HW 1-1: FUN WITH STRINGS (100/100 points)

Specs: `spec/fun_with_strings_spec.rb`

In this problem, you'll implement three functions that perform basic string processing. You can start from the template `fun_with_strings.rb`

Part A — Palindromes:

A palindrome is a word or phrase that reads the same forwards as backwards, ignoring case, punctuation, and nonword characters. (A "nonword character" is defined for our purposes as "a character that Ruby regular expressions would treat as a nonword character".)

You will write a method `palindrome?` that returns true if and only if its receiver is a palindrome.

As you can see in the template `fun_with_strings.rb`, we arrange to mix your method into the `String` class so it can be called like this:

```
"redivider".palindrome?    # => should return true
"adam".palindrome?         # => should return false or nil
```

Your solution shouldn't use loops or iteration of any kind. Instead, you will find regular-expression syntax very useful; it's reviewed briefly in the book, and the website rubular.com lets you try out Ruby regular expressions "live". Some methods that you might find useful (which you'll have to look up in Ruby documentation, ruby-doc.org) include: `String#downcase`, `String#gsub`, `String#reverse`.

The spec file contains a number of test cases. At a minimum, all should pass before you submit your code. We may run additional cases as well.

Part B — Word Count: Define a function `count_words` that, given an input string, return a hash whose keys are words in the string and whose values are the number of times each word appears:

```
"To be or not to be" # => {"to"=>2, "be"=>2, "or"=>1, "not"=>1}
```

Your solution shouldn't use for-loops, but iterators like `each` are permitted. As before, nonwords and case should be ignored. A word is defined as a string of characters between word boundaries.

Part C — Anagrams:

An anagram group is a group of words such that any one can be converted into any other just by rearranging the letters. For example, "rats", "tars" and "star" are an anagram group.

Given a space separated list of words in a single string, write a method that groups them into anagram groups and returns the array of groups. Case doesn't matter in classifying string as anagrams (but case should be preserved in the output), and the order of the anagrams in the groups doesn't matter.

Browse... No files selected.

```
On Time
palindrome detection
  should work for simple strings [10 points]
  should be case-insensitive [10 points]
  should ignore nonword characters [10 points]

word count
  should return a hash [5 points]
  works on simple strings [10 points]
  ignores punctuation [5 points]
  works on the empty string [10 points]
  ignores leading whitespace [10 points]
  ignores embedded whitespace [10 points]

anagram grouping
  for "scream cars for four scar creams" [10 points]
  sanity checks
    should work on the empty string [5 points]
    should return an array of arrays for nonempty string [5 points]

Finished in 0.01011 seconds
12 examples, 0 failures
```

SUBMIT URL TO PAIRING VIDEO (SCREENCAST)

(10 points possible)

Please submit the URL to an unlisted youtube video recording (screencast) of your pairing session on this assignment below.

?

If you are unable to access YouTube and/or G+, feel free to submit a link to a video hosted on some other service (such as Zoom).

Note: we are hoping to see screencasts with screen sharing plus text chat, or even better, audio chat, but video from webcams are not required.

- Bonus Videos