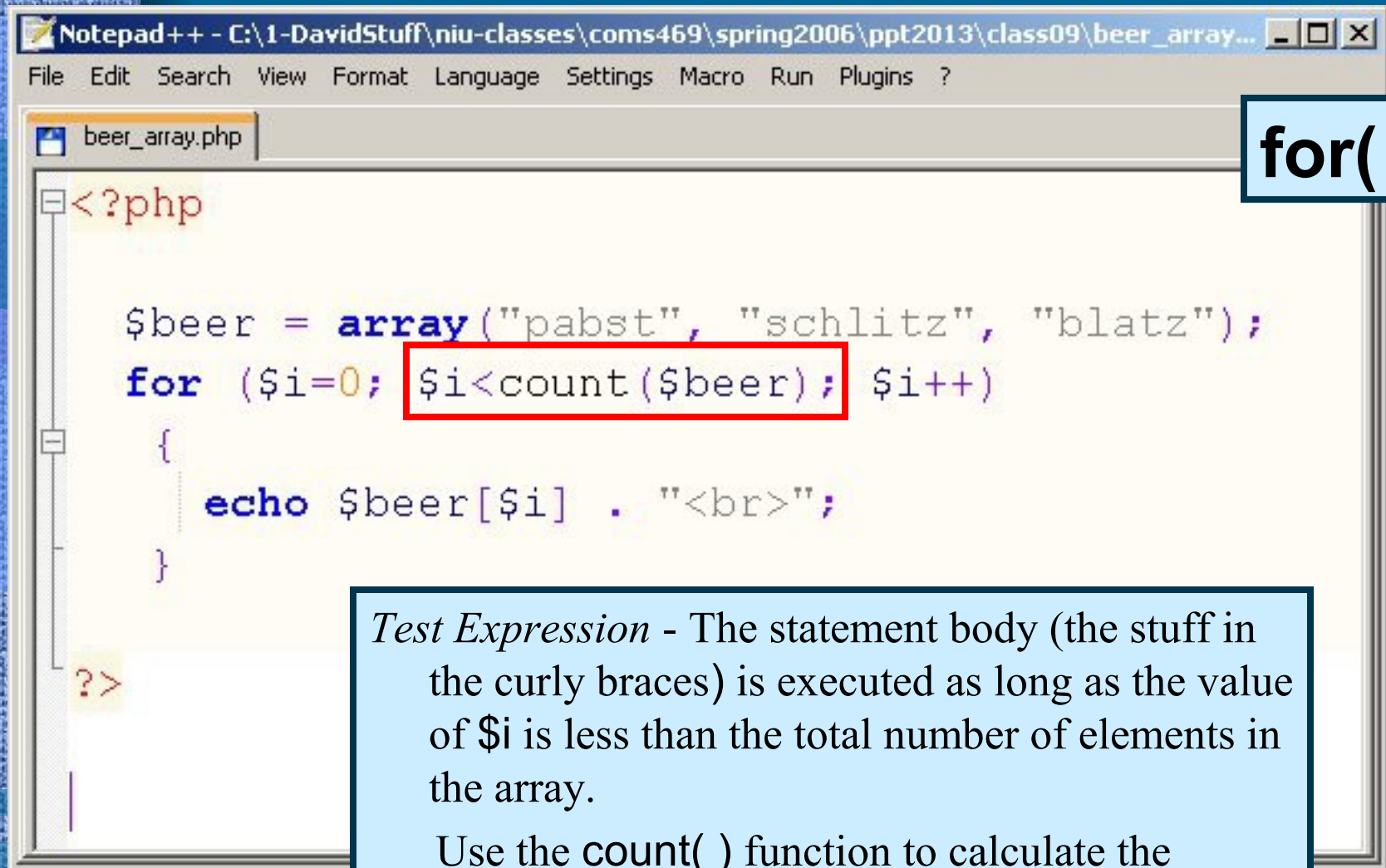
?>
```

for()

for() statement (3 components)

Initialization Expression - Set \$i to the value 0

This variable is the index counter



Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\beer_array...

File Edit Search View Format Language Settings Macro Run Plugins ?

beer_array.php

```
<?php

$beer = array("pabst", "schlitz", "blatz");
for ($i=0; $i<count($beer); $i++)
{
    echo $beer[$i] . "<br>";
}

