Ch04: Strings

Objectives

- Formatting strings
- Comparing strings
- Matching and replacing substrings with string functions
- Using regular expressions

What to Know about Chapter 4

- PHP has over 2000 functions and at least one hundred string functions. The best way to to learn any of PHP's functions is to write some code that explores some of the more popular functions. It is also worthwhile to look at the string functions in the PHP manual. Take a quick look and you will see that there is a string function for just about any purpose.
- You may scan/omit the part on the **`printf()`** function. Unless you need this kind of specific functionality in files that you will print from your site you can obtain the same results in almost all cases by using HTML tags.
- We will cover the section on Regular expressions, however, keep in mind that the syntax for regular expressions is difficult and unless you use them on a regular basis (close to daily) you'll find that you can just look up what you need as you go. PHP provides a lot of functions that perform the same functionality as regular expressions and from what I have read, the PHP functions actually perform these tasks faster.
- The **`mail()`** function will not work on your localhost unless you configure the Mercury Mail Server on XAMPP. I have not been able to get this to work in the past but after going online it looks as if It may not be that difficult to set up. I'll try it first before deciding to add it to the assignment. However if you are curious about using **`mail()`**, try modifying the author's code and run it from your Web Host. Your webhost should be configured to handle mail. It it doesn't work right away hold off before becoming frustrated -- some configuration may be needed.

Assignment

File structure

Create a new directory named **ch04** and save it in your **web182\my-code\ch04** folder. Your path should look like

```
web182\my-code\ch04\
```

All of your code for this set of exercises will reside in your **ch04** folder

Create a file named **index.php** and save it in your **ch04** folder. The **index.php** file should have links to the following files

```
email-parser.html
Palindrome-str-replace.html
palindrome-array.html
palindrome-regex.html
my-boot.html
```

Parsing and email

(20 points)

This exercise is not meant to check for all correct emails -- there are libraries of pre-written code for that. It is an exercise to work with strings. Also, there is no string validation for this exercise.

Create a file named **email-parser.html**

This file will contain an HTML form that asks the user to enter an email.

Create a file named **email-parser.php**.

This file accepts the user input and will display the following

- the complete email address
- the username
- the domain
- the top level domain.

For instance, if the user enters: billy@goat.com, the output would be

```
Email: billy@goat.com
User: billy
Domain: goat
TLD: com
```

Functions you will need (you may also use others if you like - there are many ways to solve this problem)

```
strpos()
substr() -- you can also start at the end of a string and work backwards
with this function
```

Palindromes

Read through this set of exercises and do them in order. You will solve the same problem with three different approaches.

Palindrome defined

From Wikipedia: "A palindrome is a word, phrase, number, or other sequence of symbols or elements that reads the same forward or reversed, with general allowances for adjustments to punctuation and word dividers. Examples include "Amor, Roma", "A man, a plan, a canal: Panama", "Race car" and "No 'x' in 'Nixon'"." Feel free to look up some on your own. Some are funny and pretty clever.

There are two basic types of palindromes -- ones that require the case to match and those that do not. In this exercise you will write a palindrome checker that **ignores case**.

Using str_replace

(20 points)

Create an HTML file named **palindrome-str-replace.html** and a PHP file named **palindrome-str-replace.php**.

palindrome-str-replace.html

The **palindrome-str-replace.html** file consists of a text box prompting the user to enter a palindrome and a submit button.

palindrome-str-replace.php

The **palindrome-str-replace.php** file will receive the string from the **palindrome-str-replace.html** file then check to see if it is a palindrome. If it is a palindrome the program will display a congratulatory message otherwise it will let the user know that it is not a palindrome and ask if they would like to try again.

Here are some steps to help solve the problem

- Change the case so the string is all the same case -- upper or lower -- it does not matter
- Remove any spaces. You will want the string to be all smashed together in one long set of characters
- Remove the following punctuation characters
    - Empty space used to separate words

  - Period
  - Comma
  - Apostrophe
  - Single quote
  - Double quote

Test to make sure the string reads the same in both directions. A good test case is the phrase *Madam I'm Adam*

Functions that may help include

```
strcmp()
str_replace()
strrev()
```

You may have to look up some functions online or feel free to use the forum to find out what other students are doing to solve the problem.

Palindrome with an array

(20 points)

- Copy your existing **palindrome-str-replace.html** file and name the new file **palindrome-array.html**.
- Copy your existing **palindrome-str-replace.php** file and name the new file **palindrome-array.php**.
- Change this file so the action in the HTML form points to the **palindrome-array.php** file.
- Alter the code so all of the punctuation marks (be sure to include the space) are stored in an array.
- Loop through the array and remove any punctuation marks found in the palindrome string. You can use a **foreach** or **for** statement.

The output should be the same as the first palindrome exercise where it displays if the text entered is a palindrome or not.

Palindrome using regular expressions

(20 points)

- Copy your existing **palindrome-str-replace.html** file and name the new file **palindrome-regex.html**.
- Copy your existing **palindrome-str-replace.php** file and name the new file **palindrome-regex.php**.
- Change this file so the action in the HTML form in the **palindrome-regex.html** so points to the **palindrome-regex.php** file.

