Lab 1: Strings-n-things

Jan 25, 2018

Main Event

1. From class

Recall the three exercises we discussed in class:

1. Print the chorus of Queen's "We Will Rock You" using only two strings. FYI: The chorus is "we will we will rock you rock you".
2. We saw that `1 / 2 = 0.5`, while `1 // 2 = 0`. Make changes to an expression that uses `/` so that it produces the same value as `//`. Play with different numerator and denominator values to confirm your results.
3. Write a program that rounds positive floating point numbers to the nearest integer. This code should be two lines: the first should assign a floating point number to a variable; the second should print the rounded integer version.

Implement these programs on your own.

Solution:

```
# String operations
print('We will ' * 2 + 'Rock You ' * 2)

# Casting
print(int(1 / 2))

# Rounding
val = float(input('Enter a positive number: '))
print(int(val + 0.5))
```

2. *“Remember the string methods, Luke”*

The following exercises will require you to perform manipulation on strings. Please keep the Python documentation in mind when thinking about string methods.

3. Bob Barker

Create a variable containing the string “BOB! Come get Bob’s bobblehead”:

```
x = "BOB! Come get Bob's bobblehead"
```

Write a program that creates a new variable with all instances of “bob” in `x` replaced with “bob barker”. Capitalization does not matter: “bob”, “Bob”, and “bOB” are identical.

Solution:

```
bob = "BOB! Come get Bob's bobblehead"
barker = bob.lower().replace('bob', 'bob barker')
print(bob, barker, sep='\n')
```

4. String Report

Create another string as follows:

```
x = "The quick brown fox jumps over the lazy dog"
```

Using `x`,

1. Print the lowercase version;
2. Print the uppercase version;
3. Print the string as if it were the title of a book;
4. Count and print the number of vowels in the string.
5. Which letter is the letter z? For example, is it the first character? Second? As was the case with the previous questions, there is a string method to help you out with this.

Solution:

```
sreport = 'The quick brown fox jumped over the lazy dog'
print(sreport.lower(),
```

```
      sreport.upper(),
      sreport.title(),
      sep='\n')
slower = sreport.lower()
a = slower.count('a')
e = slower.count('e')
i = slower.count('i')
o = slower.count('o')
u = slower.count('u')
vowels = a + e + i + o + u
print('Vowel count: ' + str(vowels))
print('z:', sreport.find('z'))
```

5. Madlibs

Make a sentence based on user input. Prompt the user for four inputs (a mix of words and numbers); create a two-line story based on the input. Example:

Input

```
Enter an adjective: purple
Enter another adjective: ancient
Enter a noun: dragon
Enter a number: 2
```

Output

```
The purple bear went into the ancient house.
There she saw a 2 year old dragon.
```

Your story should be different!

Solution:

```
adj1 = input("Enter an adjective: ")
adj2 = input("Enter another adjective: ")
adj3 = input("Enter one more adjective: ")

noun1 = input("Enter a noun: ")
noun2 = input("Enter another noun: ")
noun3 = input("Enter one more noun: ")
```

```
num1 = input("Enter a number: ")
num2 = input("Enter another number: ")

print("The " + adj1 + " " + adj2 + " bear " +
      "went into a " + noun1 + " to get " + num1 + " apples. " +
      "She then entered a " + noun2 + " store " + num2 + " times " +
      "to have a " + adj3 + " " + noun3)
```

6. Interactive addition

Request four numbers from the user using four separate input statements:

```
Please enter the first number: 1
Please enter the second number: 2
Please enter the third number: 3
Please enter the fourth number: 4
```

Concatenate the first two numbers into a single value, and the second two numbers into a single value. Print the sum of these combined numbers. In the example above the correct answer would be 46, since 46 = 12 + 34.

Note that this should work even if the user adds whitespace before or after their input; for example:

```
Please enter the first number:       1
Please enter the second number:   2
Please enter the third number:  3
Please enter the fourth number:          4
```

Should still print 46.

Solution:

```
a = input('Please enter the first number: ')
b = input('Please enter the second number: ')
c = input('Please enter the third number: ')
d = input('Please enter the fourth number: ')

x = int(a.strip() + b.strip())
y = int(c.strip() + d.strip())

print(x + y)
```

Additional Practice

1. Time difference

Write a program that uses three input prompts to obtain the date (day, month, and year) then prints the number of days that have occurred between that date and today.

Python has a library to assist with date/time manipulations: `datetime`. You can use the library to get todays date as follows:

```
>>> import datetime
>>> today = datetime.datetime.today()
>>> another_day = datetime.datetime(2000, 2, 28)
```

`today` and `another_day` are objects (a special data type that we'll learn about later) that happen to have subtraction defined on them:

```
>>> diff = today - another_day
>>> toseconds = diff.total_seconds()
```

`diff` is the difference between the two datetime objects; `toseconds` is that difference in seconds.

In this example, `another_day` was created using an arbitrary date. The program you write should create this date from information obtained from the user. You should then report the difference between today and the user-supplied date in days (or even years, whichever is most comfortable to you). Your program should produce something like this:

```
Please enter a month: 9
Please enter a day: 2
Please enter a year: 2014
That was 1 day ago
```

Pluralization of the output ("day" versus "days") is optional.

Solution:

```
import datetime
from scipy import constants
```

```
month = int(input('Please enter a month: '))
day = int(input('Please enter a day: '))
year = int(input('Please enter a year: '))

today = datetime.datetime.today()
yesterday = datetime.datetime(year, month, day)
diff = today - yesterday

days = diff.total_seconds() / constants.day

print('That was ' + str(days) + ' days ago')
```

Introduction to Computer Science

Introduction to Computer Science
jerome.white@nyu.edu

jerome-white

Learning computer science concepts through practice.