?>
```

for()

Test Expression - The statement body (the stuff in the curly braces) is executed as long as the value of \$i is less than the total number of elements in the array.

Use the count() function to calculate the number of elements in the array

Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\beer_array...

File Edit Search View Format Language Settings Macro Run Plugins ?

beer_array.php

for()

```
<?php

$beer = array("pabst", "schlitz", "blatz");
for ($i=0; $i<count($beer); $i++)
{
    echo $beer[$i] . "<br>";
}

?>
```

Step Expression – Modifies one or more variables referenced in the test expression; increment the value of \$i


```
Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\beer_array...
File Edit Search View Format Language Settings Macro Run Plugins ?

beer_array.php

<?php

$beer = array("pabst", "schlitz", "blatz");
for ($i=0; $i<count($beer); $i++)
{
    echo $beer[$i] . "<br>";
}

?>
```

for()

Body Statement

echo the elements of the array.
Access all three elements of the
array by altering the value of the
array key [\$i]

Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\beer_array...

File Edit Search View Format Language Settings Macro Run Plugins ?

beer_array.php

for()

```
<?php
```

```
$beer = array("pabst", "schlitz", "blatz");
```

```
for ($i=0; $i<count($beer); $i++)
```

```
{
```

```
    echo $beer[$i] . "<br>";
```

```
}
```

```
?>
```

Mozilla Firefox

File Edit View History Bookmarks Tools Help

Novell Group... http:....php x ^\$ Regi

www2.con Search

pabst
schlitz
blatz

```
<?php

$beer = array("pabst", "schlitz", "blatz");
foreach ($beer as $c)
{
    echo $c . "<br>";
}

?>
```

foreach()

Unlike a `for()` loop, `foreach()` does not need a counter or the increment operator. It simply works through the array, assigning each element to the temporary variable `$c` and then echoing that value. *For each* element of `$beer` we extract and echo the value.

Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\beer_...

File Edit Search View Format Language Settings Macro Run Plugins ?

beer_array4.php

```
<?php

$beer = array("good" => "pabst",
              "bad" => "schlitz",
              "ugly" => "blatz");
foreach ($beer as $key => $value)
{
    echo "$value is a $key beer <br>";
}

?>
```

foreach()

Since there is no counter, foreach() can be used to iterate over non-sequential arrays

Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\beer_...

File Edit Search View Format Language Settings Macro Run Plugins ?

beer_array4.php

```
<?php
```

```
$beer = array("good" => "pabst",  
              "bad"  => "schlitz",  
              "ugly" => "blatz");  
foreach ($beer as $key => $value)  
{  
    echo "$value is a $key beer <br>";  
}
```

```
?>
```

foreach()

Mozilla Firefox

File Edit View History Bookmarks Tools Help

< Novell Group... http:...php x > + -

← → www2.cc S

pabst is a good beer
schlitz is a bad beer
blatz is a ugly beer

