CS 6324: Information Security

PHP

PHP: Hypertext Preprocessor

- PHP, or PHP: Hypertext Preprocessor, has become one of the most popular server-side scripting languages for creating dynamic web pages.
- PHP code is embedded directly into XHTML documents, though these script segments are interpreted by a server before being delivered to the client.
- PHP script file names end with .php
- In PHP, code is inserted between the scripting delimiters <?php and ?>. PHP code can be placed anywhere in XHTML markup, as long as the code is enclosed in these delimiters.

PHP basics

- Variables are preceded by a $ and are created the first time they are encountered.
  - Variable names in PHP are case sensitive. Failure to use the proper mixture of cases to refer to a variable will result in a logic error, since the script will create a new variable for any name it doesn't recognize as a previously used variable.
- PHP statements terminate with a semicolon (;).
- Single-line comments begin with two forward slashes (//) or a pound sign (#). Text to the right of the delimiter is ignored by the interpreter. Multiline comments begin with delimiter /* and end with delimiter */.
- When a variable is encountered inside a double-quoted ("") string, PHP interpolates the variable. In other words, PHP inserts the variable's value where the variable name appears in the string.
- All operations requiring PHP interpolation execute on the server before the XHTML document is sent to the client.
- PHP variables are loosely typed—they can contain different types of data at different times.

Outline

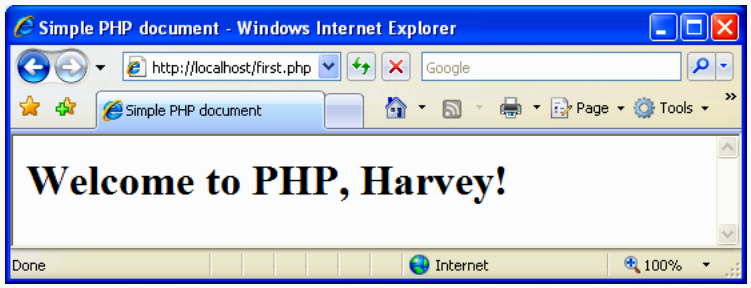
first.php

```
1  <?php print( '<?xml version = "1.0" encoding = "utf-8"?>' ) ?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 23.1: first.php -->
6  <!-- Simple PHP program. -->
7  <html xmlns = "http://www.w3.org/1999/xhtml">
8  <?php
9     $name = "Harvey"; // declaration and initialization
10 ?><!-- end PHP script -->
11    <head>
12       <title>Using PHP document</title>
13    </head>
14    <body style = "font-size: 2em">
15       <p>
16          <strong>
17             <!-- print variable name's value -->
18             Welcome to PHP, <?php print( "$name" ); ?>!
19          </strong>
20       </p>
21    </body>
22 </html>
```

Delimiters enclosing PHP script

Declares and initializes a PHP variable

Interpolates the variable so that its value will be output to the XHTML document



PHP basics

- Superglobal arrays are associative arrays predefined by PHP that hold variables acquired from user input, the environment or the web server and are accessible in any variable scope.
- The arrays `$_GET` and `$_POST` retrieve information sent to the server by HTTP `get` and `post` requests, respectively.
- Using `method = "post"` appends form data to the browser request that contains the protocol and the requested resource's URL. Scripts located on the web server's machine can access the form data sent as part of the request.
- Function `extract` creates a variable/value pair corresponding to each key/value pair in the associative array passed as an argument.
- Function `die` terminates script execution. The function's optional argument is a string, which is printed as the script exits.

PHP Basics

Variable name	Description
`$_SERVER`	Data about the currently running server.
`$_ENV`	Data about the client's environment.
`$_GET`	Data sent to the server by a `get` request.
`$_POST`	Data sent to the server by a `post` request.
`$_COOKIE`	Data contained in cookies on the client's computer.
`$GLOBALS`	Array containing all global variables.

PHP and MySQL

- Function `mysql_connect` connects to the MySQL database. It takes three arguments—the server's hostname, a username and a password, and returns a database handle—a representation of PHP's connection to the database, or `false` if the connection fails.
- Function `mysql_select_db` specifies the database to be queried, and returns a bool indicating whether or not it was successful.
- To query the database, we call function `mysql_query`, specifying the query string and the database to query. This returns a resource containing the result of the query, or false if the query fails. It can also execute SQL statements such as `INSERT` or `DELETE` that do not return results.
- Function `mysql_error` returns any error strings from the database.
- `mysql_close` closes the connection to the database specified in its argument.

Outline

database.php

(2 of 3)

```
29          // Connect to MySQL
30          if ( !( $database = mysql_connect( "localhost",
31             "iw3htp4", "iw3htp4" ) ) )
32             die( "Could not connect to database </body></html>" );
33
34          // open Products database
35          if ( !mysql_select_db( "products", $database ) )
36             die( "Could not open products database </body></html>" );
37
38          // query Products database
39          if ( !( $result = mysql_query( $query, $database ) ) )
40          {
41             print( "Could not execute query! <br />" );
42             die( mysql_error() . "</body></html>" );
43          } // end if
44
45          mysql_close( $database );
46       ?><!-- end PHP script -->
47       <h3>Search Results</h3>
48       <table>
49          <?php
50             // fetch each record in result set
51             for ( $counter = 0; $row = mysql_fetch_row( $result );
52                $counter++ )
53             {
54                // build table to display results
55                print( "<tr>" );
56
```

Connects to database using server hostname `localhost` and username and password "`iw3htp4`"

Returns any error strings from the database

Specifies `products` as the database to be queried

Closes the connection to the database

Queries `$database` with `$query`

Returns an array with the values for each column of the current row in `$result`

PHP and Cookies

- A cookie is a text file that a website stores on a client's computer to maintain information about the client during and between browsing sessions.
- A server can access only the cookies that it has placed on the client.
- Function setcookie takes the name of the cookie to be set as the first argument, followed by the value to be stored in the cookie. The optional third argument indicates the expiration date of the cookie. A cookie without a third argument is known as a session cookie, while one with an expiration date is a persistent cookie. If only the name argument is passed to function setcookie, the cookie is deleted from the client's computer.
- Cookies defined in function setcookie are sent to the client at the same time as the information in the HTTP header; therefore, it needs to be called before any XHTML is printed.
- The current time is returned by function time.