INFO I-CE9224: Introduction to PHP Programming

Session 3
June 20, 2012

Resources

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

Username: nyuscps
Password: $nyuscps$

Class 3 Agenda

- PHP Language Basics Part 3
  - Array Basics
  - Truthy vs. Falsey Review
    - Logical and Comparison Operator Review
    - Flow Control Statements
- Lab Assignment 2
- Overview: Homework Assignment 1

Lab Assignment

Any Questions?

Array Basics

Array Basics

Problem:

```
$dogType = ‘Labrador’;
$dogName = ‘Max’;
$dogColor = ‘brown’;
```

Array Basics

Problem:

```
$dogType = ‘Labrador’;
$dogName = ‘Max’;
$dogColor = ‘brown’;

echo ‘Type:’ . $dogType .‘<br />’;
echo ‘Name:’ . $dogName .‘<br />’;
echo ‘Color:’ . $dogColor .‘<br />’;
```

Array Basics

Solution:

```
$dog = array(
    ‘type’ => ‘Labrador’,
    ‘name’ => ‘Max’,
    ‘color’ => ‘brown’,
);

echo gettype($dog); // prints out ‘array’
```

Array Basics

Solution:

```
$dog = array(
    ‘type’ => ‘Labrador’,
    ‘name’ => ‘Max’,
    ‘color’ => ‘brown’,
);

foreach($dog as $key => $val) {
    echo $key .‘: ’ . $val .‘<br />’;
}
```

Array Basics

Solution:

```
$dog = array(
    'type' => 'Labrador',
    'name' => 'Max',
    'color' => 'brown',
    'weight' => '65lbs',
);

foreach($dog as $key => $val) {
    echo $key .': ' . $val .'<br />';
}
```

Add new value.

Implementation doesn't change.

More on loops
next class...

Back to Array basics.

Array Basics

```
array (
    key  => value,
    key2 => value2,
    key3 => value3,
    ...
)
```

Using the array construct, takes a comma separated key => value pair of arguments

Array Basics

- Index Arrays
- Associative Arrays

Indexed Arrays

- All elements in the array are accessed by some number, the index, or sometimes key.
- Indexed arrays begin at 0 (where 0 is the first index of the array).
- As elements are added to the array, the index increments by one.
- Elements can be added in any order.
- Elements can be of any type, an array can contain mixed types.

Indexed Arrays

Syntax for creating indexed arrays:

$cities = array('New York', 'Chicago', 'Philadelphia');

```
$cities = array(
    'New York',
    'Chicago',
    'Philadelphia'
);
```

Indexed Arrays

Accessing:

```
$cities = array('New York', 'Chicago', 'Philadelphia');
```

```
echo $cities[0]; // will print 'New York'
echo $cities[2]; // will print 'Philadelphia'
echo $cities[3]; // will give a PHP warning
```

Index

Indexed Arrays

Adding additional values to the array:

```
$cities[3] = 'Los Angeles';
echo $cities[3]; // prints out 'Los Angeles'

$cities[] = 'Detroit';
echo $cities[4]; // prints out 'Detroit'

$cities[7] = 'New Orleans';
echo $cities[7]; // prints out 'New Orleans'
```

Indexed Arrays

```php
<?php

$cities = array(
    'New York',
    'Chicago',
    'Philadelphia',
);

print_r($cities);

$cities[3] = 'Los Angeles';
$cities[]  = 'New Orleans';
$cities[7] = 'Detroit';
$cities[]  = 'Atlanta';

print_r($cities);
```

```
Array
(
    [0] => New York
    [1] => Chicago
    [2] => Philadelphia
)
Array
(
    [0] => New York
    [1] => Chicago
    [2] => Philadelphia
    [3] => Los Angeles
    [4] => New Orleans
    [7] => Detroit
    [8] => Atlanta
)
```

Associative Arrays

- All elements in the array are accessed by some string index, also called a key.
- Elements can be added in any order.
- Elements can be of any type, an array can contain mixed types.
- Also called a map, hash map, hash table, dictionary, etc...
- PHP sees this the same as an indexed array under the hood: it's just an array.

