

## Exercises – Week 1. Introduction to Python, Variables and Operators

### Introduction

#### Getting Started

- To open Anaconda on a Linux lab computer
  - Click on Activities in the top left corner of the screen.
  - Open 'Terminal' from the menu tab.
  - Type or copy and paste the following:  

```
/opt/anaconda/2020.07/bin/anaconda-navigator
```
  - Press enter
- [If completing these exercises off-campus/on your personal laptop, first install Anaconda by following the instructions in the links on blackboard **or** use remote desktop to access a lab computer, then choose Anaconda from the programs menu.]
- Launch Spyder IDE from the Anaconda Navigator window by clicking on the Launch button next to the Spyder logo - we will use Spyder to write and run Python code.
- Complete the exercises using the Spyder IDE, save your work as a .py file.
- Try to complete all of the **Essential** questions in the lab.
- Complete any unfinished **Essential** questions for homework before the next class.
- The **Advanced** questions are optional. You may attempt these if you have finished the Essential questions.

#### Creating, saving and opening Python files in the Spyder IDE

- To create a new Python file click “File” and then “New File”.
- Write some code.
- To save a Python file click “File” and then “Save”.
- If this is the first time you are saving the file, you will be prompted to choose a file name and location. Name your file and save it somewhere appropriate with .py extension.
- Next time you open Spyder, you can access the file by clicking on “File” then “Open”, and then navigating to where your file is stored.

## Rules for naming variables

- Variables can be assigned values
- Variable names may contain letters or numbers
- Variable names must begin with a letter
- Variable names are case sensitive (**time** is not the same as **Time**)
- The value of the variable can be re-assigned. If we have `x = 1` in a program followed by `x = 2`, the original value of `x` will be overwritten with the new value, 2.
- We can also assign multiple variables on the same line.

```
x, y = 5, 10
```

- Some **keywords** are reserved by the Python language and cannot be used as variable names. For a full list of keywords reserved by Python, enter the following in Spyder and press Run:

```
import keyword
print(keyword.kwlist)
```

- Use a consistent naming convention such as **snake\_case** when naming variables.

## Part 1 - Variables

### Exercise 1 - Numbers and Arithmetic Operators (Essential)

Python can be used as a calculator. You can input operations, and store the results of operations as variables for use in additional calculations.

1. Create two variables, `a` and `b`, and assign a numerical value of your choice to each of these variables.
2. Calculate the sum of `a` and `b` and print the result in the Console window in Spyder.
3. Calculate the product of `a` and `b` and store the result as a new variable.
4. Overwrite the value of the variable you just created with the value  $\frac{a+b}{3}$ .  
Hint: Python follows the same ordering of mathematical operations as any other calculator.
5. Find the remainder when `a` is divided by `b` and print the result..
6. In a single line of code, create 3 variables for the length, width and height of a cuboid. Then, write a program to store the volume of the cuboid as a new variable.

## Exercise 2 - Strings (Essential)

Strings are text data.

1. Create two variables `c` and `d`. Assign the value 'Hello' to `c` and the value 'World' to `d`.
2. Use arithmetic operators and these variables to create a new string variable with the value: 'Hello World'
3. Print the third letter of `c`.
4. Print the last three letters of `d`

## Exercise 3 - Modelling with Variables (Essential)

1. Write a program that finds the volume of a sphere and the area of the cross section that passes through its centre, using its radius, and then displays these values.

## Exercise 4 - More Variables (Advanced)

1. Practise expressing some simple mathematical expressions of your own using arithmetic operations.
2. What happens if you use arithmetic operators on Boolean (`True` or `False`) values?

## Exercise 5 - Comparison Operators (Essential)

Comparison Operators output a `True` or `False` (Boolean) value.

Use variables `a` and `b` (from Exercise 1), and `c` and `d` (defined in Exercise 2) for these exercises.

1. Write a line of code that prints `True` in the Console window in Spyder if variable `a` is greater than `b`.
2. Write a line of code that prints `True` if the `type` of `a` is the same as `b`. Can you change the value of `a` and/or `b` to output a different result?
3. Write a line of code that prints `True` if the first letter of `c` is the same as the first letter of `d`. Change the value of `c` and `d` to check your code works as expected.

## Exercise 6 - Logical Operators (Essential)

Logical Operators output a `True` or `False` (Boolean) value.

Use variables `a`, `b` Exercise 1 for these exercises.

1. Write a line of code that prints `True` in the Console window in Spyder if both variables `a` **and** `b` are even.
2. Write a line of code that prints `True` if either variable `a` **or** `b` is even.
3. Create three variables in a single line of code. Each variable should be the name of a student and the value of the variable should be their score in an imaginary assignment

e.g `Valentina = 75`. The pass mark for the assignment is 40. Write a program that outputs `True` if any of the three students passed the assignment.

## Exercise 7 - Modelling with Operators (Essential)

1. Write a program that compares the volume of a sphere and the volume of a cuboid and outputs a message telling the user which is larger e.g.  

```
'The volume of the sphere is greater:  True'
```

```
'The volume of the cuboid is greater:  False'
```
2. Write a program that sorts animals into three categories: herbivores, carnivores and omnivores based on two facts about the animal: (1) whether or not it eats meat and (2) whether or not it eats plants.

## Exercise 8 - More Operators (Advanced)

1. Create a variable with a numerical value. Write a program that prints `True` if the number is a square number or perfect square (i.e. the product of some integer with itself).
2. Write a program, like the example shown in today's class, based on your own typical day, that tells you what action to do based on the time of day.

## Checklist

- Check that you understand the basics: variables, different types of variables (numbers (integers, floats), Booleans, strings), the different types of operators, and how these work with both numbers and strings.
- Finish any incomplete Essential exercises for homework.
- Attend the drop-in session for one-to-one support from a Teaching Assistant if there was anything you didn't understand.