```
<?php

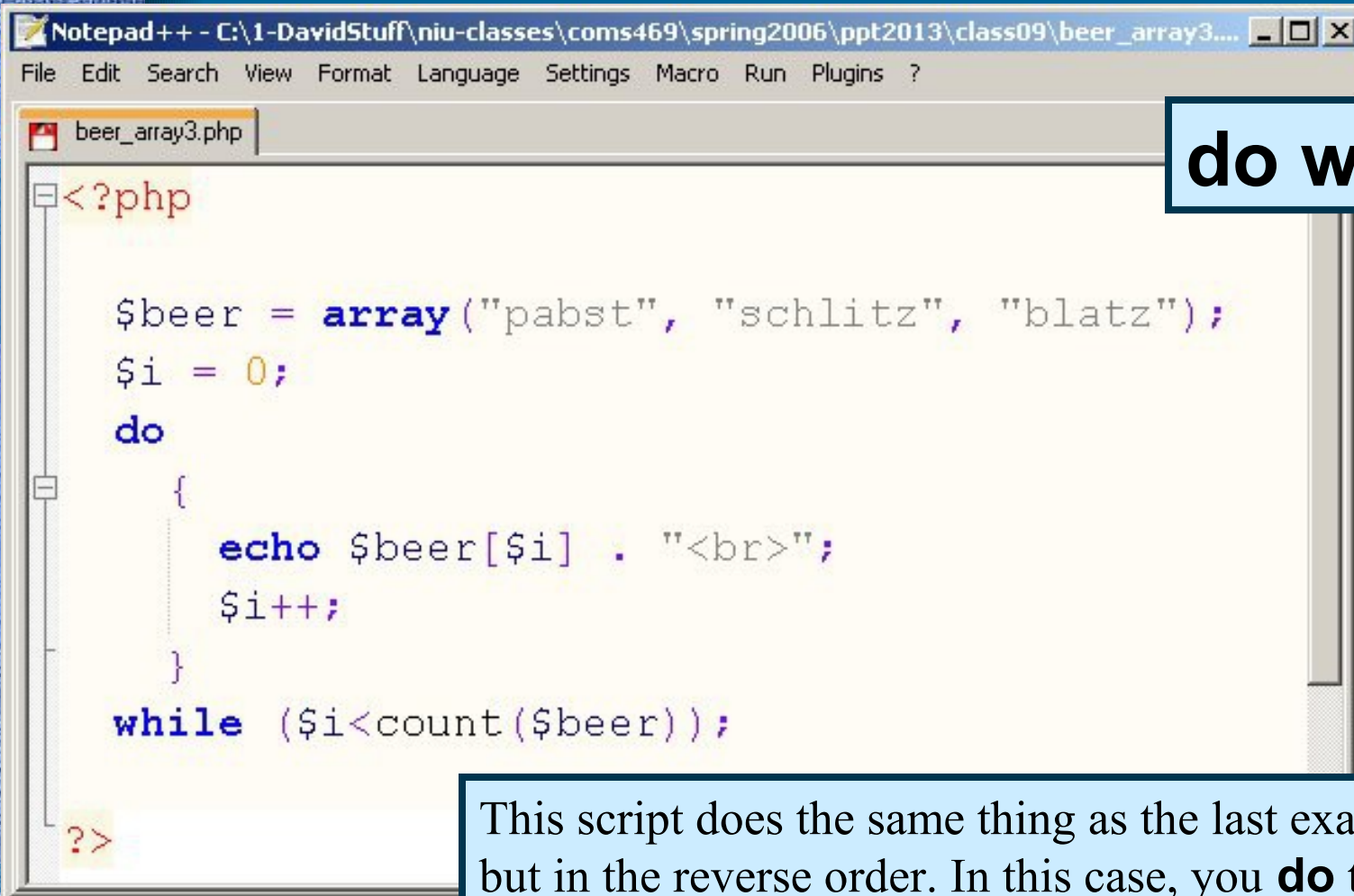
$beer = array("pabst", "schlitz", "blatz");
$i = 0;
while ($i<count($beer))
{
    echo $beer[$i] . "<br>";
    $i++;
}

?>
```

while()

This script does the same thing as the `for()` loop. When you use `while()`, however, the initialization and increment steps are performed by other parts of the script.

This script “says” as long as `$i` is less than the total number of items in the array, then `echo` the elements of the array and increment `$i`



```
<?php

$beer = array("pabst", "schlitz", "blatz");
$i = 0;
do
{
    echo $beer[$i] . "<br>";
    $i++;
}
while ($i < count($beer));

?>
```

do while()

This script does the same thing as the last example but in the reverse order. In this case, you **do** the echo and incrementing as long as (**while**) the condition—`$i` is less than the total number of elements in the array—is true.

Arrays

- Array Functions
 - PHP has several built-in array manipulation functions
 - Features: search, compare, sort, organize and convert array data

Common PHP array Functions

FUNCTION	DESCRIPTION
print_r()	Looks inside an array and displays the values
explode()	Splits a string into array elements
implode()	Joins array elements into a string
range()	Generates a number range as an array
min()	Finds the smallest value in an array
max()	Finds the largest value in an array
shuffle()	Randomly rearranges the sequence of elements in an array
array_slice()	Extracts a segment of an array
array_shift()	Removes an element from the beginning of an array
array_unshift()	Add an element to the beginning of an array
array_pop()	Removes an element from the end of an array
array_push()	Add an element to the end of an array
array_unique()	Removes duplicated elements from an array

Common PHP array Functions

FUNCTION	DESCRIPTION
<code>array_reverse()</code>	Reverses the sequence of elements in an array
<code>array_merge()</code>	Combines two or more arrays
<code>array_intersect()</code>	Calculates the common elements between two or more arrays
<code>array_diff()</code>	Calculates the difference between two arrays
<code>in_array()</code>	Checks if a particular value exists in an array
<code>array_key_exists()</code>	Checks if a particular key exists in an array
<code>sort()</code>	Sorts an array
<code>asort()</code>	Sorts an associative array by value
<code>ksort()</code>	Sorts an associative array by key
<code>rsort()</code>	Reverse-sorts an array
<code>krsort()</code>	Reverse-sorts an associative array by key
<code>arsort()</code>	Reverse-sorts an associative array by value

Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\array_functions.php

File Edit Search View Format Language Settings Macro Run Plugins ?

array_functions.php