Associative Arrays

Syntax for creating associative arrays:

```
$city = array(
    ‘name’ => ‘New York’,
    ‘country’ => ‘USA’,
    ‘population’ => 8391881,
);
```

Associative Arrays

Syntax for creating associative arrays:

```
$city = array();
$city['name'] = 'New York';
$city['country'] = 'USA';
$city['population'] = 8391881;
```

Associative Arrays

Accessing:

```
$city = array(
    ‘name’ => ‘New York’,
    ‘country’ => ‘USA’,
    ‘population’ => 8391881,
);
echo $city[‘name’]; // will print ‘New York’
echo $city[‘pop’]; // will give a PHP warning
```

Key

Associative Arrays

Adding additional values to the array:

```
$city['area'] = '468.48 sq mi';
echo $city['area']; // prints out '468.48 sq mi'

$city[] = 'The Big Apple';
echo $city[0]; // prints out 'The Big Apple'

$city[7] = 'Hello, NYC';
echo $city[7]; // prints out 'Hello, NYC'
```

Associative Arrays

```php
<?php

$city = array(
    'name'       => 'New York',
    'country'    => 'USA',
    'population' => 8391881,
);

print_r($city);

$city['area'] = '468.48 sq mi';
$city[]       = 'The Big Apple';
$city[7]      = 'Hello, NYC';

print_r($city);
```

```
Array
(
    [name] => New York
    [country] => USA
    [population] => 8391881
)
Array
(
    [name] => New York
    [country] => USA
    [population] => 8391881
    [area] => 468.48 sq mi
    [0] => The Big Apple
    [7] => Hello, NYC
)
```

Arrays

A literal value is not required.
An expression can be used.

```
$index = 2;
echo $cities[$index + 1]; // prints out ‘Los Angeles’

echo $city[‘na’ . ‘me’]; // prints out ‘New York’
```

Updating Array Elements

```
$cities = array('New York', 'Chicago', 'Philadelphia');
echo $cities[2]; // prints out 'Chicago'

$cities[2] = 'London';
echo $cities[2]; // prints out 'London'
```

Updating Array Elements

```
$city = array(
    ‘name’ => ‘New York’,
    ‘country’ => ‘USA’,
    ‘population’ => 8391881,
);
echo $city[‘population’]; // will print 8391881
$city[‘population’] = ‘over 8 million’;
echo $city[‘population’]; // will print ‘over 8 million’
```

Array Super Globals

Problem: I need to process an HTML form.

- I need the values of the form fields.
- The user also uploaded an image, I need to know all of the information about that image.
- But wait, I want to make sure the user is logged in before I process the form.
- I may also need the URL query parameters and a cookie I set last time they were here.
- I want to log the user's IP address with this request.
- Finally, I need to know whether it's an HTTP GET or an HTTP POST request.

Array Super Globals

Built in variables always available in all scopes.
Are all associative arrays.

Covering Today

- $_GET
- $_POST
- $_REQUEST
- $_SERVER

Covering Eventually

- $_FILES
- $_COOKIE
- $_SESSION
- $_ENV

$_GET

http://myblog.com/posts?page=3&per_page=5

```
echo $_GET['page']; // prints out 3
echo $_GET['per_page']; // prints out 5
```

By default are all passed through urldecode()

$_GET

By default are all passed through urldecode()

http://myblog.com/posts?place=new+york+city

```
echo $_GET['place']; // prints out 'new york city'
```

$_POST

```php
<?php
    if($_POST) {
        echo $_POST['product_name'];
    }
?>

<form action="handle.php" method="post">
    <input type="text" name="product_name" />
    <input type="submit" value="Submit" />
</form>
```

$_POST

Form ‘enctype’ and HTTP Content Types

application/x-www-form-urlencoded (default)
multipart/form-data (uploading images)

