

PHP

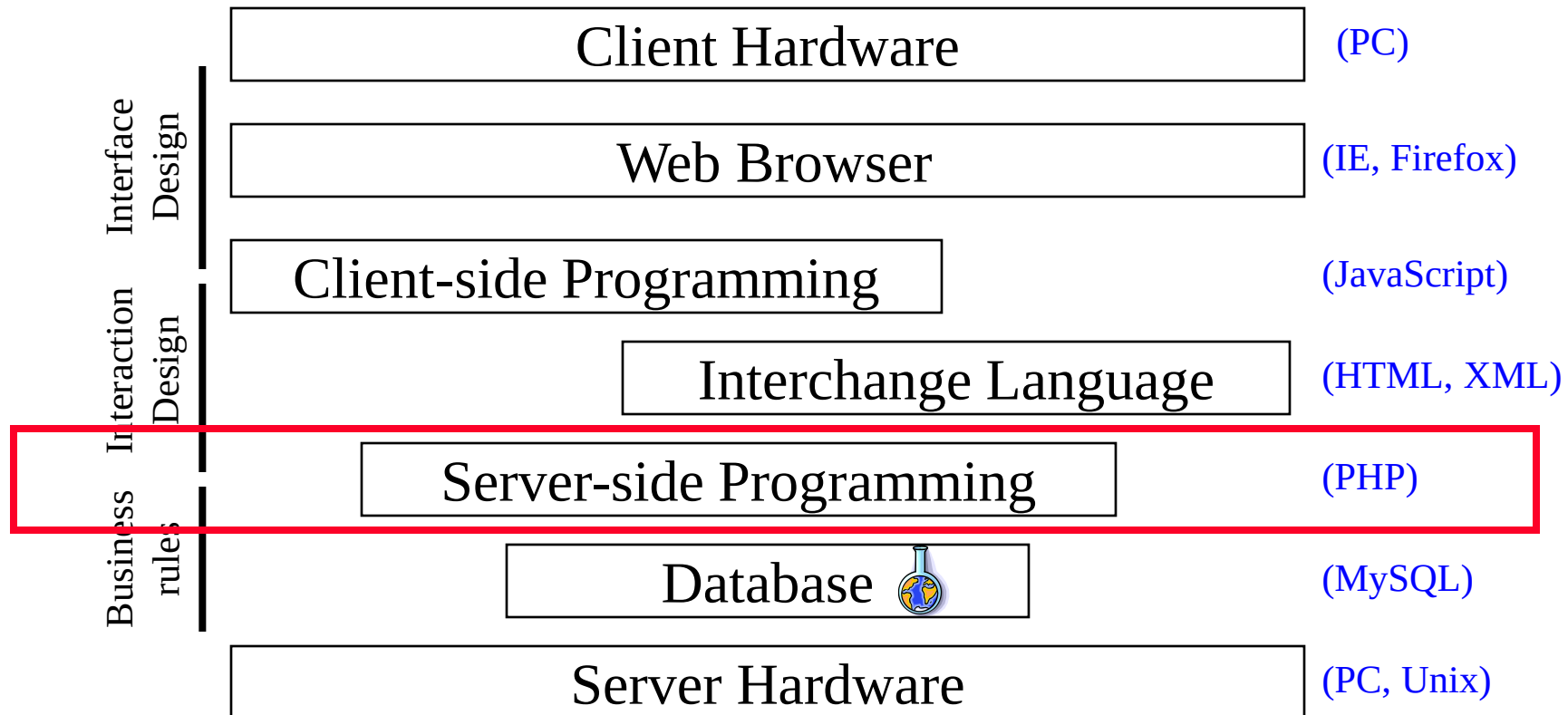
Week 8

INFM 603

Agenda

- Questions
- PHP
- Drupal
- Project Plan

- Relational normalization
- Structured programming
- Software patterns
- Object-oriented design
- Functional decomposition



Thinking About PHP

- Local vs. Web-server-based display
- HTML as an indirect display mechanism
- “View Source” for debugging
- Procedural perspective (vs. object-oriented)

Making PHP

----- HTML stuff -----

<?php

----- PHP stuff -----

?>

----- HTML stuff -----

http://---URL stuff---/xxxxxx.php

Programming Skills Hierarchy

- Reusing code [run the book's programs]
- Understanding patterns [read the book]
- Applying patterns [modify programs]
- Coding without patterns [programming]
- Recognizing new patterns

Some Things to Pay Attention To

Syntax

- How layout helps reading
- How variables are named
- How strings are used
- How input is obtained
- How output is created

Structured Programming

- How things are nested
- How arrays are used

Modular Programming

- Functional decomposition
- How functions are invoked
- How arguments work
- How scope is managed
- How errors are handled
- How results are passed

Variables

- All variable names starts with a \$
 - Case sensitive (assume everything could be!)
- Variables can hold any scalar value
 - Number (integer, float)
 - String (double quotes, \ escape character)
 - TRUE, FLASE
 - NULL
- Need not be declared, automatically cast

Operators in PHP

- Arithmetic operators

+ - * /

- Logical operators

< <= == != >= > && || !