```
<?php
$beer = array("pabst","schlitz","blatz","old style");

print_r($beer);

$string_beer = implode( " / " , $beer);
echo "<p> $string_beer ";

$two_beers = array_slice($beer, 2,2);
echo "<p>";
print_r($two_beers);

array_shift($beer);
echo "<p>";
print_r($beer);

array_push($beer, "miller lite");
echo "<p>";
print_r($beer);

sort($beer);
echo "<p>";
print_r($beer);

?>
```

Mozilla Firefox

File Edit View History Bookmarks Tools Help

Novell GroupWise (David GUNKEL)

http://www2.comm.niu...rray_functions.php

www2.comm.niu.edu/sp2013/jarada/array_functions.php

Search Results

Array ([0] => pabst [1] => schlitz [2] => blatz [3] => old style)

pabst / schlitz / blatz / old style

Array ([0] => blatz [1] => old style)

Array ([0] => schlitz [1] => blatz [2] => old style)

Array ([0] => schlitz [1] => blatz [2] => old style [3] => miller lite)

Array ([0] => blatz [1] => miller lite [2] => old style [3] => schlitz)

Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\array_functions.php

File Edit Search View Format Language Settings Macro Run Plugins ?

array_functions.php

```
<?php
$beer = array("pabst","schlitz","blatz","old style");

print_r($beer);

$string_beer = implode(" / ", $beer);
echo "<p> $string_beer ";

$two_beers = array($beer,$string_beer);
print_r($two_beers);

array_shift($beer);
echo "<p>";
print_r($beer);

array_push($beer, "miller lite");
echo "<p>";
print_r($beer);

sort($beer);
echo "<p>";
print_r($beer);

?>
```

Mozilla Firefox

File Edit View History Bookmarks Tools Help

Novell GroupWise (David GUNKEL)

http://www2.comm.niu...rray_functions.php

www2.comm.niu.edu/sp2013/jarada/array_functions.php

Search Results

Array ([0] => pabst [1] => schlitz [2] => blatz [3] => old style)

pabst / schlitz / blatz / old style

Array ([0] => blatz [1] => old style)

Array ([0] => schlitz [1] => blatz [2] => old style)

Array ([0] => schlitz [1] => blatz [2] => old style [3] => miller lite)

Array ([0] => blatz [1] => miller lite [2] => old style [3] => schlitz)

Print_r()

Display the contents of the Array

Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\array_functions.php

File Edit Search View Format Language Settings Macro Run Plugins ?

array_functions.php

```
<?php
$beer = array("pabst","schlitz","blatz","old style");

print_r($beer);

$string_beer = implode(" / ", $beer);
echo "<p> $string_beer ";

$two_beers = array_slice($beer, 2, 2);

print_r($beer);

sort($beer);
echo "<p>";
print_r($beer);

?>
```

Mozilla Firefox

File Edit View History Bookmarks Tools Help

Novell GroupWise (David GUNKEL)

http://www2.comm.niu...rray_functions.php

www2.comm.niu.edu/sp2013/jarada/array_functions.php

Search Results

Array ([0] => pabst [1] => schlitz [2] => blatz [3] => old style)

pabst / schlitz / blatz / old style

Array ([1] => old style)

Array ([1] => blatz [2] => old style)

Array ([1] => blatz [2] => old style [3] => miller lite)

Array ([0] => blatz [1] => miller lite [2] => old style [3] => schlitz)

implode()

Display the contents of the Array as a string value.

Two arguments: the separator "/" and the array name \$beer

Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\array_functions.php

File Edit Search View Format Language Settings Macro Run Plugins ?

array_functions.php

```
<?php
$beer = array("pabst","schlitz","blatz","old style");

print_r($beer);

$string_beer = implode(" / ", $beer);
echo "<p> $string_beer ";

$two_beers = array_slice($beer, 2,2);
echo "<p>";
print_r($two_beers);
```

```
array_shift($beer);
```

array_slice()

Cut an array into smaller parts.

Three arguments: the array name `$beer`, the offset of the slice, and the number of elements to include in the new array.

