INFO I-CE9224: Introduction to PHP Programming

Session 1
May 30, 2012

Resources

http://davehauenstein.com/nyu/INFO1-CE9224-2012-Summer

Username: nyuscps
Password: $nyuscps$

Class 1 Agenda

- Introductions
- Review the Syllabus
- Web Technology Primer
- Getting Set Up on Windows and Mac
- PHP Language Basics Part 1

Dave Hauenstein

- Experience building web applications for ~9 years
- PHP, Python, Java, C++
- MySQL, MongoDb, SQLServer, Oracle
- Built and contributed to several Open Source Software projects
- BA in Economics from Rutgers University

BUSINESS INSIDER

- Platform Architect at Business Insider
- ~17 Million Uniques hits per Month
- ~125 Million Page views per Month

Who are you?

This class is as much an introduction to programming as it is an introduction to the PHP programming language.

The only assumption I'm going to make is that you know and can write HTML.

Syllabus Review

Markup Languages

HTML, XML

Presentational
Descriptive

HTML

Presentational + Descriptive

```
<div id="content-container">
    <header>
        <hgroup class="ir">
            <h1>Newtons</h1>
            <h2>Made with Real Fruit</h2>
        </hgroup>
    </header>
    <section id="page-content" class="ir">
        <h1>When it comes to cookies, we keep it real.</h1>
        <p>
            Ever since the first Fig Newtons were made over
            a century ago, etc...
        </p>
    </section>
    <footer>
        <p><small>&copy; 2011 Kraft Foods Company.</small></p>
    </footer>
</div>
```

HTML: HyperText Markup Language

XHTML and HTML5 (and HTML4)

- application/xhtml+xml
  - XML Namespace is required
  - XML Declaration is required
  - Parsed as XML
- text/html
  - XML namespace must not be provided
  - No XML declaration can be provided
  - Multiple HTML parsing types

XML: Extensible Markup Language

Descriptive

```
<?xml version="1.0"?>
<application xmlns="http://wadl.dev.java.net/2009/02">
  <method name="GET" id="ItemSearch">
    <request>
      <param name="Version" style="query" fixed="2005-07-26"/>
      <param name="Operation" style="query" fixed="ItemSearch"/>
      <param name="SearchIndex" style="query">
         <option value="Books"/>
         <option value="DVD"/>
         <option value="Music"/>
      </param>
    </request>
    <response>
      <representation mediaType="text/xml" element="SearchResponse"/>
    </response>
  </method>
</application>
```

XML

- XML Namespaces (XMLNS) - Allow documents to contain elements and attributes of different vocabularies without naming collisions.

- XPath - A syntax for identifying components (attributes, elements, etc...) of an XML document.

- XSLT - An XML based language used for transforming XML documents to another format like HTML. See this example.

Presentation

- CSS: Cascading Style Sheets

CSS

- Separates presentation (look and formatting) from markup

- Used to apply style to markup languages; most commonly HTML and XHTML

```
*{
    margin: 0;
    padding: 0;
}
body{
    font-family: arial;
    font-size: 12px;
    text-align: center;
    background: #000;
}
a:hover{
    text-decoration: none;
}
#container{
    margin: 60px auto 0;
    padding: 15px 25px;
    text-align: left;
    width: 705px;
    background: #171717;
    font-size: 52px;
    text-align: justify;
    line-height: 55px;
    border: solid 40px #111;
}
#relevant{
    margin: 40px auto 0;
    color: #666;
    width: 705px;
}
```

Presentation

- CSS: Cascading Style Sheets
- JavaScript

JavaScript

- A scripting language used for DOM manipulation, communicating with a server, client-side validation, etc...

- Object Oriented - prototype based - classes aren't present, inheritence is based on cloning objects.

- Additionally implemented as a server-side language using Node.js

```
if(typeof(arguments[0]) == 'object') {
    var evt = arguments[0];
    evt.data = evt.data || {};

    url = evt.data.url || location.href;
    trackers = evt.data.trackers || tracking.defaults.trackers;

// expect parameters passed to the function directly
} else {
    url = arguments[0] || location.href;
    trackers = arguments[1] || tracking.defaults.trackers;
}

if((trackers & TRACKER_GOOGLE_ANALYTICS) && window._gaq) {
    if(window.console && tracking.defaults.debug) {
        console.log("Page view tracked: url '%s'", url);
    }

    window._gaq.push(['_trackPageview', url]);
}
```

Presentation

- CSS: Cascading Style Sheets
- JavaScript
- AJAX: Asynchronous Javascript and XML

AJAX: Asynchronous JavaScript and XML

- Uses XMLHttpRequest Object to send HTTP or HTTPS requests directly to a web server and load the response back into the script.
- Allows partial updates to the webpage without a full refresh.
- XML isn't always used, most of the time it's JSON.
- Requests can be made synchronously as well.
- Revolutionized the way that web applications work. See an Example.