$_REQUEST

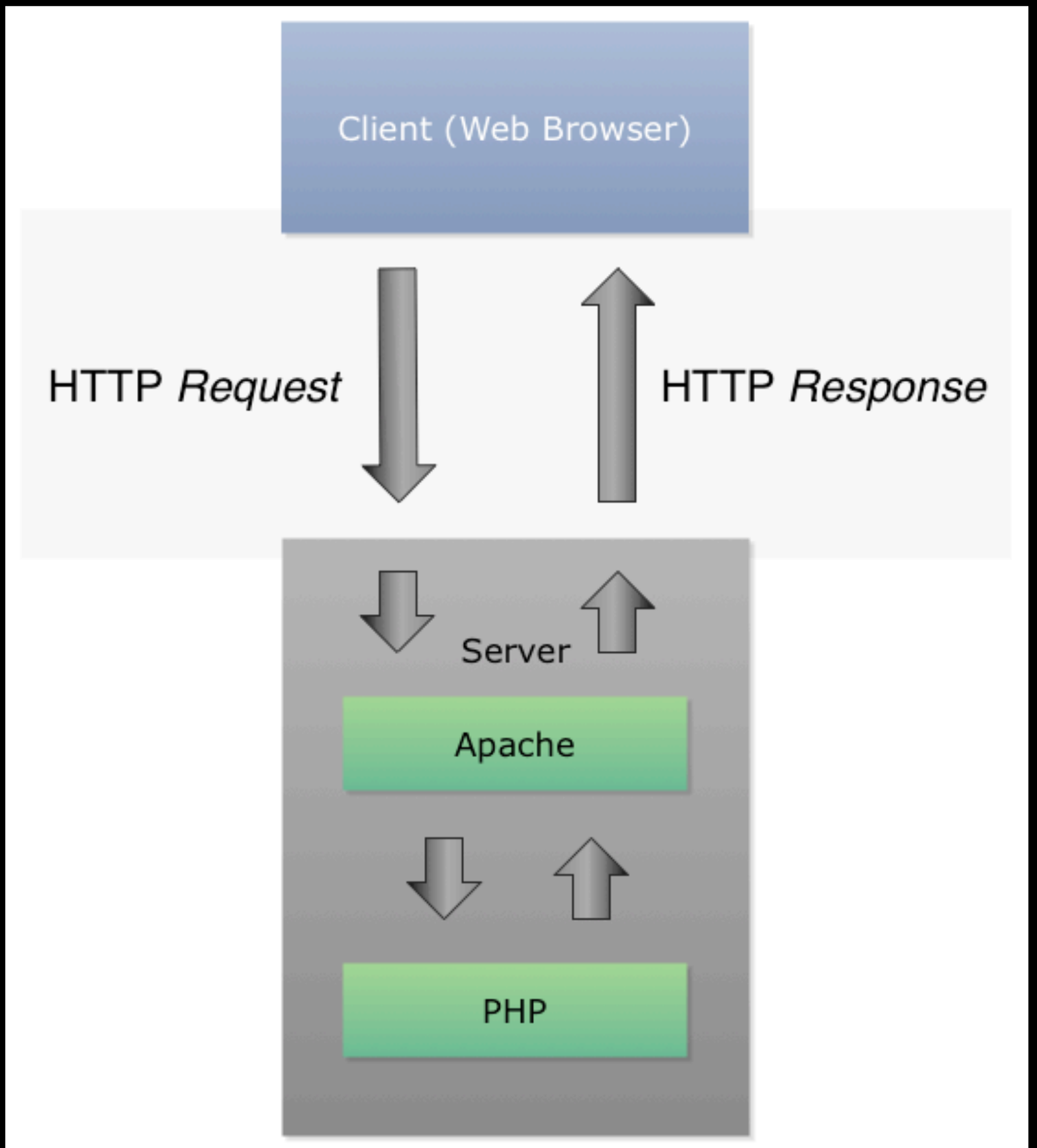
An associative array that by default* contains the contents of $_GET, $_POST, and $_COOKIE.

PHP deals with collisions and assignment ordering via configuration. By default it's EGPCS**.

Array Super Globals

Problem: I need to process an HTML form.

- I need the values of the form fields.
- The user also uploaded an image, I need to know all of the information about that image.
- But wait, I want to make sure the user is logged in before I process the form.
- I may also need the URL query parameters and a cookie I set last time they were here.
- I want to log the user's IP address with this request.
- Finally, I need to know whether it's an HTTP GET or an HTTP POST request.


