Lab 2: variables, types, and operators

All work should now be done in PyCharm, if you are still having trouble setting up your IDE please contact me at the start of the lab.

Refresher quiz 📝

What kind of language is Python?

- ○ A compiled language.
- ○ An interpreted language.

Check

1. VARIABLES 👩‍💻

In this task you will create numeric variables, focusing on datatypes.

I. Create a script in your project to which you will add comments including your name and a description at the start of the file.
II. Create a variable for a flavour of ice-cream and give it the value "Vanilla"
III. Create a variable for the price of the ice-cream scoop and give it the value 2.35
IV. Create a variable for the number of ice-creams scoops and give it the value 3.
V. Print the details of all variables to the screen.
VI. Print out the type of each of the variables that are used in the script using the `print()` and `type()` functions. Hint: `print(type(variable_name))`
VII. Use the multiply operator * to calculate the total cost of the ice-creams.
VIII. Display the total cost of the ice creams to the screen.

SELF-CHECK – DISCUSS WITH YOUR NEIGHBOURS ❓

- ❖ In Python, what type is used for decimal numbers?
- ❖ In Python, what type is used for text?
- ❖ In Python, what type is used for whole numbers?

2. RE-ASSIGNING VALUES TO VARIABLES 👩‍💻

Create a script that has a variable called max_speed_limit that is assigned a value of 120. Use print to produce the following output.

```
The speed limit is 120 km
```

SELF-CHECK – DISCUSS WITH YOUR NEIGHBOURS 🤔

- ❖ What type do you think max_speed_limit is?
- ❖ Verify using the `type()` function.

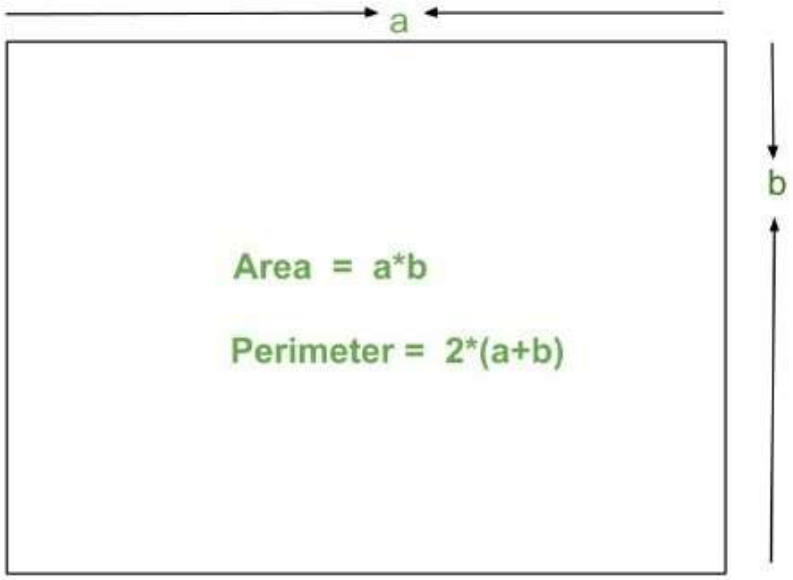
After the print line in your script modify the value stored in max_speed_limit to be 100 and add another print message.

The output of the program should now be:

```
The speed limit is 120 km
The speed limit is now reduced to 100 km
```

3. RECTANGLE 👩‍💻

I. Create a variable for the length of a rectangle and give it the value 10.6.
II. Create a variable for the width of a rectangle and give it the value 3.4.
III. Write code to calculate the area of the rectangle and store it in a variable.
IV. Write code to calculate the perimeter of the rectangle and store it in a variable.
V. Add code to display the area and perimeter of the rectangle in a neat format.



4. CIRCLE 👩‍💻

Calculating the area of a circle requires the use of a mathematical constant called pi. Pi can be approximated as 3.14, but for some applications, it is important to have a more accurate estimate. In this exercise, we will calculate the area of a circle using a variable called pi which is defined in the math module. $A = \pi r^2$

Note: in python, there are add-on modules that can be included in your python script. These modules often have additional functions and variables defined.

I. At the start of a new python script (just after your comments for author and purpose) import the math module

```
import math
```

II. Create a variable for the radius of the circle and give it the value 1.5
III. Write code to calculate the area of the circle and store it in a variable. The value for pi in the math module can be accessed by writing `math.pi`
IV. Print the area of your circle

5. TRIANGLE 👩‍🎨

The length of the base of a triangle is 10.5 and its height is 8.5.
Calculate and display the area of the triangle.
Note: The area of the triangle is (base x height) / 2.

6. IKEA QUESTION 👩‍🎨

A logistics firm only allows full vans to leave their depot. Each of their vans can hold 50 of Ikea's Billy Bookcases.
Write an application that asks the user for the number of Billy Bookcases to be delivered and displays the number of full vans along with the number of bookcases that will be left behind.
Note: there are two different operators that facilitate division in Python (/ and //). What is the difference between these two operators? Which one is more appropriate to use here?

7. STRING VARIABLES 👩‍🎨

After this task you should know:

- ❖ How to add strings
- ❖ How to get Python to display quotes, either single (') or double (")

7a. ADDING STRINGS (CONCATENATION)

Create a new script and add the following code to it:

```
firstname = 'fred'
surname = 'fox'
at_symbol = '@'
dot = '.'
domain_name = 'gmail.com'
```

Add these stings together to form a valid email address, store the result in a new variable, and print it.

This is called **string concatenation**.

7b. PRINTING SPECIAL CHARACTERS

I. Create a new script

II. Add the following `print()` statements

```
print("'I can print single quotes'")
print('"I can print double quotes"')
print('He said "You\'re the best!"')
```

Before you run your script, take a second to think about what these lines of code will print.....
The first two print statements should be easy to figure out.
The last one, however, is a bit more challenging. It needs a quote inside a quote so we use \'' to tell Python to print it.

III. Print the following using a new line sequence ('\n') and a tab sequence ('\t')

```
print("the\ncat\nin\nthe\nhat")

print("the\n\tcat\n\t\tin\n\t\t\tthe\n\t\t\t\that")
```

Be sure you understand what is happening here and if not then please ask

ADDITIONAL EXERCISES 📚

If you are finished the above exercises try your hand at the following exercises:
Lab_week_2_additional_exercises.docx
(https://cit.instructure.com/courses/108475/files/3033180?wrap=1) ⬇
(https://cit.instructure.com/courses/108475/files/3033180/download?download_frd=1)

GROUP EXERCISE 🤸 (please wait for me to initiate this exercise)

In exercise 5, you wrote code to calculate the area of a triangle. If you were a lecturer and you were in charge of evaluating student's code how would you assign marks for this exercise?

Consider the importance of code functionality (does the code do what it is meant to do?), code readability (choice of variable identifiers, comments, clear layout).

You have 10 marks to give for successful completion of this exercise, create a rubric for these 10 marks. Write your completed rubric down and clearly define each gradable component and the marks assigned to it.

Once you have completed your rubric discuss it with your neighbour. After your discussion use the rubric to evaluate your neighbour's code. Ensure you point out to good things that they did that gained them marks. Make a note of the improvements they could make in future iterations of their code.