























CSC 443: Web Programming

Haidar Harmanani

Department of Computer Science and Mathematics
Lebanese American University
Byblos, 1401 2010 Lebanon

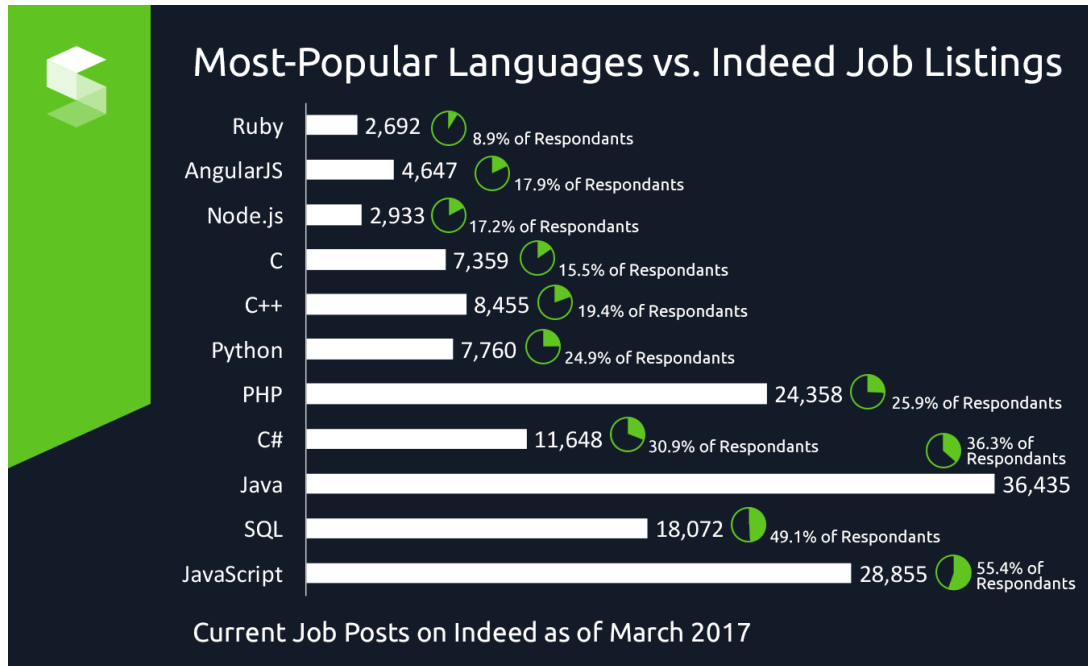
CSC443: Web Programming

The 2017 Top Programming Languages

Language Rank	Types	Spectrum Ranking
1. Python	 	100.0
2. C	  	99.7
3. Java	  	99.5
4. C++	  	97.1
5. C#	  	87.7
6. R		87.7
7. JavaScript	 	85.6
8. PHP		81.2
9. Go	 	75.1
10. Swift	 	73.7

