1

Basic PHP Syntax

Arrays

Strings and regular expressions

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

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

Array function example


```
$tas = array("MD", "BH", "KK", "HM", "JP");
for ($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, ...

foreach loop

5

```
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

Multidimensional Arrays

6

```
<?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

Multidimensional Arrays (cont.)

7

```
<?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

String compare functions

8

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

- Comparison can be:
  - Partial matches
  - Others
- Variations with non case sensitive functions
  - strcasecmp

`String` compare functions examples


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

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

```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


```
[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**

11 Embedded PHP

Printing HTML tags in PHP = bad style


```php
<?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


```
<?= 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

Expression block example


```
<!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

Common errors: unclosed braces, missing = sign


```
...
<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

Complex expression blocks


```
...
<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

17

Advanced PHP Syntax

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


```
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

PHP Arrays Ex. 1

- *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.*

PHP Arrays Ex. 2

- *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.*