Developing Web Applications with PHP

Agenda

- – Introduction
- – PHP Language Basics
- – Built-in Functions
- – Tricks and Tips
- – HTML
- – Embedding PHP in HTML
- – MySQL
- – Resources

Introduction

- PHP: Hypertext Preprocessor.
- Server-side scripting languages for creating dynamic Web pages.
- Platform-independent (implementations exist for all major UNIX, Linux, and Windows operation systems).
- Also supports a large number of DB's including MySQL.
- PHP was written specifically for interacting with the Web.
- PHP code is embedded directly into XHTML document.
- PHP script file names usually ends with .php
- It's open-source

PHP Language Basics

- **The Script Tags**
  - All PHP code is contained in one of several script tags:

    -
      ```
      <?
      // Some code
      ?>
      ```

    -
      ```
       <?php
      // Some code here
      ?>
      ```

PHP Language Basics

- **Hello World!: An Example**
  - There is more than one way to do it
    - <?php echo “Hello World!”; ?>

    -
      ```
      <?php
          $greeting = “Hello World!”
          printf(“%s”, $greeting);
      php?>
      ```

PHP Language Basics

- **Hello World!: An Example (cont.)**

  -
    ```
    <script language="PHP">
        $hello = "Hello";
        $world = "World!";
        print $hello . $world
    </script>
    ```

PHP Language Basics

- **Constants, Data Types and Variables**
  - Constants define a string or numeric value
  - Constants do not begin with a dollar sign
  - Examples:
    - define(“COMPANY”, “TIFR”);
    - define(“YELLOW”, “#FFFF00”);
    - define(“PI”, 3.14);
    - define(“NL”, “<br>\n”);

PHP Language Basics

- **Constants, Data Types and Variables**
  - Using a constant
    - print(“Company name: “ . COMPANY . NL);

PHP Language Basics

- **Constants, Data Types and Variables**
  - – Data types
    - Integers, doubles and strings
      - – isValid = true; // Boolean
      - – 25 // Integer
      - – 3.14 // Double
      - – ‘Four’ // String
      - – “Total value” // Another string

PHP Language Basics

- **Constants, Data Types and Variables**
  - Data types
    - Strings and type conversion
      - $street = 123;
      - $street = $street . “ Main Street”;
      - $city = ‘Naperville’;
        $state = ‘IL’;
      - $address = $street;
      - $address = $address . NL . “$city, $state”;
      - $number = $address + 1;   // $number equals 124

PHP Language Basics

- **Constants, Data Types and Variables**
  - Data types
    - Arrays
      - syntax
        - $arr = array("foo" => "bar", 12 => true);
      - same as
        - $arr[“foo”] = “bar”;
        - $arr[12] = true;

PHP Language Basics

- **Constants, Data Types and Variables**
  - Arrays (cont.)
    -
      ```
      <?php
          $arr = array("somearray" => array(6 => 5, 13 => 9,
                                            "a" => 42));

          echo $arr["somearray"][6];     // 5
          echo $arr["somearray"][13];   // 9
          echo $arr["somearray"]["a"];  // 42
      ?>
      ```

PHP Language Basics

- **Constants, Data Types and Variables**
  - Objects
  - An *object* is a data type that allows for the storage of not only data but also information on how to process that data. The data elements stored within an object are referred to as its *properties*, also sometimes called the *attributes* of the object. The information, or code, describing how to process the data compromises what are called the *methods* of the object.

PHP Language Basics

- **Constants, Data Types and Variables**
  - Operators
    - Contains all of the operators like in C
  - Statements
    - if, if/elseif
    - Switch/case
    - for, while, and do/while loops
    - Include and require statements for code reuse

Built-in Functions

- **What comes In the box?**
  - – Array Manipulator Functions
    - sort, merge, push, pop, slice, splice, keys, count
  - – Date and Time Functions
    - getdate, mkdate, date, gettimeofday, localtime, strtotime, time

Built-in Functions

- **What comes In the box?**
  - – Directory Functions
    - Platform independent
  - – Error Handling Functions
    - Recover from warnings and errors
  - – Filesystem Functions
    - Access flat files
    - Check directory, link, and file status information
    - Copy, delete, and rename files

Built-in Functions

- **What comes In the box?**
  - – IMAP Functions
    - Manipulate mail boxes via the IMAP protocol
  - – LDAP Functions
    - Works with most LDAP servers
  - – Mail Functions
    - mail($recipient, $subject, $message)

Built-in Functions

- **What comes In the box?**
  - Database Functions
    - MySQL
    - Oracle
    - PostgreSQL
    - SQL Server
  - MING
    - Macromedia Flash
  - PDF
    - Create/manipulate PDF files dynamically

Built-in Functions

- **What comes In the box?**
  - POSIX Functions
    - Manipulate process information
  - Regular Expression Functions
    - Uses POSIX regex
  - Semaphore and Socket Functions
    - Available only on Unix
  - Session Management Functions

Tricks and Tips

- **Coding**
  - – Prototype your web pages first
    - Separate the design of the site from the coding
  - – Turn repetitive code into functions
    - Makes for more maintainable and reusable code
  - – Turn grunt code into functions
    - Database access, configuration file access

Tricks and Tips

- **Debugging**
  - – Feature: PHP is not a strongly typed language
    - Variables can be created anywhere in your code
    - Typos in variable names will cause stuff to happen
    - Basic debugging technique is to simply turn on error reporting. Another slightly more advanced technique involves using print statements, which can help pinpoint more elusive bugs by displaying what is actually going onto the screen.

Tricks and Tips

- **Development Tools**
  - Color coding editors
    - vim, Emacs

HTML