Most-Popular Languages vs. Indeed Job Listings

3



CSC443: Web Programming

URLs and web servers

4

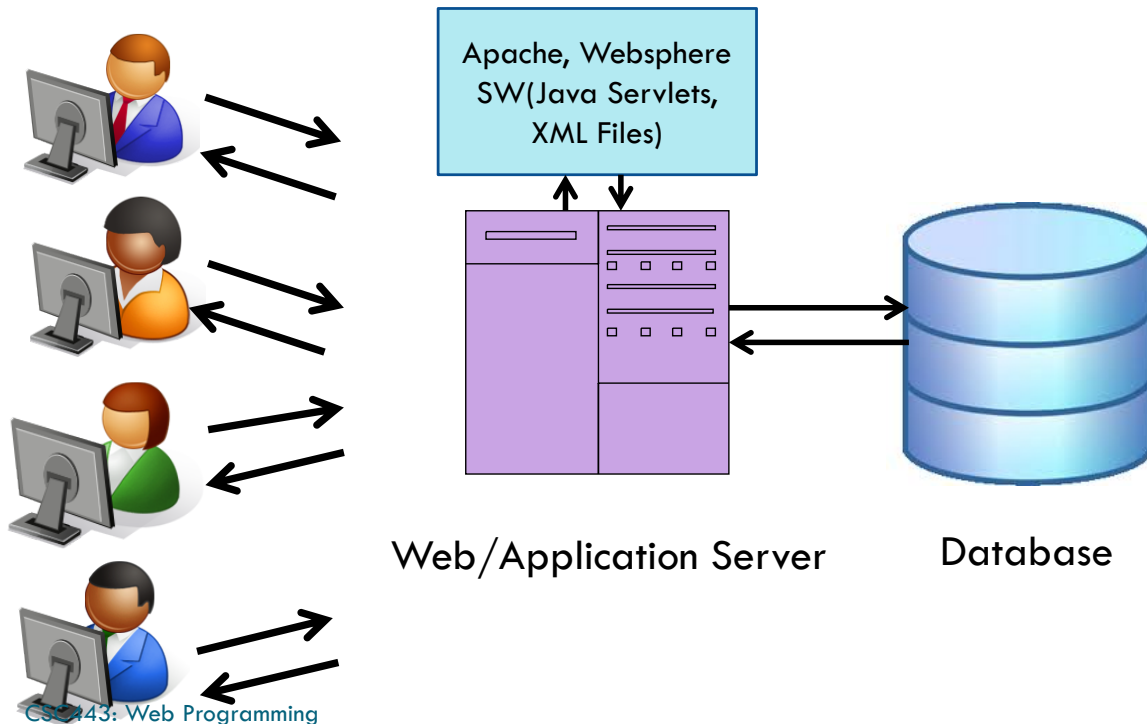
`http://server/path/file`

- Usually when you type a URL in your browser:
 - ▣ Your computer looks up the server's IP address using DNS
 - ▣ Your browser connects to that IP address and requests the given file
 - ▣ The web server software (e.g. Apache) grabs that file from the server's local file system
 - ▣ The server sends back its contents to you

CSC443: Web Programming

URLs and web servers (cont.)

5



URLs and web servers (cont.)

6

`http://www.facebook.com/home.php`

- Some URLs actually specify programs that the web server should *run*, and then send their output back to you as the result:
 - ▣ The above URL tells the server **facebook.com** to run the program **home.php** and send back its output

Server-Side web programming

7

- Server-side pages are programs written using one of many web programming languages/frameworks
 - ▣ examples: PHP, Java/JSP, Ruby on Rails, ASP.NET, Python, Perl



CSC443: Web Programming

Server-Side web programming (cont.)

8

- Also called *server side scripting*:
 - ▣ Dynamically edit, change or add any content to a Web page
 - ▣ Respond to user queries or data submitted from HTML forms
 - ▣ Access any data or databases and return the results to a browser
 - ▣ Customize a Web page to make it more useful for individual users
 - ▣ Provide security since your server code cannot be viewed from a browser

CSC443: Web Programming

Server-Side web programming (cont.)

9

- Web server:
 - ▣ contains software that allows it to run server side programs
 - ▣ sends back their output as responses to web requests
- Each language/framework has its pros and cons
 - ▣ we use PHP

CSC443: Web Programming

What is PHP?

10

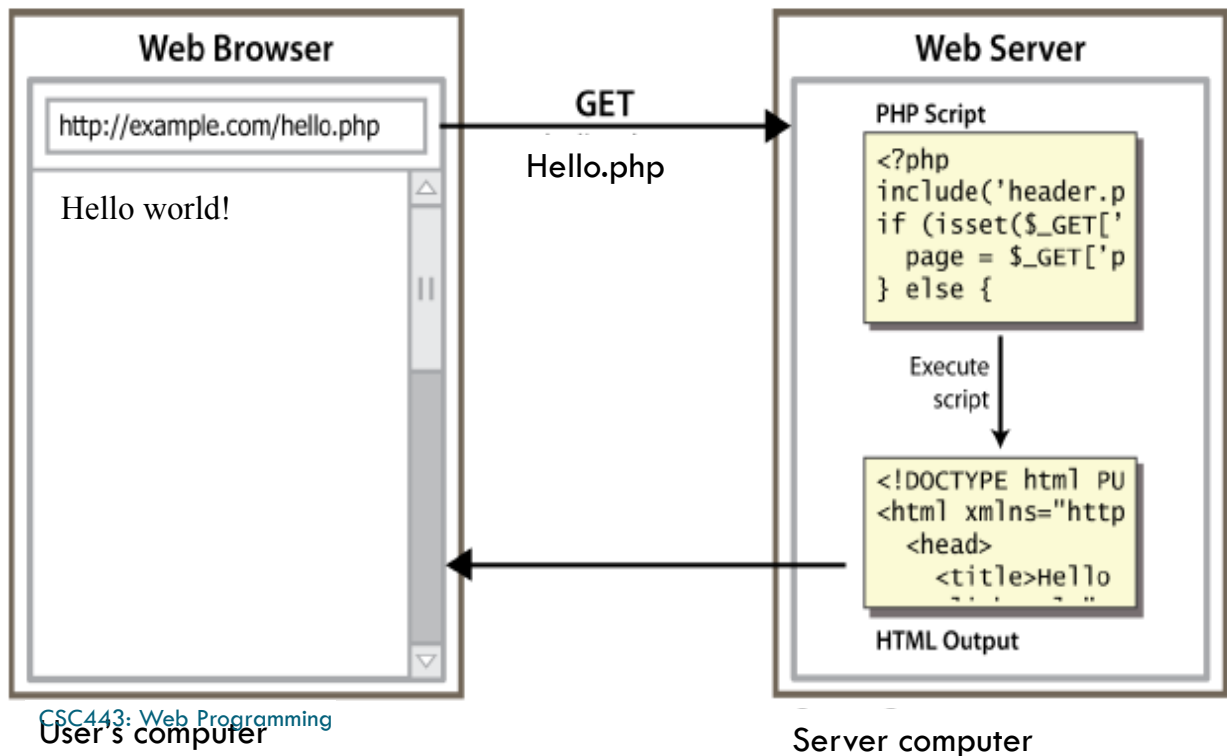
- PHP stands for "PHP Hypertext Preprocessor"
- Server-side scripting language
- Used to make web pages dynamic:
 - ▣ provide different content depending on context
 - ▣ interface with other services: database, e-mail, etc.
 - ▣ authenticate users
 - ▣ process form information
- PHP code can be embedded in XHTML code



