LAB 2: Variables in programs

We said that we use computer applications to process data. So far, the data we have processed in our programs are *literal* data. By literal, we mean they are the actual data. Literal data cannot be modified.

Computer programs that are only capable of processing literal data offer little use to the user. For example, a greeting program is only capable of saying 'hello' when the user sometimes wants 'Good morning'. To have the flexibility of processing user provided data, we need to use variables.

What is a variable?

A variable is a name associated with a value. Once a variable is been assigned a value, the variable can be used in place of that value. More importantly, a variable's value can be changed in the program, which gives us the flexibility of processing user data on the fly.

When you want to refer to a value in your program, you can either refer to the literal value itself ("hello", for example), or a variable that is associated with the literal value:

```
>>> print("hello")
hello
>>> greeting = "hello"
>>> print(greeting)
hello
```

The first print statement uses the **literal string** value "hello" for printing to the screen.

In the second print statement, the name greeting is a **variable**, which represents the literal string value "hello".

A variable represents a storage in computer memory. You can think of a variable as a box – what is stored in the box can be changed - hence the name variable.

In python you create a variable by giving a value to a name. The statement is called "assignment statement", which assigns the string value "hello" to the name variable named "greeting". The statement greeting = "hello" is called the **assignment statement**. We use assignment statement to create variables.

Assignment statements follow the pattern of

variable name = literal value or variable name

One thing to remember about assignment statement is that the one to receive is always on the left with the one to give on the right, for example, greeting = “hello”.

It is common for beginners to confuse the two sides, especially when using variables on both sides.

One way to remember this is to try one variable and one literal 5, because if you do 5 = num, you’ll get an error (literal cannot be changed)

A variable sometimes is also called an *identifier*.

```
>>> "hello" = "something else"
SyntaxError: can't assign to literal
```

Rules of naming an identifier

Identifier naming rules:

- Must not be a keyword
- May only contain alphabet, digits, or underscores
- Must not start with a digit

Make sure that the variable name explains the variable it stands for instead of using random characters. For example,

- To store 2 numbers, you may want to use variables **num1** and **num2**.
- To store the total, you may want to use variable **total** instead of just num
- To store a greeting message, you may want to use a variable **greeting** or may be **message**

It is good practice to use either **camelCase** (firstName), or separated by an underscore (first_name). Do not use single letter variable names, unless the single letter makes sense for the context.

Get user input as string

Besides using literal string values to assign variables, we can also use user-entered values and assign them to variables. To read user input from the keyboard, you can use the built-in **input([prompt])** function, as follows:

```
>>> greeting = input("Please enter the greeting: ")
Please enter the greeting: hello
>>> print(greeting)
hello
```

The input function reads a line of input, converts it to string (stripping the newline character at the end, which the user has to type to signify the end of their input), return the string.

Your task: ask the user to enter his name, and then display back 'hello' followed by the name.

Common string functions in python

To see what string operations are available, try any of the following:

- In the shell, type in dir(); dir(_builtins_); dir(_builtins_.str);
- Check on https://docs.python.org/3/library, then click Text_Sequence Type – str.
- Use *intellisense*. Type "hello"_ you will be presented by a dropdown list of functions that are available for string types.

Common string functions

```
>>> "hello world".capitalize()
'Hello world'
>>> "hello world".title()
'Hello World'
>>> "hello world".count("l")
3
>>> "hello world".count("a")
0
>>> "hello world".count("H")
0
>>> "hello world".startswith("H")
False
>>> "hello world".endswith("d")
True
```

String formatting functions

```
>>> "hello world".center(20)
'    hello world     '
>>> "hello world".center(20, "*")
'****hello world*****'
>>> "hello world".rjust(20)
'         hello world'
>>> "hello world".rjust(20, ".")
'.........hello world'
>>> "hello world".ljust(20)
'hello world         '
>>> "hello world".ljust(20,".")
'hello world.........'
```

Formatting templates

```
>>> "restrict decimal places: {0:6.2f}".format(1.1418)
'restrict decimal places:   1.14'
>>> "restrict decimal places: {0:6.3f}".format(1.1418)
'restrict decimal places:  1.142'
```

using str.replace() function

```
>>> original = "hello world"
>>> modified = original.replace("world","today")
>>> print(original)
hello world
>>> print(modified)
hello today
```

We can also allocate the word (or sentence) to be replace as a variable. Try the following:

```
>>> original = "hello world"
>>> replacement =input("Please enter the new word: ")
Please enter the new word: tomorrow
>>> modified = original.replace("world", replacement)
>>> print(modified)
hello tomorrow
```

TASK 1: Modify the following puzzle by asking the user for an input. The value entered will be used to replace John Smith in the below story

John Smith arrived to a small town on Friday. He stayed there for two days and left on Friday. How is this possible?