```
echo "<p> ";
print_r($beer);
```

```
?>
```

Mozilla Firefox

File Edit View History Bookmarks Tools Help

Novell GroupWise (David GUNKEL)

http://www2.comm.niu...rray_functions.php

www2.comm.niu.edu/sp2013/jarada/array_functions.php

Search Results

Array ([0] => pabst [1] => schlitz [2] => blatz [3] => old style)

pabst / schlitz / blatz / old style

Array ([0] => blatz [1] => old style)

itz [1] => blatz [2] => old style)

itz [1] => blatz [2] => old style [3] => miller lite)

z [1] => miller lite [2] => old style [3] => schlitz)

Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\array_functions.php

File Edit Search View Format Language Settings Macro Run Plugins ?

array_functions.php

```
<?php
$beer = array("pabst","schlitz","blatz","old style");
```

```
print_r($beer);
```

array_shift()

Remove an element from the beginning of the array

```
print_r($two_beers);
```

```
array_shift($beer);
```

```
echo "<p>";
```

```
print_r($beer);
```

```
array_push($beer, "miller lite");
```

```
echo "<p>";
```

```
print_r($beer);
```

```
sort($beer);
```

```
echo "<p>";
```

```
print_r($beer);
```

```
?>
```

Mozilla Firefox

File Edit View History Bookmarks Tools Help

Novell GroupWise (David GUNKEL)

http://www2.comn.ni...rray_functions.php

www2.comn.niu.edu/sp2013/jarada/array_functions.php

Search Results

Array ([0] => pabst [1] => schlitz [2] => blatz [3] => old style)

schlitz / blatz / old style

[0] => blatz [1] => old style)

Array ([0] => schlitz [1] => blatz [2] => old style)

Array ([0] => schlitz [1] => blatz [2] => old style [3] => miller lite)

Array ([0] => blatz [1] => miller lite [2] => old style [3] => schlitz)

```
<?php
$beer = array("pabst","schlitz","blatz","old style");

print_r($beer);

$string_beer = implode(" / ", $beer);
echo "<p> $string_beer </p>";

array_push($beer, "miller lite");
echo "<p>";
print_r($beer);

sort($beer);
echo "<p>";
print_r($beer);

?>
```

Array ([0] => pabst [1] => schlitz [2] => blatz [3] => old style)

pabst / schlitz / blatz / old style

array_push()

Add an element to the end of the array.

Two arguments: the array name \$beer and the name of the new element "miller lite"

old style)

blatz [2] => old style)

Array ([0] => schlitz [1] => blatz [2] => old style [3] => miller lite)

Array ([0] => blatz [1] => miller lite [2] => old style [3] => schlitz)

Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\array_functions.php

File Edit Search View Format Language Settings Macro Run Plugins ?

array_functions.php

```
<?php
$beer = array("pabst","schlitz","blatz","old style");
```

```
print_r($beer);
```

```
$string_beer = implode(" / ", $beer);
echo "<p> $string_beer ";
```

```
$two_beers = array_slice($beer, 2,2);
echo "<p>";
print_r($two_beers);
```

array_sort()

Arrange array elements in alphabetical order