CSC443: Web Programming

Lifecycle of a PHP web request

11



Why PHP?

12

- Free and open source
- Compatible
 - ▣ as of November 2006, there were more than 19 million websites (domain names) using PHP.
- Simple

Hello World!

13

```
<?php  
print "Hello, world!";  
?>
```

PHP

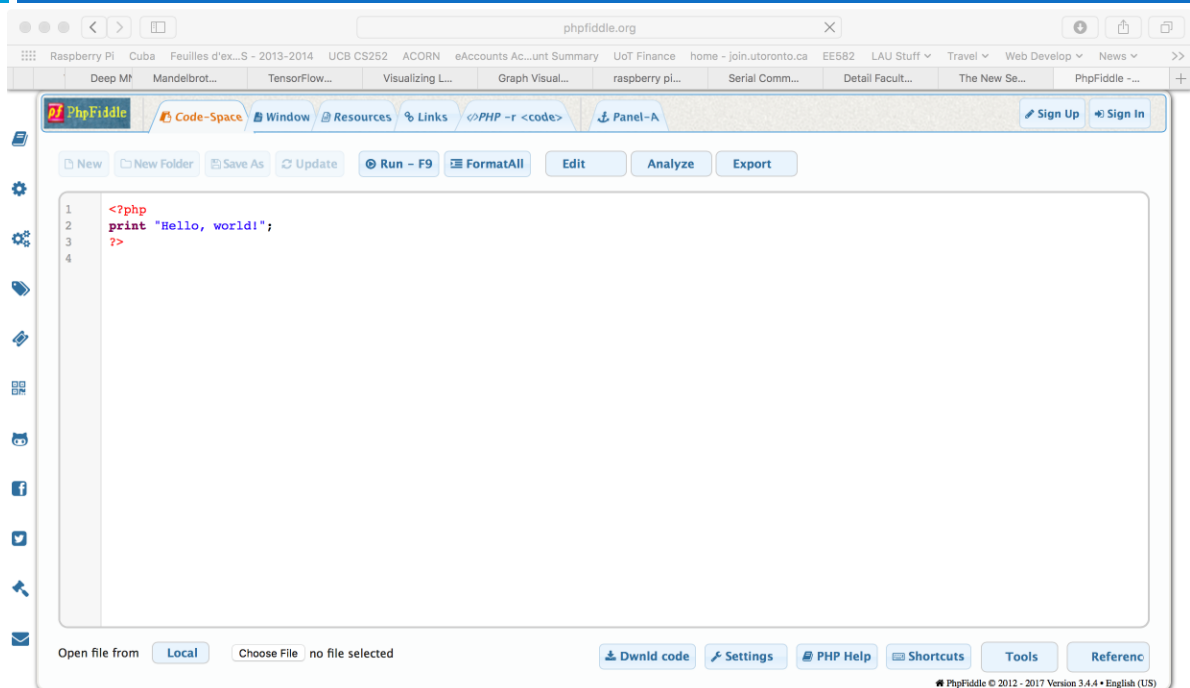
Hello world!

