Lab 1b - command line

Dates, Operators, Strings, Conditionals, Loops

- Purpose: To write a PHP script using readline, string methods, conditionals, and loops
- Submit: A PHP script to the Learning Hub

The goal of this lab is to create a simple PHP command-line application using strings, and readline() for getting user input

---

Write a PHP script that has the following functionality:

Part 1:

Create an infinite while loop that prompts the user to enter a name (e.g. chris). A valid name must not be empty and may contain a mix of alphabet and numeric characters, but cannot consist only of numeric characters. e.g.:
chris1 (valid)
1chris (valid)
chr1s (valid)
1234 (invalid)

If an invalid name is entered, print a message to the user indicating this, and allow the loop to continue until a valid name is received. Once a valid name *is* entered, break out of the loop.

Part 2:

As above, create an infinite loop but this time prompt the user for a single alphabetic character. A valid response must be a single alphabet character only, no numbers or symbols.

If an invalid character is entered, print a message to the user indicating this, and allow the loop to continue until a valid character is received. Once a valid character *is* entered, break out of the loop.

Part 3:

Check if the character from part 2 is found in the name entered in part 1. Print a message to the user indicating whether the letter was found or not found in the name they entered.

Part 4:

Create an infinite while loop that prompts the user to enter a quote (e.g. to be or not to be). A valid quote is any non-empty response.

If no quote is entered, print a message to the user indicating this, and allow the loop to continue until a valid quote is received. Once a valid quote *is* entered, break out of the loop.

Part 5:

Use explode() to turn the quote from part 4 into an array of strings, using the space character as a delimiter

Part 6:

Create an infinite while loop that has the following behaviour:

```
a. Prompt the user to enter a number corresponding to the word in the
quote they would like to see. The prompt should indicate the valid range
of numbers that can be entered.

    e.g. if the quote the user entered was 'to be or not to be'
(containing 6 words), the prompt should look something like the following:
        Which word in the quote would you like to see? (1-6):

b. If the user enters a number n between 1 and x (where x is the number of
words in the quote), print the nth - 1 word from the array created in step
5 and break out of the loop. Otherwise, allow the loop to continue until
they enter a valid number.
```

Save your script as "lastname-firstname-lab1b.php" (e.g. chris harris would submit chris-harris-lab1b.php) and upload it to the Lab 1 dropbox on the Learning Hub.

Due before the beginning of next week's class