- String concatenation operator

. **Different from JavaScript!**

Statements in PHP

- Sequential

`{...; ...;...;}`

Semicolons are **required** at the end of every statement

- Conditional

`if (3==i) {...} else {...}`

- Loop

`foreach ($array as $key => $value) {...}`

`while ($row=mysql_fetch_array(...)) {...}`

`For ($i=0; $i<10; $i++) {...}`

- Braces are optional around a single statement

Arrays in PHP

- A set of key-element pairs

```
$days = array("Jan"->31, "Feb"=>28, ...);
```

```
$months = explode("/", "Jan/Feb/Mar/.../Dec");
```

```
$_POST
```

- Each element is accessed by the key
 - `$months[0];`
 - `{ $days["Jan"] }`
- PHP unifies arrays and hashtables
 - Elements may be different types

Functions in PHP

- Declaration

```
function multiply($a, $b=3){return $a*$b;}
```

- Invoking a method

```
$b = multiply($b, 7);
```

- All variables in a function have only local scope
 - Unless declared as global in the function

Using PHP with (X)HTML Forms

```
<form action="formResponseDemo.php", method="post">
    email: <input type="text", name="email", value="<?php echo $email ?>", size=30 />
    <input type="radio", name="sure", value="yes" /> Yes
    <input type="radio", name="sure", value="no" /> No
    <input type="submit", name="submit", value="Submit" />
    <input type="hidden", name="submitted", value="TRUE" />
</form>
```

```
if (isset($_POST["submitted"])) {
    echo "Your email address is $email.";
} else {
    echo "Error: page reached without proper form submission!";
}
```

Connecting PHP to MySQL

- On XAMPP:

```
$dbc=mysql_connect ('localhost', 'userid', 'password');
```

- On unix:

```
$dbc=mysql_connect('/:export/software/otal/mysql/run/mysqld.sock',  
                    'userid', 'password');
```

```
<?php # Script 8.1 - mysql_connect.php
```

```
// Set the database access information as constants.
```

```
DEFINE ('DB_USER', 'tester');
```

```
DEFINE ('DB_PASSWORD', 'tester');
```

```
DEFINE ('DB_HOST', 'localhost');
```

```
DEFINE ('DB_NAME', 'sitename');
```

```
// Make the connection.
```

```
$dbc = @mysql_connect (DB_HOST, DB_USER, DB_PASSWORD) OR die ('Could not connect to  
MySQL: ' . mysql_error() );
```

```
// Select the database.
```

```
@mysql_select_db (DB_NAME) OR die ('Could not select the database: ' . mysql_error() );
```

```
// Create a function for escaping the data.
```

```
function escape_data ($data) {
```

```
    // Address Magic Quotes.
```

```
    if (ini_get('magic_quotes_gpc')) {
```

```
        $data = stripslashes($data);
```

```
    }
```

```
    // Check for mysql_real_escape_string() support.
```

```
    if (function_exists('mysql_real_escape_string')) {
```

```
        global $dbc; // Need the connection.
```

```
        $data = mysql_real_escape_string (trim($data), $dbc);
```

```
    } else {
```

```
        $data = mysql_escape_string (trim($data));
```

```
    }
```

```
    // Return the escaped value.
```

```
    return $data;
```

```
} // End of function.
```

```
?>
```

```
<?php # Script 9.15 - login.php (7th version after Scripts 9.1, 9.3, 9.6, 9.10. 9.13 & 9.14)
// Send NOTHING to the Web browser prior to the session_start() line!
// Check if the form has been submitted.

if (isset($_POST['submitted'])) {
    require_once ('../mysql_connect.php'); // Connect to the db.
    $errors = array(); // Initialize error array.

    // Check for an email address.
    if (empty($_POST['email'])) {
        $errors[] = 'You forgot to enter your email address.';
    } else {
        $e = escape_data($_POST['email']);
    }

    // Check for a password.
    if (empty($_POST['password'])) {
        $errors[] = 'You forgot to enter your password.';
    } else {
        $p = escape_data($_POST['password']);
    }
}
```



```

if (empty($errors)) { // If everything's OK.
    /* Retrieve the user_id and first_name for that email/password combination. */
    $query = "SELECT user_id, first_name FROM users WHERE email='$e' AND password=SHA('$p')";
    $result = @mysql_query ($query); // Run the query.
    $row = mysql_fetch_array ($result, MYSQL_NUM); // Return a record, if applicable.
    if ($row) { // A record was pulled from the database.
        // Set the session data & redirect.
        session_name ('YourVisitID');
        session_start();
        $_SESSION['user_id'] = $row[0];
        $_SESSION['first_name'] = $row[1];
        $_SESSION['agent'] = md5($_SERVER['HTTP_USER_AGENT']);
        // Redirect the user to the loggedin.php page.
        // Start defining the URL.
        $url = 'http://' . $_SERVER['HTTP_HOST'] . dirname($_SERVER['PHP_SELF']);
        // Check for a trailing slash.
        if ((substr($url, -1) == '/') OR (substr($url, -1) == '\\') ) {
            $url = substr ($url, 0, -1); // Chop off the slash.
        }
        // Add the page.
        $url .= '/loggedin.php';
        header("Location: $url");
        exit(); // Quit the script.
    } else { // No record matched the query.
        $errors[] = 'The email address and password entered do not match those on file.'; // Public message.
        $errors[] = mysql_error() . '<br /><br />Query: ' . $query; // Debugging message.
    }
} // End of if (empty($errors)) IF.
mysql_close(); // Close the database connection.
} else { // Form has not been submitted.
    $errors = NULL;
} // End of the main Submit conditional.