output

CSC443: Web Programming

Viewing PHP output: phpfiddle.org

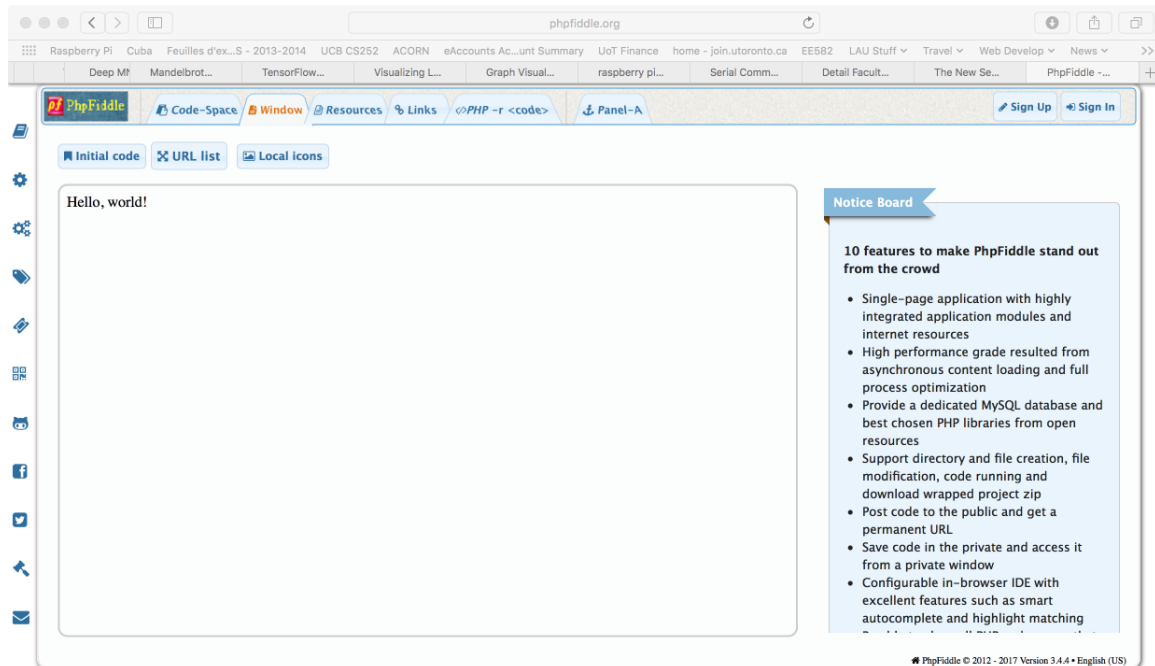
14



CSC443: Web Programming

Viewing PHP output: `phpfiddle.org`

15



CSC443: Web Programming

16

PHP Basic Syntax

CSC443: Web Programming

PHP syntax template

17

HTML content

<?php

PHP code

?>

HTML content

<?php

PHP code

?>

HTML content ...

PHP

- ❑ Contents of a .php file between **<?php** and **?>** are executed as PHP code
- ❑ All other contents are output as pure HTML
- ❑ We can switch back and forth between HTML and PHP "modes"

CSC443: Web Programming

Console output: print

18

```
print "text";
```

PHP

```
print "Hello, World!\n";  
print "Escape \"chars\" are the SAME as in Java!\n";  
print "You can have  
line breaks in a string."  
print 'A string can use "single-quotes". It\'s cool!';
```

PHP

Hello world! Escape "chars" are the SAME as in Java! You can have line breaks in a string. A string can use "single-quotes". It's cool!

output

CSC443: Web Programming

Variables

19

```
$name = expression;
```

PHP

```
$user_name = "mundruid78";  
$age = 16;  
$drinking_age = $age + 5;  
$this_class_rocks = TRUE;
```

PHP

- ❑ names are case sensitive
- ❑ names always begin with \$, on both declaration and usage
- ❑ always implicitly declared by assignment (type is not written)
- ❑ a loosely typed language (like JavaScript or Python)

CSC443: Web Programming

Variables

20

- ❑ basic types: *int, float, boolean, string, array, object, NULL*
 - ▣ test type of variable with `is_type` functions, e.g. `is_string`
 - ▣ `gettype` function returns a variable's type as a string
- ❑ PHP *converts between types automatically* in many cases:
 - ▣ `string` → `int` auto-conversion on `+`
 - ▣ `int` → `float` auto-conversion on `/`
- ❑ type-cast with **(type)**:
 - ▣ `$age = (int) "21";`

