

Python String Assignments

For this series of assignments, refer to the [Strings](#) reference page for assistance.

Assignment 1

Prompt for a string to be entered from the keyboard, then print the string to the screen, one character per line.

Save as "string1.py".

```
Enter a word: test
t
e
s
t
```

Assignment 2a

Prompt for a multi-word string to be entered from the keyboard, then print the string to the screen, one word per line.

Save as "string2a.py".

```
Enter some words: this is a sentence
this
is
a
sentence
```

Assignment 2b

Initialize a string to "abc def\tghi\njkl", then print the string to the screen, one "word" per line. Notice that this string contains a space, a tab, and a newline character to separate the words.

When writing this program, use the `string.whitespace` constant to compare to the characters to determine when to print on the next line.

Save as "string2b.py".

Output:

```
abc  
def  
ghi  
jkl
```

Assignment 2c

Amend the previous assignment to use the `split()` method to put the words into a list before printing them to the screen. (Again, one word per line.)

Isn't that much simpler than the previous methods? Can you see any potential problems using this method?

Save as "string2c.py".

Assignment 2d

Also using the `split()` method used in the previous program, write a program that reads in a text file from disk and prints out the total number of words in the file.

Locate some "large" text documents from a source like [Project Gutenberg](#) (here is [Mary Shelley's Frankenstein](#), as a starting point) to test your program on.

Save as "string2d.py".

Output:

```
There are 77986 words in Frankenstein.txt
```

Assignment 3

Write a program that continuously prompts for a string to be entered from the keyboard until one of "y", "yes", "oui", "n", "no", or "non" (in lower or upper case) is entered, then outputs "You entered 'yes'" or "You entered 'no'", as appropriate.

For this assignment, use a list containing each of the test cases, as follows:

```
yesNo = ["y", "yes", "oui", "n", "no", "non"]
```

Your program should loop through each of the test cases to determine if one of those cases was chosen.

Hint: You'll find lower() very useful here!

Save as "string3.py".

Assignment 4

Prompt for a string to be entered from the keyboard, then print a count of the usage of each A-Z character (disregarding case) to the screen. Do not print the count of any characters that are not present in the string (use string.ascii_uppercase to achieve this).

Save as "string4.py".

Assignment 5

As in the previous assignment, print a count of all A-Z characters (again, disregarding case), but this time include other characters such as "!@#\$\$%", etc. Do not print the count of any characters that are not present in the string.

Save as "string5.py".

Assignment 6

Write a program that continuously prompts for words to be entered from the keyboard until "end" is entered in either upper or lower case. The program should then print back all words to the screen, on one line.

Save as "string6.py".

Assignment 7a

Write a program that continuously prompts for words to be entered from the keyboard until "end" is entered in either upper or lower case. For each word entered, print "word is a palindrome" if it is a [palindrome](#), otherwise print "word is not a palindrome". Be sure to disregard case.

Save as "string7a.py"

Assignment 7b

Rewrite your palindrome program so this time it is searching for palindromes in a text file. Again, start with [Mary Shelley's Frankenstein](#), and then try some others from [Project Gutenberg](#). Record each palindrome in a list, then report the palindromes and the number of times each occurs.

Save as "string7b.py".

Assignment 7b_2

Rewrite the previous program, this time researching and incorporating [how to retrieve the text file directly from the Internet](#).

Save as "string7b_2.py".