```
print_r($beer);
```

```
sort($beer);
echo "<p>";
print_r($beer);
```

?>

Mozilla Firefox

File Edit View History Bookmarks Tools Help

Novell GroupWise (David GUNKEL)

http://www2.comn.ni...rray_functions.php

www2.comn.niu.edu/sp2013/jarada/array_functions.php

Search Results

Array ([0] => pabst [1] => schlitz [2] => blatz [3] => old style)

pabst / schlitz / blatz / old style

Array ([0] => blatz [1] => old style)

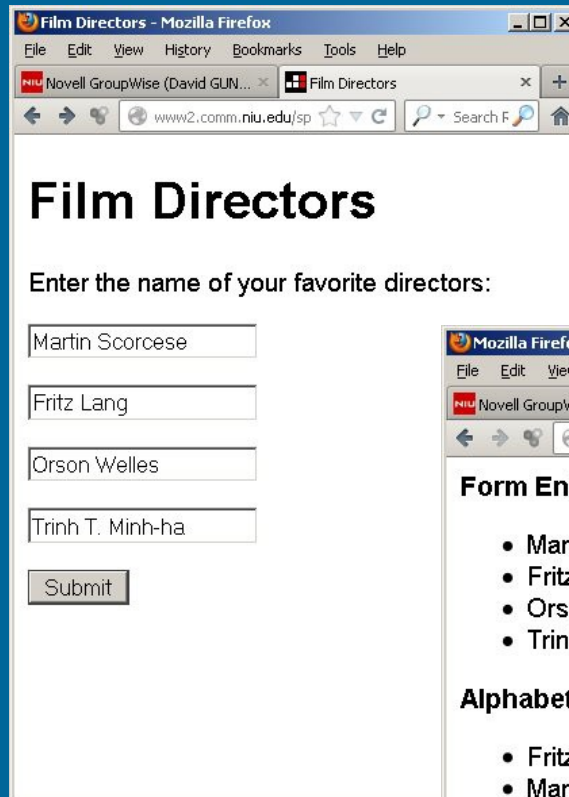
Array ([0] => schlitz [1] => blatz [2] => old style)

Array ([0] => schlitz [1] => blatz [2] => old style [3] => miller lite)

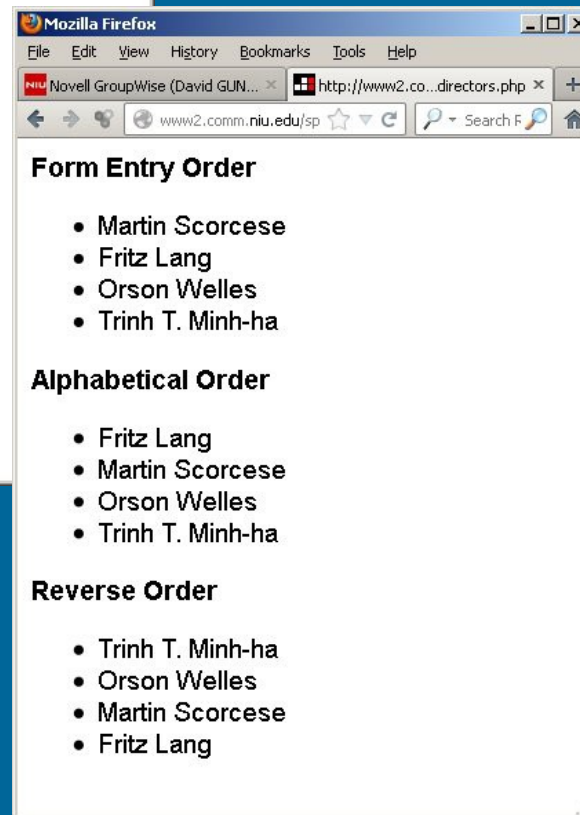
Array ([0] => blatz [1] => miller lite [2] => old style [3] => schlitz)

Exercise

- Use an **array** to collect values from an HTML form
- Iterate over the array to process input data
- Use array functions to organize the array elements



The screenshot shows a Mozilla Firefox browser window with the title 'Film Directors - Mozilla Firefox'. The address bar shows 'www2.comm.niu.edu/sp'. The page has a title 'Film Directors' and a prompt 'Enter the name of your favorite directors:'. Below the prompt are four text input fields containing the names 'Martin Scorsese', 'Fritz Lang', 'Orson Welles', and 'Trinh T. Minh-ha'. At the bottom of the form is a 'Submit' button.



The screenshot shows the same Mozilla Firefox browser window after the form has been submitted. The page title is 'Form Entry Order'. It displays three bulleted lists of the director names entered in the form:

- Form Entry Order**
 - Martin Scorsese
 - Fritz Lang
 - Orson Welles
 - Trinh T. Minh-ha
- Alphabetical Order**
 - Fritz Lang
 - Martin Scorsese
 - Orson Welles
 - Trinh T. Minh-ha
- Reverse Order**
 - Trinh T. Minh-ha
 - Orson Welles
 - Martin Scorsese
 - Fritz Lang