CSC443: Web Programming

Arithmetic operators

21

- + - * / % . ++ --
- = += -= *= /= %= .=
- many operators auto-convert types: 5 + "7" is 12

CSC443: Web Programming

Comments

22

```
# single-line comment
// single-line comment
/*
multi-line comment
*/
```

PHP

- like Java, but # is also allowed
 - a lot of PHP code uses # comments instead of //

CSC443: Web Programming

String Type

23

```
$favorite_food = "Ethiopian";  
print $favorite_food[2];  
$favorite_food = $favorite_food . " cuisine";  
print $favorite_food;
```

PHP

- ❑ zero-based indexing using bracket notation
- ❑ there is no char type; each letter is itself a String
- ❑ string concatenation operator is . (period), not +
 - ❑ `5 + "2 turtle doves" === 7`
 - ❑ `5 . "2 turtle doves" === "52 turtle doves"`

CSC443: Web Programming

String Functions

24

```
# index 0123456789012345  
$name = "Stefanie Hatcher";  
$length = strlen($name);  
$cmp = strcmp($name, "Brian Le");  
$index = strpos($name, "e");  
$first = substr($name, 9, 5);  
$name = strtoupper($name);
```

PHP

CSC443: Web Programming

String Functions (cont.)

25

Name	Java Equivalent
<u>strlen</u>	length
<u>strpos</u>	indexOf
<u>substr</u>	substring
<u>strtolower</u> , <u>strtoupper</u>	toLowerCase, toUpperCase
<u>trim</u>	trim
<u>explode</u> , <u>implode</u>	split, join
<u>strcmp</u>	compareTo

CSC443: Web Programming

Interpreted Strings

26

```
$age = 16;
print "You are " . $age . " years old.\n";
print "You are $age years old.\n"; # You are 16 years old.
PHP
```

- strings inside " " are interpreted
 - ▣ variables that appear inside them will have their values inserted into the string
- strings inside ' ' are not interpreted:

```
print 'You are $age years old.\n'; # You are $age years
old. \n
PHP
```

CSC443: Web Programming

Interpreted Strings (cont.)

27

```
print "Today is your $ageth birthday.\n"; # $ageth not found
print "Today is your {$age}th birthday.\n";
```

PHP

- if necessary to avoid ambiguity, can enclose variable in {}

CSC443: Web Programming

Interpreted Strings (cont.)

28

```
$name = "Xenia";
$name = NULL;
if (isset($name)) {
    print "This line isn't going to be reached.\n";
}
```

PHP

- a variable is NULL if
 - ▣ it has not been set to any value (undefined variables)
 - ▣ it has been assigned the constant NULL
 - ▣ it has been deleted using the unset function
- can test if a variable is NULL using the isset function
- NULL prints as an empty string (no output)

CSC443: Web Programming

for loop (same as Java)

29

```
for (initialization; condition; update) {  
    statements;  
}
```

PHP

```
for ($i = 0; $i < 10; $i++) {  
    print "$i squared is " . $i * $i . ".\n";  
}
```

PHP

CSC443: Web Programming

bool (Boolean) type

30

```
$feels_like_summer = FALSE;  
$php_is_great = TRUE;  
$student_count = 7;  
$nonzero = (bool) $student_count; # TRUE
```

PHP

- ❑ the following values are considered to be FALSE (all others are TRUE):
 - 0 and 0.0 (but NOT 0.00 or 0.000)
 - "", "0", and NULL (includes unset variables)
 - arrays with 0 elements
- ❑ FALSE prints as an empty string (no output); TRUE prints as a 1

CSC443: Web Programming

if/else statement

31

```
if (condition) {  
    statements;  
} elseif (condition) {  
    statements;  
} else {  
    statements;  
}
```

PHP

CSC443: Web Programming

while loop (same as Java)

32

```
while (condition) {  
    statements;  
}
```

PHP

```
do {  
    statements;  
} while (condition);
```

PHP

CSC443: Web Programming

Math operations

33

```
$a = 3;  
$b = 4;  
$c = sqrt(pow($a, 2) + pow($b, 2));
```

PHP

math functions

<u>abs</u>	<u>ceil</u>	<u>cos</u>	<u>floor</u>	<u>log</u>	<u>log10</u>	<u>max</u>
<u>min</u>	<u>pow</u>	<u>rand</u>	<u>round</u>	<u>sin</u>	<u>sqrt</u>	<u>tan</u>

math constants

M_PI	M_E	M_LN2
------	-----	-------

CSC443: Web Programming

int and float Types

34

```
$a = 7 / 2; # float: 3.5  
$b = (int) $a; # int: 3  
$c = round($a); # float: 4.0  
$d = "123"; # string: "123"  
$e = (int) $d; # int: 123
```