Protocols

- TCP/IP: Transmission Control Protocol/ Internet Protocol
- HTTP: Hypertext Transfer Protocol
- FTP: File Transfer Protocol
- SFTP: Secure File Transfer Protocol

A protocol is a system of digital message formats and rules for exchanging those messages in or between computing systems.

-Wikipedia

Protocols

- TCP/IP: Transmission Control Protocol/ Internet Protocol
- HTTP: Hypertext Transfer Protocol
- FTP: File Transfer Protocol
- SFTP: Secure File Transfer Protocol

HTTP: Hypertext Transfer Protocol

- Requst-Response communication method (as opposed to one-way communication like publish subscribe).
- Browsing the web is an example of request-response.
- Client-Server model: Client(s) independently make requests to a server over a computer network; server responds to each clients' request.
- Stateless protocol: Each request is completely independent of each other. One request has no knowledge of a previous or concurrent request.
- State or session is application logic outside of HTTP.
- Defines methods (GET, POST, etc...), status codes (404, 500, etc...), content negotiation, caching, and more...

HTTP Request

```
GET / HTTP/1.1
User-Agent: curl/7.21.4 (universal-apple-darwin11.0) libcurl/7.21.4 OpenSSL/0.9.8r zlib/1.2.5
Host: www.google.com
Accept: */*
```

HTTP Response

```
HTTP/1.1 200 OK
Date: Thu, 08 Mar 2012 05:29:40 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
Set-Cookie: PREF=ID=caf5278a4242164d:FF=0:TM=1331...
Set-Cookie: NID=57=EMQB4w_njAJSGpjqGTNxYTP4HKp816...
Server: gws
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked

<!doctype html>
<html>
    etc...
</html>
```

MySQL

- Is an Open Source (and commercially supported) RDBMS (Relational Database Management System)
- Uses SQL (Structured Query Language) for data management (retrieving, inserting, updating, deleting, etc...)
- Extremely popular especially with PHP based web applications

MySQL

```
CREATE TABLE `users` (
    `id` INT(11) UNSIGNED NOT NULL auto_increment,
    `first_name` VARCHAR(255) NOT NULL,
    `last_name` VARCHAR(255) NOT NULL,
    `email` VARCHAR(255) NOT NULL,
    `registration_date` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `validated` ENUM('Y','N') DEFAULT 'N',
    PRIMARY KEY (`id`)
) ENGINE = INNODB;

INSERT INTO `users` (`first_name`, `last_name`, `email`)
    VALUES ('Dave', 'Hauenstein', 'dah16@nyu.edu');

SELECT * FROM `users`;

+----+------------+------------+----------------+---------------------+-----------+
| id | first_name | last_name  | email          | registration_date   | validated |
+----+------------+------------+----------------+---------------------+-----------+
|  1 | Dave       | Hauenstein | dah16@nyu.edu | 2012-03-07 21:11:21 | N         |
+----+------------+------------+----------------+---------------------+-----------+
```

Apache

- Apache is an open source, fully featured, and extremely popular HTTP web server
- Processes an incoming HTTP request, calls PHP interpreter to generate the response, returns response to client

Getting Set Up on Windows and Mac

Windows: WAMP

Mac: MAMP

PHP Language Basics

Part I

PHP: Hypertext Preprocessor

- Create dynamic (as opposed to static) web pages
- Create interactive web pages that respond to users' input
- Is an interpreted (as opposed to compiled) language
- Dynamically typed (as opposed to statically typed)
- Has a command line component as well

Interpreted vs. Compiled

- Conversion to machine level instructions happens at run time
- Interpreter must be present on system, no executable is available
- Generally slower
- PHP, JavaScript, Java (JIT)

- Static or ahead-of-time
- Compiled programs are already in the format native to the hardware and OS it's running on
- Have to recompile to see changes
- Generally faster
- C, C++

Variables

A variable is a container that holds a certain value.

PHP Variables

Consider the statement:

echo “Hello, Dave. How are you?”;

This is a general purpose statement,
but what if we wanted to make it a
little bit more extensible?

PHP Variables

```
$name = “Dave”;
echo “Hello, $name. How are you?”;
```

We’ve made the script more extensible by adding a variable where the name used to be, so it can change at any time, w/out changing the statement.

Naming Variables

- Variables always begin with a dollar sign ($)
- The first character after the $ must be a letter or an underscore (_)
- The remaining characters may be letters, numbers, or underscores
- Variables are case-sensitive

Valid vs Invalid

- $some_variable
- $_someVariable
- $_someVariable2
- $_123
- $1badvariable
- $another-bad-var

PHP Warnings

Warnings will be raised, but the program will continue to execute if a variable is not first declared.

```php
<?php

echo $name;
```

Browser Output

(!) Notice: Undefined variable: name in /var/app/com.dave.dev/www/test.php on line 7

Call Stack

#	Time	Memory	Function	Location
1	0.0002	633896	{main}( )	../test.php:0

PHP Warnings

Instead, first declare and initialize the variable; no warnings present

```php
<?php

$name = "Dave";

var_dump($name);
```

Browser Output

```
string 'Dave' (length=4)
```

Data Types

- All variables are of a certain type.
- Think of a variable as a bucket with a label.
- What's inside the bucket is the variables *value*.
- The label on the bucket is the *type*.
- For example, a variable might contain the value "person". It's type would be *string*.

