Lab 2.3 Variables and State

Introduction.

I would suggest that you use the directory "week02", that you used for the other labs. For the file names, I put the prefix lab2.3.X- in front of each of the solutions so that they are in order in the directory, you can do that as well if you wish.

Types

1. Create a file called testTypes.py, in the file create 5 variables one for each of the following types:
   - int
   - float
   - boolean
   - str
   - list (look at w3schools.com)

   Use the **`type()`** function to check that the variables are of that type. Your program should output.

```
variable i is of type:<class 'int'> and value:3
variable fl is of type:<class 'float'> and value:3.5
variable is is of type:<class 'bool'> and value:True
variable memo is of type:<class 'str'> and value:how now Brown Cow
variable lots is of type:<class 'list'> and value:[]
```

```
i = 3
fl = 3.5
isa = True
memo = 'how now Brown Cow'
lots = []

print('variable {} is of type:{} and value:{}'.format('i', type(i), i))
print('variable {} is of type:{} and value:{}'.format('fl', type(fl), fl))
print('variable {} is of type:{} and value:{}'.format('is', type(isa), isa))
print('variable {} is of type:{} and value:{}'.format('memo', type(memo), memo))
print('variable {} is of type:{} and value:{}'.format('lots', type(lots), lots))
```

'lots' is a string that contains the letters l, o, t and s, not the variable lots

This is the variable lots

2. Write a program (sub.py) that reads in two numbers and subracts the first one from the second one. The program should look like this when it is run.

```
Enter first number: 10
Enter second number: 4
10 minus 4 is 6
```

Answer

```
# Program to subtract on number from another.

# input reads in a string so we need to convert it into an int
# so we can perform mathematical operations

x = int(input("Enter first number: "))
y = int(input("Enter second number: "))
answer= x-y
print("{} minus {} is {} ".format(x, y, answer))
```

Extra: when the program is running, try entering in something that is not an int eg 1.1 or hello

3. Write a program (div.py) that reads in two numbers and divides the first one by the second and give the integer result and the remainder.

```
Enter first number: 10
Enter the number you want to divide by: 3
10 divided by 3 is 3 with remainder 1
```

Not 3.3

Answer

```
# program that reads in two numbers and
# outputs the integer answer and remainder

x = int(input("Enter first number: "))
y = int(input("Enter the number you want to divide by: "))
answer = int(x//y) # // gives the int division
remainder = x%y    # % gives the remainder

print("{} divided by {} is {} with remainder {}".format( x, y,
answer, remainder))
```

4. Write a program (randomGenerator.py) that prints out a random number between 1 and 10. You will need to import the module random.

```
# program that prints out a random number between 1 and 10

import random

number = random.randint(1,10)
print("here is a random number {}".format(number))
```

Extra try modifying the program so that the user inputs the range (more information on the random module can be found here

5. Write a program (normalise.py) that reads in a string and strips any leading or trailing spaces, the program should also convert the string to lower case. The program should also output the length of the input and output strings. See Python - String Methods (w3schools.com) for more information of string methods

```
Please enter a string:          Some StRiNg
That String normalised is :some string
we reduced the input string from 57 to 11 characters
```

Answer

```
# This program reads in a string and strips
# any leading or trailing spaces.
# It also converts all the letters to lower case
# this program also outputs the lenght of the original string
# and the normalised one

rawString = input("please enter a string:")
normalisedString = rawString.strip().lower()

lenghtOfRawString = len(rawString)
lenghtOfNormalised = len(normalisedString)

print("That String normalised is :{}".format(normalisedString))
print("we reduced the input string from {} to {} characters".format(
lenghtOfRawString, lenghtOfNormalised ))
```

Looking ahead: Lists and tuples

6. Write a program (randomfruit.py) that prints out a random fruit.

```
> python randomfruit.py
A Random Fruit:Banana
> python randomfruit.py
A Random Fruit:Apple
```

Answer

```
# This program prints out a random fruit

import random

fruits = ['Apple', 'Orange', 'Banana', 'Pear']

# we want a random number between 0 and lenght-1
index = random.randint(0,len(fruits)-1)

fruit = fruits[index]
print("A Random Fruit:{}".format(fruit))
```

list is square brackets [ ]

There is a neater way of doing this, the random module can pick a random choice from a list, see later weeks.

And for this example we should have used a tuple, because we don't change the list

Modify the program in 6 (randomFruit2.py) so that it uses a tuple ( ) not a list [ ]. The functionality of the program will not change.

```
# This program prints out a random fruit

import random

fruits = ('Apple', 'Orange', 'Banana', 'Pear')

# we want a random number between 0 and lenght-1
index = random.randint(0,len(fruits)-1)

fruit = fruits[index]
print("A Random Fruit:{}".format(fruit))
```

Tuple is round brackets ( )

Extra: (Questions that I don't give the solution for)

7. Why does this expression cause an error? How can you fix it?

```
message = 'I have eaten ' + 99 + ' burritos.'
print (message)
```

8. Why is eggs a valid variable name while 100 is invalid?
9. What three functions can be used to get the integer, floating-point number, or string version of a value?