PHP

- ❑ int for integers and float for reals
- ❑ division between two int values produces a float

CSC443: Web Programming

PHP exercise 1 [In class, use

<http://phpfiddle.org>]

35

- For your first PHP exercise, echo the following statement to the browser:
 “Twinkle, Twinkle little star.”
- Next, create two variables, one for the word “Twinkle” and one for the word “star”.
 - ▣ Echo the statement to the browser, this time substituting the variables for the relevant words.
 - ▣ Change the value of each variable to whatever you like, and echo the statement a third time.
 - ▣ Remember to include code to show your statements on different lines.

CSC443: Web Programming

PHP exercise 2 [In class, use

<http://phpfiddle.org>]

36

- PHP includes all the standard arithmetic operators.
- For this PHP exercise, you will use them along with variables to print equations to the browser. In your script, create the following variables:
 $x=10$;
 $y=7$;
- Write code to print out the following:
 $10 + 7 = 17$
 $10 - 7 = 3$
 $10 * 7 = 70$
 $10 / 7 = 1.4285714285714$
 $10 \% 7 = 3$
- Use numbers only in the above variable assignments, not in the echo statements.

CSC443: Web Programming

PHP exercise 3 [In class, use

<http://phpfiddle.org>]

37

- Arithmetic-assignment operators perform an arithmetic operation on the variable at the same time as assigning a new value.
- For this PHP exercise, write a script to reproduce the output below. Manipulate only one variable using no simple arithmetic operators to produce the values given in the statements.
- Hint: In the script each statement ends with "Value is now \$variable."
Value is now 8.
Add 2. Value is now 10.
Subtract 4. Value is now 6.
Multiply by 5. Value is now 30.
Divide by 3. Value is now 10.
Increment value by one. Value is now 11.
Decrement value by one. Value is now 10.

CSC443: Web Programming

PHP exercise 4 [In class, use

<http://phpfiddle.org>]

38

- When you are writing scripts, you will often need to see exactly what is inside your variables. For this PHP exercise, think of the ways you can do that, then write a script that outputs the following, using the echo statement only for line breaks.

```
string(5) "Harry"  
Harry  
int(28)  
NULL
```

CSC443: Web Programming

PHP exercise 5 [In class, use

<http://phpfiddle.org>]

39

- For this PHP exercise, write a script using the following variable:
`$around="around";`
- Single quotes and double quotes don't work the same way in PHP. Using single quotes (' ') and the concatenation operator, echo the following to the browser, using the variable you created:
What goes around, comes around.

CSC443: Web Programming

PHP exercise 5 [In class, use

<http://phpfiddle.org>]

40

- In this PHP exercise, you will use a conditional statement to determine what gets printed to the browser. Write a script that gets the current month and prints one of the following responses, depending on whether it's August or not:

It's August, so it's really hot.

Not August, so at least not in the peak of the heat.

- Hint: the function to get the current month is `'date('F', time())'` for the month's full name.

CSC443: Web Programming

PHP exercise 6 [In class, use

<http://phpfiddle.org>]

41

- Loops are very useful in creating lists and tables. In this PHP exercise, you will use a loop to create a list of equations for squares.
- Using a for loop, write a script that will send to the browser a list of squares for the numbers 1-12. Use the format, "1 * 1 = 1", and be sure to include code to print each formula on a different line.

CSC443: Web Programming

PHP exercise 7 [In class, use

<http://phpfiddle.org>]

42

- HTML tables involve a lot of repetitive coding - a perfect place to use for loops. You can do even more if you nest the for loops.
- In this PHP exercise, use two for loops, one nested inside another. Create the following multiplication table:

1	2	3	4	5	6	7
2	4	6	8	10	12	14
3	6	9	12	15	18	21
4	8	12	16	20	24	28
5	10	15	20	25	30	35
6	12	18	24	30	36	42
7	14	21	28	35	42	49

CSC443: Web Programming

Arrays

```
$name = array();           # create
$name = array(value0, value1, ..., valueN);
$name[index]               # get element value
$name[index] = value;      # set element value
$name[] = value;           # append
```

PHP

```
$a = array();              # empty array (length 0)
$a[0] = 23;                # stores 23 at index 0 (length 1)
$a2 = array("some", "strings", "in", "an", "array");
$a2[] = "Ooh!";            # add string to end (at index 5)
```