Client (Web Browser)
HTTP *Request*
HTTP *Response*
Server
Apache
PHP

HTTP Request

```
GET /?page=5 HTTP/1.1
User-Agent: curl/7.21.4
Host: davehauenstein.com
Accept: */*
```

HTTP Response

```
HTTP/1.1 200 OK
Date: Thu, 29 Mar 2012 17:34:53 GMT
Server: Apache/2.2.16 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 5407
Content-Type: text/html

<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0...
<html xmlns="http://www.w3.org/1999/xhtml" xml:...
    <head>
        <title>Dave Hauenstein | Programmer</ti...
        <link href="favicon.png" rel="shortcut ico...
        etc ...
```

$_SERVER

- An array containing information such as headers, paths, and script locations.
- The entries in this array are created by the web server.
- No guarantee that every web server will provide any of these.

$_SERVER

- PHP_SELF
- REMOTE_ADDR
- HTTPS
- REQUEST_METHOD
- QUERY_STRING
- DOCUMENT_ROOT
- SERVER_PORT
- SERVER_ADDR
- SERVER_SOFTWARE
- HTTP_* (ex: HTTP_ACCEPT)

$_SERVER

```php
<?php
    echo $_SERVER['PHP_SELF']          . '<br />';
    echo $_SERVER['REMOTE_ADDR']       . '<br />';
    echo $_SERVER['REQUEST_METHOD']    . '<br />';
    echo $_SERVER['QUERY_STRING']      . '<br />';
    echo $_SERVER['DOCUMENT_ROOT']     . '<br />';
    echo $_SERVER['SERVER_PORT']       . '<br />';
    echo $_SERVER['SERVER_ADDR']       . '<br />';
    echo $_SERVER['SERVER_SOFTWARE']   . '<br />';
    echo $_SERVER['HTTP_ACCEPT']       . '<br />';

    if (isset($_SERVER['HTTPS'])) {
        echo $_SERVER['HTTPS'];
    }
?>
```

$_SERVER

```
/nyu/arrays.php
172.16.176.1
GET
place=new+york+city
/var/app/com.dave.dev/www
80
172.16.176.131
Apache/2.2.20 (Ubuntu)
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

Review

Truthy vs. Falsey
Logical and Comparison Operators

Comparison Operators

Operator	Example	Result
== (equal)	$x == $y	true if $x equals $y; false otherwise
!= or <> (not equal)	$x != $y	true if $x does not equal $y; false otherwise
=== (identical)	$x === $y	true if $x equals $y and they are of the same type; false otherwise
!== (not identical)	$x !== $y	true if $x does not equal $y or they are of the same type; false otherwise
< (less than)	$x < $y	true if $x is less than $y; false otherwise
> (greater than)	$x > $y	true if $x is greater than $y; false otherwise
<= (less than or equal to)	$x <= $y	true if $x is less than or equal to $y; false otherwise
>= (greater than or equal to)	$x >= $y	true if $x is greater than or equal to $y; false otherwise

Logical Operators

Operator	Example	Result
&& (and)	$x && $y	true if both $x and $y evaluate to true; false otherwise
and	$x and $y	true if both $x and $y evaluate to true; false otherwise
|| (or)	$x || $y	true if either $x or $y evaluates to true; false otherwise
or	$x or $y	true if either $x or $y evaluates to true; false otherwise
xor	$x xor $y	true if $x or $y (but not both) evaluates to true; false otherwise
! (not)	!$x	true if $x is false; false if $x is true

True vs. False

True

- 1
- 1 == 1
- 5 > 2
- “hello” != “goodbye”

False

- 5 < 2
- gettype(3) == “array”
- “hello” == “goodbye”
- 23 === “23”

Additional False Values

- the literal value *false*
- The integer zero (0)
- The float zero (0.0)
- An empty string (“ “)
- The string zero (“0”)
- An array with zero elements
- The *null* type
- A SimpleXML object created from an empty XML tag

Flow Control Statements

Conditional Statements

Problem: I would like to run a block of code if some condition evaluates to true.

if Statement

