Arrays

Arrays in PHP are a series of key-value pairs, i.e. for every item in an array there is a key/index used to access an associated stored value.

Two types of arrays:
- uses numbers as keys.
- use strings as keys.

Arrays in PHP...
- Not declared a fixed size.
- Do not require sequential indexes.
- May hold diverse data types in a single array.
Creating numeric arrays: Method 1

List array contents with reserved word 'array'.

```php
$myarray = array("Bob", "Joe", "Sue", "James");
```

Specify a starting numeric index, if desired.

```php
$months = array(1=>"Jan", "Feb", "Mar", "Apr");
```

Specify non-consecutive numeric indices, if desired.

```php
$rooms = array(2910=>436, 3310=>492);
```

Creating numeric arrays: Method 2

Assign values to cells using traditional array syntax (no array declaration needed).

```php
$friends[0] = "Bob";
$friends[1] = "Joe";
$friends[5] = "Raymond";
```

With either method, can add to end of an existing array (using the next index) using the following syntax:

```php
$friends[] = "Amy";
$friends[] = "Kelly";
```
Removing an element from an array

`unset($array[$index])` will delete a index and its associated value from an array.

```php
$example = array("eggs", "ham", "cheese", "milk");
unset($array[1]);
```

Outputting array contents

Because array syntax is different from other variables, have to print differently.

```php
echo "Your grocery list contains $list[$item];
```

The above will not work. To print array contents, do one of the following:

- Move array variable outside of all quotes.
  ```php
echo "Your grocery list contains ".$list[$item];
```
- Wrap the array reference variable in curly braces
  ```php
  echo "Your grocery list contains {$list[$item]}";
  ```
### Array count

count() returns the size of the array.

```php
<?php
    $fruit = array("Apple", "Orange", "Banana", "Grape", "Cherry", "Peach");
    $numElements = count($fruit);
    for($x = 0; $x < $numElements; $x++)
        echo "{$fruit[$x]} ";
?>
```

### The foreach loop

Allows easy processing of entire array without having to pre-evaluate the size of the array.

```php
<?php
    $items = array("Ham", "Eggs", "Bread");
    foreach ($items as $curr)
        echo "$curr<br/>
    ?>
```

- Name of array to process. Loop is based on this array.
- Contents of array indices sequentially copied into this variable.
- If $curr is changed in body of the loop, this does not affect the value stored in $items.
foreach and non-sequential arrays

The foreach loop construct automatically skips over empty array cells. A for loop will not.
http://einstein.etsu.edu/~pittares/CSCI2910/examples/4-1.php

To reference the index of active cells when using a foreach loop, change the "as" clause.

forecah ($array as $index => $contents)

Many useful array manipulation functions


Reading documentation key as some functions behave in ways you may not expect.
http://einstein.etsu.edu/~pittares/CSCI2910/examples/4-3.php

print_r($array) dumps contents of $array to screen (mainly for debugging purposes).
http://einstein.etsu.edu/~pittares/CSCI2910/examples/4-4.php
Array sorting

`sort()` — reorders array contents in alphabetic or numeric order (watch results with mixed arrays).

`rsort()` — reorders array contents in reverse alphabetic or numeric order.

http://einstein.etsu.edu/~pittares/CSCI2910/examples/4-5.php

Visualizing Associative Arrays

<table>
<thead>
<tr>
<th>numerically indexed array</th>
<th>associative array</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Ham   5.40</td>
</tr>
<tr>
<td>1</td>
<td>Eggs  1.75</td>
</tr>
<tr>
<td>2</td>
<td>Bread 1.89</td>
</tr>
<tr>
<td>3</td>
<td>Cheese 2.49</td>
</tr>
<tr>
<td>4</td>
<td>Milk   2.89</td>
</tr>
<tr>
<td>5</td>
<td>Yogurt .89</td>
</tr>
<tr>
<td>6</td>
<td>Salt   1.09</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>index</th>
<th>cell contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5.4</td>
</tr>
<tr>
<td>1</td>
<td>4.7</td>
</tr>
<tr>
<td>2</td>
<td>2.4</td>
</tr>
<tr>
<td>3</td>
<td>1.8</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>8.6</td>
</tr>
<tr>
<td>6</td>
<td>9.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>index</th>
<th>cell contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Ham</td>
</tr>
<tr>
<td>1</td>
<td>Eggs</td>
</tr>
<tr>
<td>2</td>
<td>Bread</td>
</tr>
<tr>
<td>3</td>
<td>Cheese</td>
</tr>
<tr>
<td>4</td>
<td>Milk</td>
</tr>
<tr>
<td>5</td>
<td>Yogurt</td>
</tr>
<tr>
<td>6</td>
<td>Salt</td>
</tr>
</tbody>
</table>
Associative Arrays

Creating: same 2 methods as previous, however an index must *always* be specified.

```php
<?php
    $capitals=array("FL"=>"Tallahassee","GA"=>"Atlanta");
    $capital["NY"] = "Albany";
    $capital["VA"] = "Richmond";
    foreach ($capital as $city)
    {
        echo "<h1>$city</h1>";
    }
?>
```

Incorporating index name into loop

```php
<?php
    $capital=array("FL"=>"Tallahassee","GA"=>"Atlanta");
    $capital["NY"] = "Albany";
    $capital["VA"] = "Richmond";
    foreach ($capital as $state => $city)
    {
        echo "<h1>$state--$city</h1>";
    }
?>
```
Sorting associative arrays

`asort()` reorders array based on values in the array cell contents.
`arsort()` same as above, but in reverse order.
`ksort()` reorders array based on values of keys/indices.
`krsort()` same as above, but in reverse order.

Using PHP and arrays to build form fields

Putting it all together:

PHP arrays often used to populate lists or menus.

```php
<select name="mylist" id="mylist">
  <option value="1">apple</option>
  <option value="2">orange</option>
  <option value="3">grape</option>
</select>
```

Use PHP array and loop to build list dynamically.

http://einstein.etsu.edu/~pittares/CSCI2910/examples/4-7.php
Multi-dimensional arrays

Arrays can be n-dimensional, facilitating more complex data structures.

<table>
<thead>
<tr>
<th>$cart</th>
<th>name</th>
<th>price</th>
<th>quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Adobe Dreamweaver</td>
<td>199.95</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>MS Expression Web 3</td>
<td>149.95</td>
<td>2</td>
</tr>
</tbody>
</table>

Can be thought of as an "array of arrays".

<table>
<thead>
<tr>
<th>$cart</th>
<th>name</th>
<th>price</th>
<th>quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Adobe Dreamweaver</td>
<td>199.95</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>MS Expression Web 3</td>
<td>149.95</td>
<td>2</td>
</tr>
</tbody>
</table>

Coding multi-dimensional arrays

```
<?php
$table[] = array("X","O","O");
$table[] = array("","X","O");
$table[2][0] = "";
$table[2][1] = "";
$table[2][2] = "X";
?>
```

http://einstein.etsu.edu/~pittares/CSCI2910/examples/4-9.php
Helpful array functions

**array_keys($array)** — returns the key (index) field names from $array as an array.

http://einstein.etsu.edu/~pittares/CSCI2910/examples/4-10.php

**explode($delimiter, $string)** — returns array based on splitting $string on the $delimiter.

**implode($glue, $array)** — returns $array as a single string with cells separated by $glue.

http://einstein.etsu.edu/~pittares/CSCI2910/examples/4-11.php