Answer: puzzle = "John Smith arrived to a small town on Friday. He stayed there for two days and left on Friday. How is this possible?"

newName = input(

newPuzzle = puzzle._____

print( )

TASK 2: Create a script called **Puzzle.py** to do the above task.

Bonus Task: Modify Puzzle.py such that no matter how the user enters the name (all lower, all upper, or a combination), the first letter of the name (i.e. for the first, middle, last names) should be displayed in capital letters.

TASK 3: Create a script called **formatExample.py** to ask the user to enter a sentence. The program should then display the message in lower case. And in the next line is all upper case. And in the next line as a title.

Answer: userText = input(

print(userText.) # to display the message in lower case

print(userText.) # to display the message in upper case

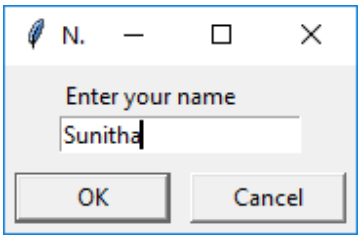
print(userText.) # to display the message as a title

Python GUI programming with tkinter

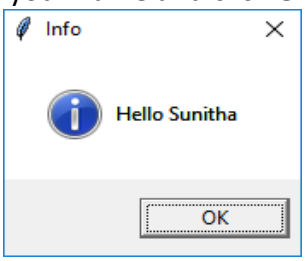
TASK 4: Create a new script **Lab2Gui1.py** and type in

```
from tkinter import *
window = Tk()
name1 = simpledialog.askstring("Name", "Enter your name")
messagebox.showinfo("Info", "Hello " + name1)
window.mainloop()        #launch the window
```

Run the script. You should come up with a window as shown below



Type in your name and click **OK**. You should get the following message window.

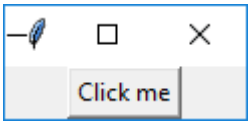


TASK 5: Create a new script **Lab2Gui2.py** and type in

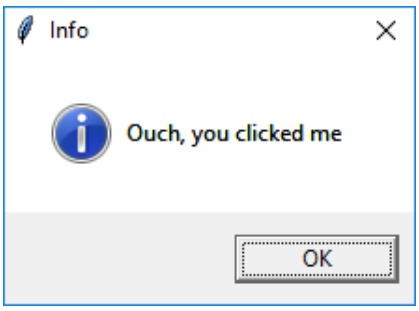
```
from tkinter import *
def processBtn():
    messagebox.showinfo("Info", "Ouch, you clicked me")

window = Tk()
btn = Button(window, text = "Click me", command = processBtn)
btn.pack()           # bind the button to the window
window.mainloop()       #launch the window
```

Run the script. You should come up with a window as shown below



Click on **Click me** button. You should get the following message window.



TASK 6: Create a new script **Lab2Gui3.py** and type in

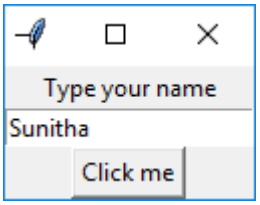
```
from tkinter import *
def processBtn():
    name = txt_in.get()
    messagebox.showinfo("Info", "Hello " + name)

window = Tk()
lb = Label(window, text = "Type your name")
lb.pack()

sv=StringVar()
txt_in = Entry(window, textvariable=sv)
txt_in.pack()

btn = Button(window, text = "Click me", command = processBtn)
btn.pack()              # bind the button to the window
window.mainloop()       #launch the window
```

Run the script. You should come up with a window as shown below



Type in your name and click the **Click me** button. You should get the following message window.