```
if ( expression ) {
    // run some code
}

// run more code
```

if Statement

```php
<?php

$person = 'Franklin';

if ( 'Franklin' === $person ) {
    echo 'Hi Frank! I was expecting you.';
    echo '<br />';
}

echo "{$person} has arrived.";
```

```php
<?php

$person = array(
    'name'       => 'Franklin',
    'age'        => 32,
    'hair'       => 'brown',
    'passcode' => 'open says me',
);

if ( 'Franklin' === $person['name'] &&
     20 === ($person['age'] / 2 + 4) &&
     'brown' === $person['hair'] &&
     'open says me' === $person['passcode']
) {
    echo 'Hi Frank! I was expecting you.';
    echo '<br />';
}

echo "{$person['name']} has arrived.";
```

Conditional Statements

Problem: I would like to run a block of code if some condition evaluates to true, however, if said condition is false, I would like to run a different block of code.

if / else Statement

```php
<?php

$age = 19;

if ($age >= 21) {
    echo 'You can enter the bar';
} else {
    echo 'You cannot enter the bar';
}
```

Conditional Statements

Problem: I would like to run different blocks of code based on several conditions.

If my first expression is false, I would like to try something else. Finally, if neither of those things are true, run some default code.

if / else if / else Statement

```php
<?php

$age = 19;

if ($age >= 21) {
    echo 'You can enter the bar';
} else if ($age >= 18 && $age < 21) {
    echo 'Enter, but no drinking';
} else {
    echo 'You cannot enter the bar';
}
```

Abuse of *if / else if / else*

```php
$title = 'friend';

if ($title == 'friend') {
    $greeting = "Hey, Buddy, what's up?";
} else if ($title == 'dad' || $title == 'mom') {
    $greeting = 'Hi Dad, good to see you.';
} else if ($title == 'boss') {
    $greeting = 'Good evening, fine sir.';
} else {
    $greeting = 'Hello, how are you?';
}

echo $greeting;
```

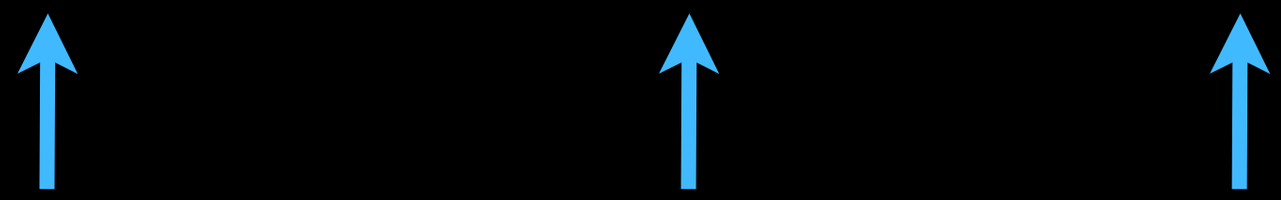
switch Statement

```php
$title = 'friend';

switch ($title) {
    case 'friend':
        $greeting = "Hey, Buddy, what's up?";
        break;

    case 'father':
    case 'mother':
        $greeting = 'Hi parental unit, good to see you.';
        break;

    case 'boss':
        $greeting = 'Good evening, fine sir.';
        break;

    default:
        $greeting = 'Hello, how are you?';
}

echo $greeting;
```

Ternary Operator

A compact version of the if/else construct

(*expression1*) ? *expression2* : *expression3*



true/false any value any value

Ternary Operator

(*expression1*) ? *expression2* : *expression3*

```php
<?php

$products = 46;

$status = ($products >= 50) ? 'surplus' : 'shortage';
echo ($status == 'surplus') ? 'Great' : 'Order More!';
```

Shorthand Ternary Operator

(expression l) ?: expression3

```php
<?php

$author = null;
echo ($author) ?: 'anonymous'; // anonymous

$author = 'Dawkins';
echo ($author) ?: 'anonymous'; // Dawkins
```

Review Next Week's Homework Assignment

Class 3 Lab

http://davehauenstein.com/nyu/INFO1-CE9224-2012-Summer/labs/class3.pdf

Resources

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

Username: nyuscps
Password: $nyuscps$