```
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

HTML Forms

HTML forms are used to pass data to a server.

A form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more.

The <form> tag is used to create an HTML form:

```
<form>
 First name: <input type="text" name="firstname" />
 <br>
 Last name: <input type="text" name="lastname" />
 </form>
```

HTML Tables

Tables are defined with the <table> tag.

A table is divided into rows (with the <tr> tag), and each row is divided into data cells (with the <td> tag). td stands for "table data," and holds the content of a data cell. A <td> tag can contain text, links, images, lists, forms, other tables, etc.

Table Example

```
<table border="1">
 <tr>
 <td>row 1, cell 1</td>
 <td>row 1, cell 2</td>
 </tr>
 <tr>
 <td>row 2, cell 1</td>
 <td>row 2, cell 2</td>
 </tr>
 </table>
```

HTML Styles

How to Use Styles

When a browser reads a style sheet, it will format the document according to it.

There are three ways of inserting a style sheet:

1. External style sheet
2. Internal style sheet
3. Inline styles

External Style Sheet

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing one file. Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the <head> section:

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css" />
</head>
```

HTML Styles

Internal Style Sheet

An internal style sheet can be used if one single document has a unique style. Internal styles are defined in the <head> section of an HTML page, by using the <style> tag, like this:

```
<head>
    <style type="text/css">
    body {background-color:yellow}
    p {color:blue}
    </style>
    </head>
```

Inline Styles

An inline style can be used if a unique style is to be applied to one single occurrence of an element.

To use inline styles, use the style attribute in the relevant tag. The style attribute can contain any CSS property. The example below shows how to change the text color and the left margin of a paragraph:

```
<p style="color:blue;margin-left:20px">This is a paragraph.</p>
```

HTML Scripts

The HTML script Element

The <script> tag is used to define a client-side script, such as a JavaScript.

The script element either contains scripting statements or it points to an external script file through the src attribute.

Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.

The script below writes Hello World! to the HTML output:

Example

```
<script type="text/javascript">
document.write("Hello World!")
</script>
```

Embedding PHP in HTML

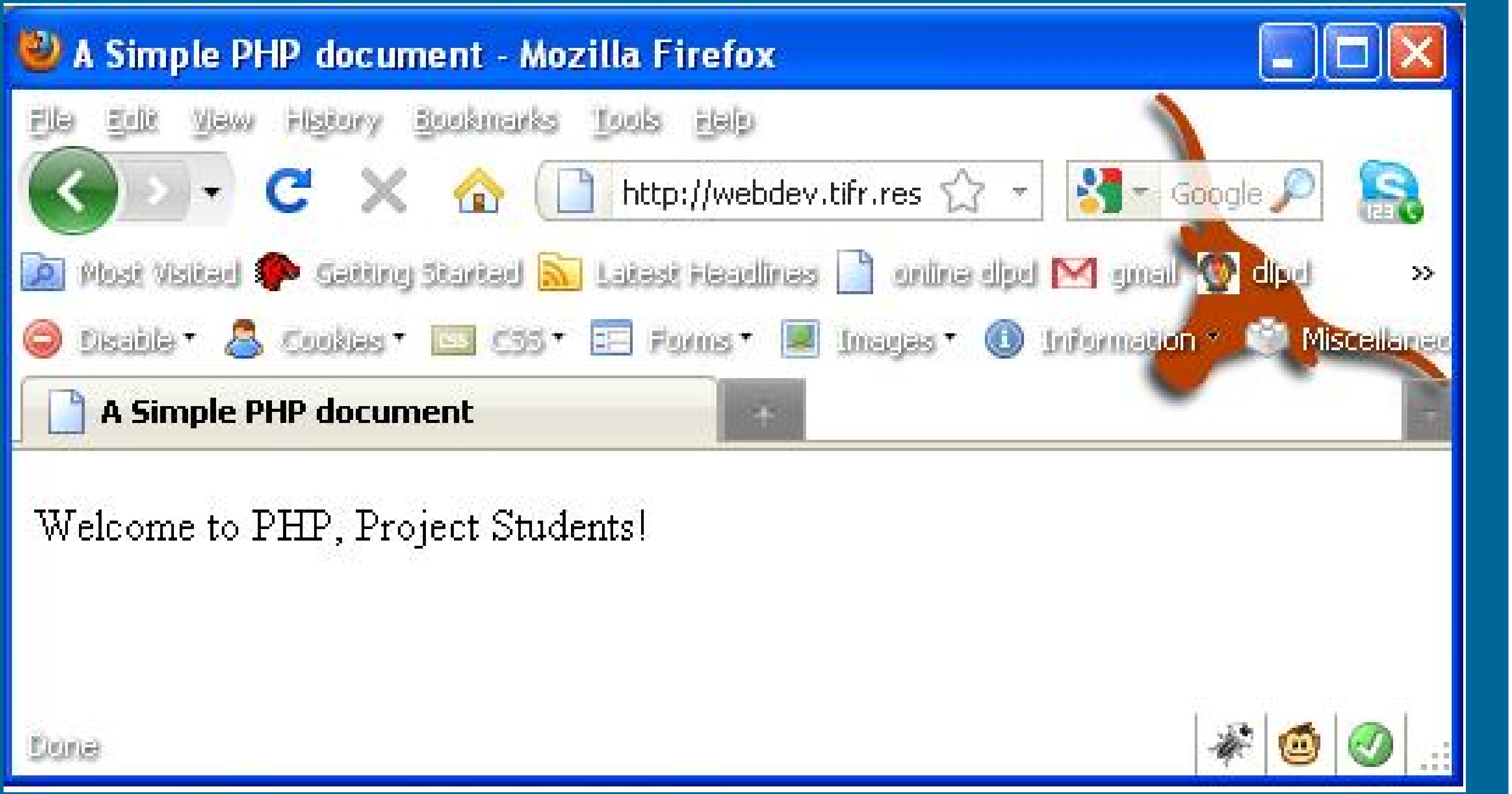
```
 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">