```
Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\directors.html
File Edit Search View Format Language Settings
directors.html directors.php
<html>
  <head>
    <title>Film Directors</title>
  </head>
  <body>
    <form method="post" action="directors.php">
      <h1>Film Directors</h1>
      Enter the name of your favorite directors:
      <p><input type="text" name="directors[]"></p>
      <p><input type="text" name="directors[]"></p>
      <p><input type="text" name="directors[]"></p>
      <p><input type="text" name="directors[]"></p>
      <p><input type="submit" value="Submit"></p>
    </form>
  </body>
</html>
```

HTML form with four input fields named "directors[]" This array will collect and hold the values entered on the form. Action of the form is **directors.php**

Notepad++ - C:\1-DavidStuff\niu-classes\coms469\spring2006\ppt2013\class09\directors.html

File Edit Search View Format Language Settings Macro Run Plugins ?

directors.html directors.php

```
<html>
  <head>
    <title>Film Directors</title>
  </head>
  <body>
    <form method="post" action="directors.php">
      <h1>Film Directors</h1>
      Enter the name of your favorite directors:
      <p><input type="text" name="directors[]"></p>

      <p><input type="text" name="directors[]"></p>

      <p><input type="text" name="directors[]"></p>

      <p><input type="text" name="directors[]"></p>

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

  </body>
</html>
```

directors.html


```
directors.php x directors.html x php_info.php x
1 <?php
2
3 $directors = $_POST['directors'];
4
5 echo "<b>Form Entry Order</b> <ul>";
6     for ($i=0; $i<count($directors); $i++)
7     {
8         echo "<li> $directors[$i] </li>";
9     }
10
11 echo "</ul> <p><b>Alphabetical Order</b> <ul>";
12     sort($directors);
13     foreach ($directors as $d)
14     {
15         echo "<li> $d </li>";
16     }
17
18 echo "</ul> <p><b>Reverse Order</b> <ul>";
19     rsort($directors);
20     $i=0;
21     while ($i<count($directors))
22     {
23         echo "<li> $directors[$i] </li>";
24         $i++;
25     }
26 echo "</ul>";
27 ?>
```

Initialize the variable \$directors

```
directors.php x directors.html x php_info.php x
1 <?php
2
3 $directors = $_POST['directors'];
4
5 echo "<b>Form Entry Order</b> <ul>";
6     for ($i=0; $i<count($directors); $i++)
7     {
8         echo "<li> $directors[$i] </li>";
9     }
10
11 echo "</ul> <p><b>Alphabetical Order</b> <ul>";
12     sort($directors);
13     foreach ($directors as $d)
14     {
15         echo "<li> $d </li>";
16     }
17
18 echo "</ul> <p><b>Reverse Order</b> <ul>";
19     rsort($directors);
20     $i=0;
21     while ($i<count($directors))
22     {
23         echo "<li> $directors[$i] </li>";
24         $i++;
25     }
26 echo "</ul>";
27 ?>
```

Use the for() function to iterate over the array \$directors and echo the values in an unordered list

```
directors.php x directors.html x php_info.php x
1 <?php
2
3 $directors = $_POST['directors'];
4
5 echo "<b>Form Entry Order</b> <ul>";
6   for ($i=0; $i<count($directors); $i++)
7   {
8       echo "<li> $directors[$i] </li>";
9   }
10
11 echo "</ul> <p><b>Alphabetical Order</b> <ul>";
12   sort($directors);
13   foreach ($directors as $d)
14   {
15       echo "<li> $d </li>";
16   }
17
18 echo "</ul> <p><b>Reverse Order</b> <ul>";
19   rsort($directors);
20   $i=0;
21   while ($i<count($directors))
22   {
23       echo "<li> $directors[$i] </li>";
24       $i++;
25   }
26 echo "</ul>";
27 ?>
```

Use the `sort()` function to order the values of the array –alphabetical A-Z

Use the `foreach()` function to iterate over the array `$directors` and echo the values in an unordered list ``

