PRINT VERSION

PHP Forms and User Input

If you type some characters in the text field below, and click the "Submit" button, the browser will send your input to a page called `action_page.php`. The page will show you the received input.

```
<form name="input" action="action_page.php" method="get">
  Username:
  <input type="text" name="firstname">
  <input type="submit" value="Submit">
</form>
```

How it looks in a browser: Username: [ ] Submit

The `$_GET` and `$_POST` variables are used to retrieve user information from forms. The example below shows the web input is saved in the variable `$_POST['string']` by using a POST method, where the `strtolower(string)` function converts a string to lowercase. The `print_r(mixed $expression)` function displays information about a variable in a way readable by humans.

HTML

```
<form method="post" action="1_1.php">
 $string = <input type="text"
   name="string" size="25"
   value="This is only a Test.">
 <input type="submit" name="act"
   value="Check">
</form>
```

1_1.php

```
<html><body>
<?php
 print_r ( $_POST );
 echo strtolower(
   $_POST['string'] );
?>
</body></html>
```

$string = This is only a Test. Check

Form Validation

User input should be validated whenever possible. Client side validation is faster, and will reduce server load. However, any site that gets enough traffic to worry about server resources, may also need to worry about site security. You should always use server side validation if the form accesses a database.

Slide 3.15: PHP user-defined functions
Slide 4.2: PHP variables $_POST and $_GET
Home

PRINT VERSION