<?php
  $name = "Project Students"; //declaration
?>

<html LANG="en">
  <head>
          <title> A Simple PHP document </title>
  </head>


  <body>
  <p>
  Welcome to PHP, <?php print $name; ?>!
  </p>
  </body>
</html>
```

Embedding PHP in HTML


A Simple PHP document - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://webdev.tifr.res
Google
Most Visited Getting Started Latest Headlines online dlpd gmail dlpd
Disable Cookies CSS Forms Images Information Miscellaneo
A Simple PHP document
Done

Welcome to PHP, Project Students!

MySQL

- Open Source database server
  - Runs on many platforms (Unix & Windows)
- Networked server
  - You can find *clients* that provide a GUI.
  - phpMyAdmin
- Great for small to medium-sized applications

MySQL

DATA MANAGEMENT

- SHOW DATABASES;
- USE databaseName;
- SHOW TABLES;
- DESCRIBE table;
- SELECT * FROM table;
- CREATE DATABASE databaseName;
- DROP DATABASE databaseName;
- CREATE TABLE tableName(name1 type1, name2 type2, ...);
- DROP TABLE tableName;
- INSERT INTO TABLE VALUES(value1, value2, ...);
- SELECT field1, field2, ... FROM tableName;
- SELECT * INTO OUTFILE ‘/home/skr.txt' FROM skr;
- LOAD DATA INFILE /home/skr.txt INTO TABLE skr;

MySQL

Opening a MySQL database

```
$username=“fred”;  $password=“newdb”;
$database=“web”;
mysql_connect("localhost",$username,$pass
  word);

@mysql_select_db($database) or die(
  "Unable to select database");
```

Assumes that the machine running the server is “localhost”

MySQL

Submitting a query to the server

```
$query = "SELECT uid from users WHERE username =
   ‘fred’";

$res = mysql_query($query);

// no result - no user so return false

if (! $res) {
   … no result (error!)
}
```

MySQL

Accessing

- `mysql_numrows($res)` : number of rows in the result.
- `mysql_result($res,$index,$field)`
  - Returns a single column value
  - $index is the row
  - $field is the field name (column name)

Example

```
$res = mysql_query(“SELECT * FROM users”);
if (! $res) { …handle error…}

$numrows = mysql_numrows($res)

for ($i=0;$i<$numrows;$i++) {
   $name = mysql_result($res,$i,”username”);
   $pass = mysql_result($res,$i,”password”);
   … Do something with $name and $password…
}
```

Resources

PHP Downloads and Online Documentation

– www.php.net

Book Reference

-PHP and MySQL, Hugh E. Williams & David Lane, O'Reilly.

Community

– www.phpbuilder.com : articles on PHP, discussion forums

Newsgroups

-comp.lang.php

Resources

PHP Downloads and Online Documentation

- http://www.w3schools.com/

MySQL Downloads and Online Documentation

- http://www.mysql.com/

THANK YOU