CS160 Computer Science I
In class lab 7

Objective
Learn about String methods
Work with the Python documentation

Specifics
Perform a simple Google search (or use any other search engine) on "Python String methods". One of the first links will be to the Python webpage detailing the built-in data types. This page contains a description of the methods available in String variables. You may use other webpages for this assignment, but you need to be aware of the official documentation for the language.

The webpage lists methods that are part of every String variable. The methods can return information about the data in the String variable, such as an integer or a Boolean value, or even return a new String value. NONE of the method will alter the value in the variable.

Given a String variable x **identify the function, method, or methods**, which can be used to perform the following tasks. Quite often there are two, and possibly more, methods that could be used perform the tasks. Write the function or method, with the argument(s) (if needed) to perform the tasks. Test this in the Python shell to ensure your answers are correct. For example, if the question was "Determine if x starts with a 'q'", the answer call would be: `x.startswith('q')`

1. Determine how many characters are in x.
   len('123456789') ⟶ 9

2. Determine the position of the first occurrence of the letter "q" in x.
   x.find('q') or x.index('q')

3. Determine if the last character in x is the letter "q".
   x.endswith('q')

4. Get a copy of x in all upper case letters.
   * only if x is already all lowercase: x.swapcase() or x.upper() } works always whether upper or lowercase

5. Is x already comprised of all upper case letters.
   x.isupper()

6. Get a copy of x in all lower case letters.

```
x.lower()
```

7. Determine if x contains any characters at all.

```
x.isascii()
```

8. Get a copy of x with the first blank space (`" "`) changed to a dash (`"-"`).

```
x.replace(" ","-",1)
```

9. Get a copy of x with **all** blank spaces (`" "`) changed to dashes (`"-"`).

```
x.replace(" ","-")
```

10. Break up x into one or more values by a delimiter. For example, if a string contained "a,b,c", how could your program return the values "a" and "b" and "c"?

```
x = "a,b,c"
x.split(",")
```

11. Determine if x "looks like" a number. Remember, the `int()` function crashes if the supplied string doesn't "look like" a number.

will return True if string looks like a number { `x.isdigit()` `x.isalpha()` } will return False if string doesn't look like a number

12. Determine if x "looks like" a name, meaning it contains only the characters `'a'` through `'z'`, either upper or lower case.

```
x.isalpha()
```

tests for letters A-Z