```

```
// Begin the page now.
$page_title = 'Login';
include ('./includes/header.html');

if (!empty($errors)) { // Print any error messages.
    echo '<h1 id="mainhead">Error!</h1>
    <p class="error">The following error(s) occurred:<br />';
    foreach ($errors as $msg) { // Print each error.
        echo " - $msg<br />\n";
    }
    echo '</p><p>Please try again.</p>';
}

// Create the form.
?>

<h2>Login</h2>
<form action="login.php" method="post">
    <p>Email Address: <input type="text" name="email" size="20" maxlength="40" /> </p>
    <p>Password: <input type="password" name="password" size="20" maxlength="20" /></p>
    <p><input type="submit" name="submit" value="Login" /></p>
    <input type="hidden" name="submitted" value="TRUE" />
</form>

<?php
include ('./includes/footer.html');
?>
```

Agenda

- Questions
- PHP
 - Drupal
- Project Plan

Content Management Systems

- Database to store content
 - Also stores access control data and parameters
- PHP to control user experience
 - Reads database, generates HTML
 - “Canned” settings provide standard behaviors
- XHTML to convey user experience
- Allows limited interactivity
 - Most user actions require a server response
 - JavaScript may be used for form validation

Installing Drupal

- Download and install XAMPP
 - Add c:\xampp\mysql\bin to your path
- Download and install Drupal version 6.x
 - Configure for local use (“first time user guide”)
 - Ignore SMTP error messages
- Configure your site
 - Add some “splash page” content
 - Set user permissions

Drupal's Use of MySQL

```
USE drupal;
```

```
SHOW TABLES;
```

```
SELECT * FROM users;
```

```
SELECT * FROM nodes;
```

```
SELECT * FROM node_revisions;
```

Modifying Drupal

- Work with what's there
 - Content
 - Configuration
- Download a distribution profile
- Edit the CSS files
- Edit the PHP code
- Edit the database contents

Agenda

- Questions
- PHP
- Drupal

➤ Project Plan

What are Requirements?

- Attributes
 - Appearance
 - Concepts (represented by data)
- Behavior
 - What it does
 - How you control it
 - How you observe the results

Who Sets the Requirements?

- People who need the task done (customers)
- People that will operate the system (users)
- People who use the system's outputs
- People who provide the system's inputs
- Whoever pays for it (requirements commissioner)

The Requirements Interview

- Focus the discussion on the task
 - Look for entities that are mentioned
- Discuss the system's most important effects
 - Displays, reports, data storage
 - Learn where the system's inputs come from
 - People, stored data, devices, ...
- Note any data that is mentioned
 - Try to understand the structure of the data
- Shoot for the big picture, not every detail

The Project Plan

- One-page contract
 - Between developer and requirements commissioner
- Goal The problem to be solved
- Product What you plan to deliver
- Scope Available time and personnel
- Roles What you expect each other to do