```
directors.php x directors.html x php_info.php x
1 <?php
2
3 $directors = $_POST['directors'];
4
5 echo "<b>Form Entry Order</b> <ul>";
6     for ($i=0; $i<count($directors); $i++)
7     {
8         echo "<li> $directors[$i] </li>";
9     }
10
11 echo "</ul> <p><b>Alphabetical Order</b> <ul>";
12     sort($directors);
13     foreach ($directors as $d)
14     {
15         echo "<li> $d </li>";
16     }
17
18 echo "</ul> <p><b>Reverse Order</b> <ul>";
19     rsort($directors);
20     $i=0;
21     while ($i<count($directors))
22     {
23         echo "<li> $directors[$i] </li>";
24         $i++;
25     }
26 echo "</ul>";
27 ?>
```

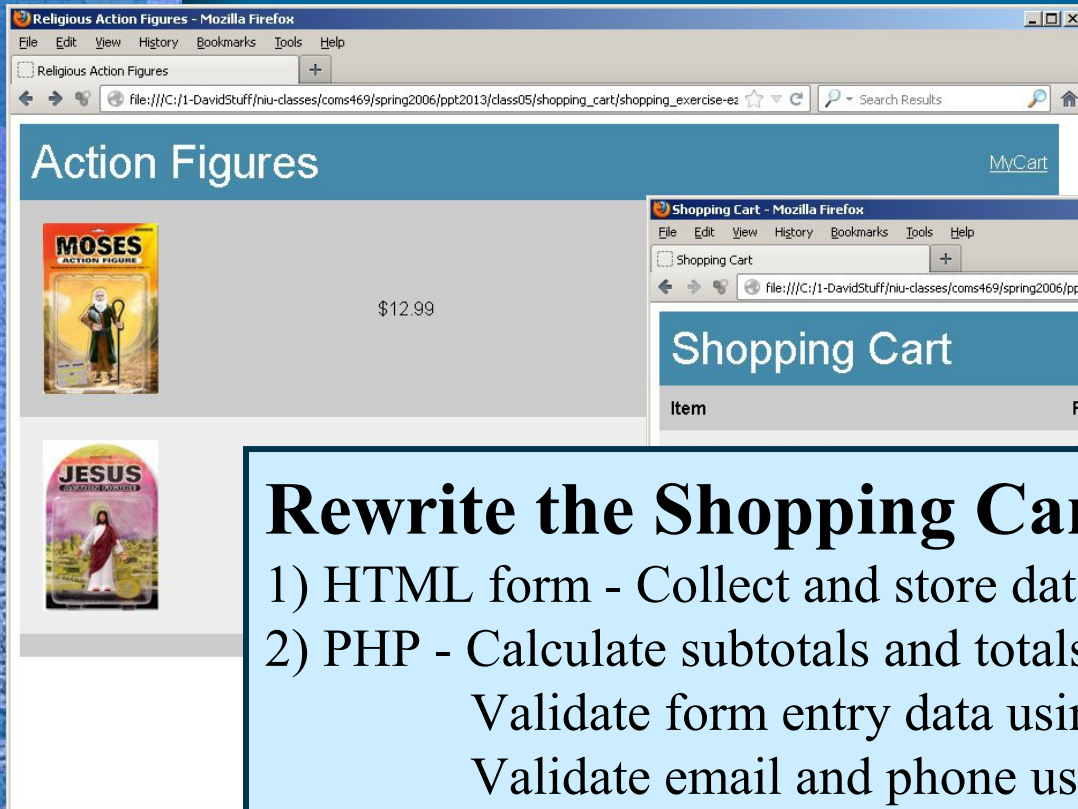
Use the `rsort()` function to order the values of the array – reverse alphabetical Z-A

Use a while statement to iterate over the array `$directors` and echo the values in an unordered list ``


```
1 <?php
2
3 $directors = $_POST['directors'];
4
5 echo "<b>Form Entry Order</b> <ul>";
6     for ($i=0; $i<count($directors); $i++)
7     {
8         echo "<li> $directors[$i] </li>";
9     }
10
11 echo "</ul> <p><b>Alphabetical Order</b> <ul>";
12     sort($directors);
13     foreach ($directors as $d)
14     {
15         echo "<li> $d </li>";
16     }
17
18 echo "</ul> <p><b>Reverse Order</b> <ul>";
19     rsort($directors);
20     $i=0;
21     while ($i<count($directors))
22     {
23         echo "<li> $directors[$i] </li>";
24         $i++;
25     }
26 echo "</ul>";
27 ?>
```

directors.php

Take Home Exercise



Rewrite the Shopping Cart with PHP

- 1) HTML form - Collect and store data in variables or an array
- 2) PHP - Calculate subtotals and totals using numeric functions
Validate form entry data using conditional statements
Validate email and phone using string functions
Provide users with feedback (i.e. errors or a receipt)
Email results to yourself by using `mail()`

Preview

- Ch. 5 - Using Functions and Classes (pp. 121-154)
- Ch. 6 - Working with Files and Directories (pp. 159-183)