IMPORTANT STRING PROCESSING FUNCTIONS/INFO:

<u>reminders:</u> string + string = "concatenation"
string * integer = "repeated concatenation"

- **string_variable[i]** returns the i'th character of the string
  when i < 0, this functions works from the end of the string, moving to the left

- **other slicing:**

s[4:13] = returns the characters from string s from position 4 (included) to position 13 (excluded)

s[:5] = returns the characters from the beginning of string s up to position 5 in the string (but not including position 5)

s[7:] = returns characters from the 8th character of the string (index 7) up to the end

s[-5:] = returns characters from the 5th to last position (included) to the end

NOTE: if there are no characters that meet the slice requirement ("hello"[100:102]),
a blank string ("") is returned

- **s.upper(), s.lower()** *returns* the uppercase and lowercase of s (doesn’t change s)

- **s.capitalize()** *returns* a version of the string s with the first letter capitalized

- **s.title()** returns a version of the string s with the first character of each word capitalized

- **s.count("i")** returns the number of times "i" occurs in the string s

- **s.replace(substring, stringreplacement)** returns a string with all instances of substring replaced with stringreplacement

- **s.find(substring)** returns the index (where the left-most character = 0) of the first occurrence of substring in s (if it cannot find it, it returns -1)

- **s.rfind(substring)** same as find, but looks for the occurrence of the substring with the highest index ("rfind" = **reverse** find!)

- **s.find(substring, location)** finds the first occurrence of substring, *but starting at the given location*

- **s.strip()** = strips away leading and trailing whitespace from a string

HOW TO LOOP THROUGH ALL CHARACTERS IN A STRING: 2 WAYS

1. USING JUST "in":

- **for letter in s:**
  print(letter) = a loop that set the variable letter to each of the characters in the string s, one at a time

2. USING RANGE AND []:

- **for i in range(len(s)):**
  print(s[i]) = a loop that does the same thing as the previous loop, but allows you access to the numerical position of the character

- **for i in range(len(s)):**
  s[-(i + 1)] = each of the characters in string s **in reverse order**

- **substring in string** = returns True or False, depending on whether the string of characters in substring are contained in string or not

Built-in strings you can access in the string module:
import string
string.**ascii_letters** = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
string.**ascii_lowercase** = 'abcdefghijklmnopqrstuvwxyz'
string.**ascii_uppercase** = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

string.**digits** = '0123456789'
string.**hexdigits** = '0123456789abcdefABCDEF'
string.**octdigits** = '01234567'
string.**printable** = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTU...
string.**punctuation** = '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
string.**whitespace** = ' \t\n\r\x0b\x0c’