PHP

- Append: use bracket notation without specifying an index
- Element type is not specified; can mix types

Array functions

45

function name(s)	description
<u>count</u>	number of elements in the array
<u>print_r</u>	print array's contents
<u>array_pop</u> , <u>array_push</u> , <u>array_shift</u> , <u>array_unshift</u>	using array as a stack/queue
<u>in_array</u> , <u>array_search</u> , <u>array_reverse</u> , <u>sort</u> , <u>rsort</u> , <u>shuffle</u>	searching and reordering
<u>array_fill</u> , <u>array_merge</u> , <u>array_intersect</u> , <u>array_diff</u> , <u>array_slice</u> , <u>range</u>	creating, filling, filtering
<u>array_sum</u> , <u>array_product</u> , <u>array_unique</u> , <u>array_filter</u> , <u>array_reduce</u>	processing elements

CSC443: Web Programming

Array function example

46

```
$tas = array("MD", "BH", "KK", "HM", "JP");  
for ($i = 0; $i < count($tas); $i++) {  
    $tas[$i] = strtolower($tas[$i]);  
}  
$morgan = array_shift($tas);  
array_pop($tas);  
array_push($tas, "ms");  
array_reverse($tas);  
sort($tas);  
$best = array_slice($tas, 1, 2);
```

PHP

- the array in PHP replaces many other collections in Java
 - ▣ list, stack, queue, set, map, ...

CSC443: Web Programming

foreach loop

47

```
foreach ($array as $variableName) {  
    ...  
}
```

PHP

```
$fellowship = array("Frodo", "Sam", "Gandalf",  
"Strider", "Gimli", "Legolas", "Boromir");  
print "The fellowship of the ring members are: \n";  
for ($i = 0; $i < count($fellowship); $i++) {  
    print "{$fellowship[$i]}\n";  
}  
print "The fellowship of the ring members are: \n";  
  
foreach ($fellowship as $fellow) {  
    print "$fellow\n";  
}
```

PHP

CSC443: Web Programming

Multidimensional Arrays

48

```
<?php $AmazonProducts = array( array("BOOK", "Books", 50),  
                                array("DVDs", "Movies", 15),  
                                array("CDs", "Music", 20)  
                                );  
for ($row = 0; $row < 3; $row++) {  
    for ($column = 0; $column < 3; $column++) { ?>  
        <p> | <?= $AmazonProducts[$row][$column] ?>  
        <?php } ?>  
    </p>  
<?php } ?>
```

PHP

CSC443: Web Programming

Multidimensional Arrays (cont.)

49

```
<?php $AmazonProducts = array( array("Code" =>"BOOK",
"Description" => "Books", "Price" => 50),
                                array("Code" => "DVDs",
"Description" => "Movies", "Price" => 15),
                                array("Code" => "CDs",
"Description" => "Music", "Price" => 20)
                                );
for ($row = 0; $row < 3; $row++) { ?>
    <p> | <?= $AmazonProducts[$row]["Code"] ?> | <?=
$AmazonProducts[$row]["Description"] ?> | <?=
$AmazonProducts[$row]["Price"] ?>
    </p>
<?php } ?>
```

PHP

CSC443: Web Programming

String compare functions

50

Name	Function
<u>strcmp</u>	compareTo
<u>strstr</u> , <u>strchr</u>	find string/char within a string
<u>strpos</u>	find numerical position of string
<u>str_replace</u> , <u>substr_replace</u>	replace string

- ❑ Comparison can be:
 - ▣ Partial matches
 - ▣ Others
- ❑ Variations with non case sensitive functions
 - ▣ [strcasecmp](#)

CSC443: Web Programming

String compare functions examples

51

```
$offensive = array( offensive word1, offensive word2);  
$feedback = str_replace($offcolor, "%!@*",  
$feedback);
```

PHP

```
$test = "Hello World! \n";  
print strpos($test, "o");  
print strpos($test, "o", 5);
```

PHP

```
$toaddress = "feedback@example.com";  
if(strstr($feedback, "shop")  
    $toaddress = "shop@example.com";  
else if(strstr($feedback, "delivery")  
    $toaddress = "fulfillment@example.com";
```

PHP

Regular expressions

52

[a-z]at	#cat, rat, bat...
[aeiou]	
[a-zA-Z]	
[^a-z]	#not a-z
[:alnum:]]+	#at least one alphanumeric char
(very) *large	#large, very very very large...
(very){1, 3}	#counting "very" up to 3
^bob	#bob at the beginning
com\$	#com at the end

