PHP

MIT 6.470, IAP 2010

Yafim Landa (landa@mit.edu)

LAMP

- We'll use Linux, Apache, MySQL, and PHP for this course
- There are alternatives
  - Windows with IIS and ASP
  - Java with Tomcat
  - Other database systems like PostgreSQL, or non-SQL databases

Why PHP?

- A very simple and straightforward syntax
  - PHP is really well documented: http://php.net/
  - Type the name of a function you want to look up at the end of the URL, and you'll be sent directly to the relevant help page, for example: http://php.net/json_encode
- Tight integration with MySQL (and lots of other database systems)
- Well established and created specifically for the web
  - Used by Facebook, Wikipedia, YouTube, Digg, and plenty of others
- Does lots of cool things like encryption, image manipulation, email, file upload, and so on with ease
- Object oriented as of PHP5: http://php.net/manual/en/language.oop5.php
- Convenient type system for the web

What Does PHP Do?

- Generates pages that the user can see
  - Retrieves any information from the database or from other sources
  - Displays an HTML page with **dynamic content**
  - Writes data back to the database or performs other operations
- Generates data for your AJAX requests

- Regular HTML pages can change only through the use of Javascript
  - Very superficial (without the use of AJAX)
- HTML can be rendered dynamically using PHP
  - The page can change depending on the time of day, the contents of the database, the user's input, etc.

- We already know how to make HTML pages that show static content
- To add dynamic content, we can simply **embed PHP code within an HTML page using a special tag**
- This embedded code is executed on the server before it is sent to the client and looks like regular HTML to the client

ymdb / Good Will Hunting

http://[redacted].php?films_idx=767

Most Visited Android Classes

ymdb / Good Will Hunting

YMDB
our film companion

News / Profile / Log out

Yafim this movie

I love this movie! Undo.

I used to like this movie in high school, but now I just find it to be annoying. :(

Modify

Friends this movie

Friends who like this movie:

Vazrik Chiloyan · Sol B · Trevor J

Get this movie

Find this film on NetFlix.

Buy on Amazon.com.

FILM search

Good Will Hunting 1997

Miramax (USA)
$10,000,000
Director: Gus Van Sant
Producer: Lawrence Bender
Editor: Pietro Scalia
Writer: Matt Damon
Ben Affleck
Lang: English

(3.25)

Starring

Matt Damon
Robin Williams
Ben Affleck
Minnie Driver
Stellan Skarsgård

Trailer

Who Fav'd It

Yafim Landa · Vazrik Chiloyan · Sol B · Trevor J

Reel Reviews

Write a review.

RATE: Okay

Ben Baumgold
Sol, you're swell lookin!

Sol B
Swell movie.

Yafim Landa
I used to like this movie in high school, but now I just find it to be annoying. :(

Guest
I'm still a guest here!

Vazrik Chiloyan
It is awesome.

News Feed

- Top 25 Movies of 2009: Honorable Mentions and Movies #21 – 25 – RopeofSilicon.com
- Matt Damon's Good Will Hunting 2? – E! Online (blog)
- Damon and Freeman Bleed for 'Invictus' – Hollywood Today Newsmagazine
- A plainer view of our past | Philadelphia Inquirer | 12/08/2009 – Philadelphia Inquirer
- New DVD/Blu-ray Discs arrive in time for holidays – phillyBurbs.com
- Hollywood glamor, at thrift store price – Boston Globe
- REVIEW: 'The Road' Casts a Spell, Never Lets Go – Big Hollywood (blog)

Done

Language Syntax

- http://6.470.scripts.mit.edu/2009/
- Double quotes vs. single quotes
  - If `$var` is set to "6.470"
    - `echo "This is $var"` will output *This is 6.470*
    - `echo 'This is $var'` will output *This is $var*
- Associative arrays
  - `$var ['foo'] = 'hello, world';`
  - `foreach ($var as $key => $value) {`
    - `$var` is an (associative) array
    - Makes `$key` (a key) and `$value` (the value stored at that key) available on each loop iteration
- === does a comparison with type
- http://www.php.net/manual/en/langref.php