- Alter the code so all of the punctuation marks (be sure to include the space) replaced using a regular expression.You can choose to use **[[:punct:]]** or write out a regular expression that searches for each of the punctuation characters you have been searching.

The output should be the same as the first palindrome exercise where it displays if the text entered is a palindrome or not.

Bootstrap

(20 points)

Bootstrap is an HTML/CSS/JS framework that allows you to create responsive websites. This is my first step into Bootstrap so I tried to break down the code and take the most basic approach and build upon it.

There are two basic ways to use the framework. You can download it to your current working directory or use a stylesheet that points to a CDN (Content Delivery Network). I chose the CDN because all I had to do to get started was link to it in my web page.

If you head out to examples at getbootstrap.com you can see some of the different “out of the box” examples. I’ve chosen Jumbotron. By the way, we will be using lots of Bootstrap specific tags. Don’t worry about what they mean for now. This is just about getting it to work. I have the complete code at the bottom of this section. `Let’s` get started!

Bootstrap is designed to work on a 12 x 12 grid system. You can divide the sections of code into any combination, such as 3 columns that are 4 units wide. Any combination that adds up to 12 will work. I have made this page so it is one big block of 12 units.

Create a new file named **my-boot.html** and store it in your **ch04** folder.

Paste the following code. It is just an HTML5 template and contains a stylesheet linking the Bootstrap CDN.

```
<!DOCTYPE html>
<html lang="en">
<head>
   <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css
"
integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68v
bdEjh4u" crossorigin="anonymous">

   <meta charset="UTF-8">
   <title>My Boot</title>
</head>
<body>

</body>
</html>
```

Next we need need to add some of the bootstrap specific tags. Start with the outer wrapping `<div>` then we will build inside it. Add the following lines inside your `<body>...</body>` tags along with an `<h1>`

```
<div class="container-fluid">
      <h1>WEB 182: Chapter 4</h1>
      <!-- navbar coming here -->

</div>
```

At this point you can run it in your browser to see what it looks like. Nice start.

Let’s add the navigation bar next. At this point it won’t link to any other files but we can fix that later. Add the following code under the `<!--navbar coming here -->` section.

```
<div class="container-fluid">
       <ul class="nav navbar-nav">
             <!--
                 I realize these do not navigate to anywhere on
                 our sites but I wanted to demonstrate how you
                 would do it. Change it to another link to see
                 how it works.
            -->

            <li><a href="https://arstechnica.com/">ArsTechnica</a></li>
            <li><a href="https://css-tricks.com/">CSS-Tricks</a></li>
       </ul>
   </div>
</nav>
<!-- content coming here -->
```

Save it and take another look. Getting better. Let’s finish this with some content.

Add the following code. Place it after the content coming here section and before the very last `<div>` tag.

```
<div class="container">
   <div class="row">
       <!--
             Here is where the grid system is laid out by 12
             Don’t worry about the sm for now
       -->
       <div class="col-sm-12">
           <ul>
               <li><a href="email-parser.html">email-parser</a></li>
                    <li><a
             href="palindrome-str-replace.html">palindrome-str-replace</a> </li>
             <li><a href="palindrome-array.html">palindrome-array</a> </li>
             <li><a href="palindrome-str-replace.html">palindrome-regex</a> </li>
           </ul>
       </div>
   </div>
```

Take one last look.

Here is the complete code.

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>WEB 182</title>
   <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">
</head>
<body>

<div class="container-fluid">

   <h1>WEB 182: Chapter 4 </h1>

   <nav class="navbar navbar-default">
       <div class="container-fluid">
           <ul class="nav navbar-nav">

                <!--
                      I realize these do not navigate to anywhere on
                      our sites but I wanted to demonstrate how you
                      would do it. Change it to another link to see
                      how it works.
               -->

               <li><a href="https://arstechnica.com/">ArsTechnica</a></li>
               <li><a href="https://css-tricks.com/">CSS-Tricks</a></li>

           </ul>
       </div>
   </nav>

   <div class="container">
       <div class="row">
       <!--
               Here is where the grid system is laid out by 12
               Don’t worry about the sm for now. I put links to
               Files for this week, but you can add any links.
       -->
           <div class="col-sm-12">
               <ul>
                   <li><a href="email-parser.html">email-parser</a></li>
                   <li><a href="palindrome-str-replace.html">palindrome-str-replace</a> </li>
                   <li><a href="palindrome-array.html">palindrome-array</a> </li>
                   <li><a href="palindrome-str-replace.html">palindrome-regex</a> </li>
               </ul>
           </div>
       </div>
   </div>

</body>
</html>
```

This is just the beginning to give you an idea of what Bootstrap can do. I hope to add more as the course continues.

Index.php

(10 points)

Update your `my-code/index.php` file so it includes a link to chapter 4.

Check your `my-code/ch04/index.php` file to ensure it is running as expected.

Upload your Files

(10 points)

Upload your files to your webhost. Check the files on your host to make sure they run as expected.

Add the URL in the Moodle comments when you submit. That way I have an updated link to your sites.

Submitting your work

Add the URL to your site for this assignment in the Moodle comments section

Zip all of the files for this assignment. Name the zipped file ch04-yourLastName.zip and submit it in Moodle.

Sources

- http://getbootstrap.com/
- http://www.w3schools.com/bootstrap/default.asp