PHP Data Types

Scalar Data Type	Description	Example
Integer	A whole number	15
Float	A floating-point number	8.23
String	A series of characters	“Hello, World!”
Boolean	Represents either true or false	TRUE

Compound Data Type	Description
Array	An ordered map (contains names or numbers mapped to values
Object	A type that may contain properties and methods

PHP Data Types cont...

Special Data Type	Description
Resource	Contains a reference to an external resource, such as a file or database
null	May only contain null as a value, meaning the variable explicitly does not contain any value

Type Checking

```php
<?php

$intVar = 12;
is_int($intVar); // returns (boolean) true

$floatVar = 3.14;
is_float($floatVar); // returns (boolean) true

$stringVar = 'Hello, class!';
is_string($stringVar); // returns (boolean) true

echo gettype($intVar); // returns (string) "integer"
echo gettype($floatVar); // returns (string) "double"
echo gettype($stringVar); // returns (string) "string"
```

PHP Type Checking

- is_int (*value*)
- is_float (*value*)
- is_string (*value*)
- is_bool (*value*)
- is_array (*value*)
- is_object (*value*)
- is_resource (*value*)
- is_null (*value*)
- gettype (*value*)

Changing a Variable's Data Type

```php
<?php

$floatVar = 3.14;
is_float($floatVar); // returns (boolean) true

settype($floatVar, 'integer');

echo $floatVar; // prints out (integer) 3
is_float($floatVar); // returns (boolean) false
is_int($floatVar); // returns (boolean) true

echo gettype($floatVar); // returns (string) "integer"
```

Type Casting

```php
<?php
$floatVar = 3.14;
is_float($floatVar); // returns (boolean) true
echo (int) $floatVar; // prints out (integer) 3
echo (string) $floatVar; // prints out (string) "3.14"
echo (boolean) $floatVar; // prints out (boolean) 1
```

Dynamic vs. Static Type

- Type checking is performed at run-time rather than compile-time
- Values have types, variables do not
- PHP, Python, Ruby
- * PHP has Type Hinting

- Type checking is performed at compile-time rather than run-time
- Allows early detection of errors and is generally more performant
- Java, ActionScript, Objective-C

Loosely vs. Strongly typed

- The type of data stored in a variable can change on different assignments
- Don't have to specify type when declared
- PHP, Perl, JavaScript

```
private int number_a = 78;
private float number_b = 4.92;
```

- Restrictions are placed on the type of data stored in a variable
- Specify the variable type when it's declared
- Python, Java, C++

```
<?php

$a = 4;
$b = 'Hello';

echo $a . $b; // 4hello
```

Basic PHP Syntax

- All PHP code must be contained between PHP *tags*.
- <?php and ?>
- Files with PHP code must end in .php for Apache to recognize it as something to send to the PHP interpreter. Ex: script.php
- All PHP statements must be terminated with a semicolon.

Basic PHP Syntax

```php
<?php

// Declare the $teacher variable
$teacher = "Dave Hauenstein";

// Print the variable out.
echo $teacher;

?>
```

```php
<?php

$nav = array(
    'Finance' => array(
        'Markets'    => 'markets.php',
        'Your Money' => 'money.php',
    ),
    'Contact' => 'contact.php',
);

function nav(array $data)
{
    $nav = "<ul>";
    foreach($data as $key => $val) {
        if(is_array($val)) {
            $nav .= "<li>$key" . nav($val) . "</li>";
        } else {
            $nav .= "<li><a href='$val'>$key</a></li>";
        }
    }
    return $nav . "</ul>";
}

echo nav($nav);

?>
```

Mixing HTML and PHP

- HTML and PHP can be combined in the same file.
- This is what makes PHP so powerful, dynamic web pages.
- Ensure PHP tags are opened and closed.
- `<p><?php echo $text; ?></p>`

Mixing HTML and PHP

```
<h3>General Information</h3>
    <h3>Slides</h3>
    <ul>
        <?php foreach($slides as $name => $url) { ?>
        <li><a href="<?php echo $url ?>"><?php echo $name ?></a></li>
        <?php } ?>
    </ul>
</div>
```

Note: If PHP's closing tag ?> is on the same line as a statement, a semicolon is not required.

Code Organization

- Mixing PHP and HTML is fine for right now.
- You may notice your scripts becoming cluttered and messy.
- There are ways that you can avoid this, and you will learn as this class goes on.

Questions?

Resources

http://davehauenstein.com/nyu/INFO1-CE9224/

Username: nyuscps
Password: $nyuscps$