PHPRegExp

- ❑ Regular expression: a pattern in a piece of text
- ❑ PHP has:
 - ▣ POSIX
 - ▣ Perl regular expressions

Printing HTML tags in PHP = bad style

```
<?php
print "<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML
1.1//EN\" \"\n";
print "
\"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd\">\n";
print "<html xmlns=\"http://www.w3.org/1999/xhtml\">\n";
print " <head>\n";
print " <title>Geneva's web page</title>\n";
...
for ($i = 1; $i <= 10; $i++) {
print "<p> I can count to $i! </p>\n";
}
?>
```

HTML

- ❑ best PHP style is to minimize print/echo statements in embedded PHP code
- ❑ but without print, how do we insert dynamic content into the page?

PHP expression blocks

55

```
<?= expression ?>
```

PHP

```
<h2> The answer is <?= 6 * 7 ?> </h2>
```

PHP

The answer is 42

output

- PHP expression block: a small piece of PHP that evaluates and embeds an expression's value into HTML

- ▣ `<?= expression ?>` is equivalent to:

```
<?php print expression; ?>
```

PHP

CSC443: Web Programming

Expression block example

56

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>CSE 190 M: Embedded PHP</title></head>
<body>
<?php
for ($i = 99; $i >= 1; $i--) {
?>
<p> <?= $i ?> bottles of beer on the wall, <br />
<?= $i ?> bottles of beer. <br />
Take one down, pass it around, <br />
<?= $i - 1 ?> bottles of beer on the wall. </p>
<?php
}
?>
</body>
</html>
```

PHP

CSC443: Web Programming

Common errors: unclosed braces, missing = sign

57

```
...
<body>
<p>Watch how high I can count:
<?php
for ($i = 1; $i <= 10; $i++) {
?>
    <? $i ?>
</p>
</body>
</html>
```

PHP

- ❑ if you forget to close your braces, you'll see an error about 'unexpected \$end'
- ❑ if you forget = in <?=?, the expression does not produce any output

CSC443: Web Programming

Complex expression blocks

58

```
...
<body>
<?php
for ($i = 1; $i <= 3; $i++) {
    ?>
    <h<?= $i ?>>This is a level <?= $i ?> heading.</h<?= $i
?>>
    <?php
}
?>
</body>
```

PHP

This is a level 1 heading.

This is a level 2 heading.

This is a level 3 heading.

output

CSC443: Web Programming

Functions

Functions

```
function name(parameterName, ..., parameterName) {  
    statements;  
}
```

PHP

```
function quadratic($a, $b, $c) {  
    return -$b + sqrt($b * $b - 4 * $a * $c) / (2  
    * $a);  
}
```

PHP

- ❑ parameter types and return types are not written
- ❑ a function with no return statements implicitly returns NULL

Default Parameter Values

61

```
function print_separated($str, $separator = ", ") {  
    if (strlen($str) > 0) {  
        print $str[0];  
        for ($i = 1; $i < strlen($str); $i++) {  
            print $separator . $str[$i];  
        }  
    }  
}
```

PHP

```
print_separated("hello"); # h, e, l, l, o  
print_separated("hello", "-"); # h-e-l-l-o
```

PHP

- if no value is passed, the default will be used

CSC443: Web Programming

PHP Arrays Ex. 1

62

- *Arrays allow you to assign multiple values to one variable. For this PHP exercise, write an array variable of weather conditions with the following values: rain, sunshine, clouds, hail, sleet, snow, wind. Using the array variable for all the weather conditions, echo the following statement to the browser:*

We've seen all kinds of weather this month. At the beginning of the month, we had snow and wind. Then came sunshine with a few clouds and some rain. At least we didn't get any hail or sleet.

- *Don't forget to include a title for your page, both in the header and on the page itself.*

CSC443: Web Programming

PHP Arrays Ex. 2

63

- *For this exercise, you will use a list of ten of the largest cities in the world. (Please note, these are not the ten largest, just a selection of ten from the largest cities.) Create an array with the following values: Tokyo, Mexico City, New York City, Mumbai, Seoul, Shanghai, Lagos, Buenos Aires, Cairo, London.*
- *Print these values to the browser separated by commas, using a loop to iterate over the array. Sort the array, then print the values to the browser in an unordered list, again using a loop.*
- *Add the following cities to the array: Los Angeles, Calcutta, Osaka, Beijing. Sort the array again, and print it once more to the browser in an unordered list.*