Superglobals

- PHP has several special variables that are global everywhere
- All of these are associative arrays
  - $_SERVER – server and execution environment information
    - $_SERVER['PHP_SELF'] is useful for the form action attribute
  - $_GET – variables passed through the URL
    - http://some.server.com/index.php?**param=value**
  - $_POST – variables passed through the HTTP POST method
  - $_REQUEST – both GET and POST combined
  - $_FILES – files uploaded through HTTP POST
  - $_COOKIE – contents of HTTP cookies
  - $_SESSION – an associative array of session variables

Error Handling

- To debug your code, insert the following two lines at the beginning of your script:

```
ini_set('display_errors',1);
error_reporting(E_ALL);
```

Example: First dynamic content

- Demo: http://landa.scripts.mit.edu/6.470/examples/example1/index.php
- Code: http://landa.scripts.mit.edu/6.470/examples/example1/code.html

Example: Superglobals

- Demo: http://landa.scripts.mit.edu/6.470/examples/example2/index.php
- Code: http://landa.scripts.mit.edu/6.470/examples/example2/code.html

Input

- We can get input from various sources
  - GET and POST request variables, from the user
    - Includes input from forms
    - Access using $_GET, $_POST, or $_REQUEST superglobal associative arrays
  - File uploads from the user
  - Changing data in the database
  - Other websites and APIs
    - Twitter, Google, Facebook, and so on

Working With MySQL

```php
<?php

$dbh = mysql_connect('sql.mit.edu', 'username', 'password')
       or die('Could not connect: ' . mysql_error() . '<br />');

mysql_select_db("username+dbname") or die("No database selected.");

?>
```

- Put the database connection code in a separate file (database.php)
- `include_once 'database.php'`

- `$sql = mysql_query($query)`
  - $query is the MySQL query string (like "SELECT * FROM comments")
  - Returns a *resource* and stores it in $sql
  - You can step over the rows in the resource one by one by writing

    `$row = mysql_fetch_object($sql)` or

    `$row = mysql_fetch_array($sql)`
  - Often used in a while loop
    - `while($row = mysql_fetch_array($sql)) {`
    - Loops until all of the rows have been examined
  - See comments.php in Feedback example

Session Management (Logging In)

- Sessions allow you to store data that persists between PHP pages
  - This means that we can create an account system
- Store the user's account data in sessions
  - Using $_SESSION superglobal
- Must call `session_start()` at the beginning of each page to use sessions

Example: Feedback

- http://landa.scripts.mit.edu/6.470/feedback/index.php
- Topics
  - Sessions
    - MIT certificates
  - Working with MySQL
  - $_POST

Example: Outputting JSON

- http://landa.scripts.mit.edu/6.470/feedback/comments.php?limit=10
- Useful for feeding data to AJAX calls
- Topics
  - Use a limit using `$_GET['limit']`
  - Enabling JSON using php.ini
  - Error reporting
- Displays all of the comments in the database in JSON format
  - Examine the JSON output using http://jsonformatter.curiousconcept.com/

Date and Time Functions

- The easiest thing to do is to convert everything into and work with seconds since January 1, 1970
- `date($format [, $timestamp])`

  formats the timestamp (used to display the date in a human readable format)
- `time()` gets the current time measured in seconds since January 1, 1970
- `strtotime($time [, $now])` converts a string like “next Monday” into seconds since January 1, 1970
- Use MySQL’s functions `FROM_UNIXTIME` and `UNIX_TIMESTAMP` to convert between PHP and MySQL date formats
- http://us.php.net/manual/en/ref.datetime.php

Input Filtering

- It's usually best not to trust external data
  - Can invoke various vulnerabilities, HTML code, and other things that you may not want
- As a first line of defense you should
  - `strip_tags($input)` to remove HTML tags
  - `addslashes($input)` before writing data to the database and `stripslashes($input)` after retrieving it back
  - `mysql_real_escape_string($input)` for SQL queries
- More about this tomorrow