

Exercise 1. Alphabetizing Input.

1. Create a file and name it `alphabetize.php`. Give it all the HTML file headers, a comment with your name and the date, etc.
2. In the file, put a `<form>` with a `<textarea>` field and a Submit button (input field with type of submit).
3. Put a paragraph at the top to give instructions to the user: enter a series of words or phrases separated by commas.
4. This will be an all-in-one form: `alphabetize.php` will display the form, and it will also display the results.
5. When they click on the submit button (to submit the form), alphabetize the words or phrases they entered, and print them out in a sorted list.

Hints

- The `<textarea>` field displays any text it finds between the `<textarea>` tag and the `</textarea>` tag. For this reason, I suggest you put absolutely nothing between those two tags, not even a space or a carriage return. In my code, I did this:

```
<textarea name='user'>$user</textarea>
```

- The above code previously got the contents of the textarea and put it in a variable called `$user`. So now it is displaying what the user entered before they clicked the button.

Pseudo-code

1. Use `explode()` based on a comma to split the user string into an array of words or phrases.
2. Your words might have some spaces before or after, so write a small loop (`for` or `while`) and use `trim()` to remove the spaces from each word. (A `foreach` loop will not work for this purpose.)
3. Use `sort()` to sort the array.
4. Use `implode()` or `join()` to put the array back together into a string.
5. Bonus: if you use `
` to join the strings together, it will display them in a list instead of all on one line.
6. Put your file in a folder named `Chapter06`.

Alphabetize Input

Enter a sequence of words or phrases separated by commas, and I will alphabetize them.

earth,air,fire , water

In alphabetical order:

air
earth
fire
water

Exercise 2: City Distances using Associative Array.

You are going to put the following data into an associative array. You will display the data in an HTML table.

In the Exercise 3, you will create a form where a user can select two cities and it will display the distance. In this exercise, you'll just create the HTML table.

→	Reno	Denver	Boise	Fresno
Reno	0	1036	422	301
Denver	1036	0	817	1142
Boise	422	817	0	721
Fresno	301	1142	721	0

Step 1. Create a data file with the distance information.

1. Create a new file called `citydata.php`.
2. This file will have **only** the distance data in an array. It starts like this:

```
<?php
$distances = array (
    "Reno" => array (
        "Reno" => 0,
        "Denver" => 1036,
        "Boise" => 422,
        "Fresno" => 301
    ),
    "Denver" => array (
        ... more ...
    )
    ... more ...
?>
```

3. This is just a PHP data file; it does **not** have any DOCTYPE, html, head, body, etc.
4. You'll have to fill in the rest of the data.
5. **It's best if the outer array has the same order as the inner array.**
6. Note that this is a two-dimensional associative array (as suggested by the Excel table at the top of this page).
7. You can get the distance using two indexes like this:

```
$distances [ "Reno" ] [ "Boise" ]
```
8. If you make the table properly, you should get the same information when you do this:

```
$distances [ "Boise" ] [ "Reno" ]
```
9. Put the file in the `Chapter06` folder.

Exercise 2 continues on the next page →

Step 2: Create a file to display the mileage information in a table.

1. Create a new HTML web page and name the file `citytable.php`.
2. You will write the PHP code to create the table above.

3. Put this line at the top of your PHP code:

```
#include "citydata.php";
```

4. You will need to use nested loops. You could use a `for` loop or a `while` loop, but I used `foreach` loops.

5. To make the first row of the table, I did this. Note that the variable `$start` contains the key, and that's what we want to put in the heading of the table.

```
echo "<tr>";  
echo "<th></th>"; // one empty cell in the upper left corner
```

```
foreach ($distances as $start => $destination) {  
    echo "<th>$start</th>";  
}
```

```
echo "</tr>";
```

6. When you make the top row and first column of the table, you probably want them to be the same. So even if the data is entered in a slightly wrong order, you'll want to use the outer level of the array to get the contents of the table headers and first column.

7. You can use `array_keys()` to get the keys to display in the first column, but if you use a `foreach` loop like I did above, you won't need to use `array_keys()`.

8. To put the data in the table: use code like this.

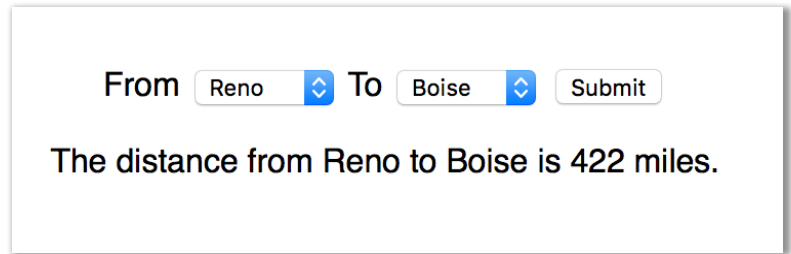
```
foreach ($distances as $start => $destinations) {  
    echo "<tr>";  
    echo "<th>$start</th>";  
  
    foreach ($destinations as $destination => $mileage) {  
        echo "<td>{$distances[$start][$destination]}</td>\n";  
    }  
    echo "</tr>";  
}
```

9. Note that there is a `foreach()` loop inside another `foreach()` loop. The first loop has keys that are starting locations, and values that are themselves associative arrays with the destination cities being keys and the mileage being the value.

10. Put the files in the `Chapter06` folder.

Exercise 3: City Distances Menu using Associative Array.

1. Create an HTML web page and name the file `citymenu.php`.
2. Create a form that has two dropdown menus. A dropdown menu uses the `<select>` tag, and the menu items use `<option>` tags.



3. You can create the first menu with code like this. It has an extra line to retain the selected item after submitting the form:

```
From <select name='from'>
<?php
foreach ($distances as $start => $destination) {
    printf ("    <option %s value='%s'>%s</option>\n",
        ($start == $from) ? "selected" : "", $start, $start);
}
?>
</select>
```

4. Create the second menu with similar code, but check the `$to` value instead of the `$from` value.
5. A select menu has a name on the menu itself, and the options will have the values. So it is possible, if you use GET and the first menu has a name of `from` and the second menu has a name of `to`, you'll get a URL that looks like this:

```
citymenu.php?from=Reno&to=Boise
```

6. You may use either GET or POST.
7. This should be an **all-in-one** form that displays the form and handles the form **in the same file**, `citymenu.php`.
8. Put the file(s) in the `Chapter06` folder.

Exercise 4. Extra Credit.

1. What happens if the data file changes so that the array content in the same, but the entries change order? For example, suppose the first entry is

```
Reno => Reno, Denver, Boise, Fresno
```

2. ... and the second entry is ...

```
Denver => Denver, Reno, Fresno, Boise
```

3. Theoretically, your table and menu should still work the same. But you may find that the table does not. If not, can you modify your table-drawing code to still work?