

## Exercises – Week 1. Introduction to Python

### Getting Started

- Install Anaconda by following the instructions in Tutorial Video 1.2
- Familiarise yourself with the basics of using Spyder (part of Anaconda) by watching Tutorial Video 1.3
- The exercises on this sheet are designed to give you a very gentle introduction to writing Python code to prepare you for the course.
- Complete the exercises using the Spyder IDE, save your work as a .py file. you should have the file ready to discuss with the TA and other group members at the tutorial on Friday
- You can discuss any questions you have or any issues with installing Anaconda with a teaching assistant at the drop-in session on Wednesday or the group tutorial on Friday.
- Don't worry if there is something you don't understand. The exercises are to give you some practise, but we will go over most things again in the lectures.

### Exercise 1 - Variables

1. Variables can be assigned values and (may) change during the execution of your program. To assign a value to a variable, we just use a single “equals” sign: =. Try:

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

What is returned now when you call `x` and `y`?

2. What happens if you now assign `x = 4`?
3. Variable names should start with a letter, and may contain letters or numbers. Be aware that some **keywords** are reserved by the Python language and cannot be used as variable names. Try the following:

```
True = 1
```

What happens here, and why? Now try:

```
true = 1
```

**Note:** Python is case-sensitive, so be careful when naming and using your variables! For a full list of keywords reserved by Python, enter the following:

```
help("keywords")
```

- (\*) We can also assign multiple variables on the same line.

`x, y = 5, 10`

How would you extend this to also assign 15 to the variable `z`?

## Exercise 2 - Numbers and Operators

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

- Create two variables called “A” and “B”, and assign a value of your choice to each of these variables.
- To calculate the addition of A and B, enter `A+B`.
- Now enter `A*(A+B)`.
- What happens if you enter `A*A+B` instead? Python follows the same ordering of mathematical operations as any other calculator.
- Set `A = 10` and `B = 3`. Now enter `A/B` to calculate the division of A and B. What happens if you enter `A//B` instead?
- Now try `A%B`.

**Note:** `%` is the modulus operator, and calculates the remainder of a division.

- (\*) Can you calculate the circumference of a circle with a 5cm radius? How about a 12.5cm radius?

**Hint:** Recall in the lectures we used the `math` module to gain extra functionality. It also gives us constant values such as `math.e`.

**Exercise 3 - Strings** When the value of a variable is a character or sequence of characters, we call it a string.

- Create two variables called “A” and “B”.
- Assign A to “Hello” and B to “world”. The quotation marks indicate that these are strings. You can check this with `type(A)` and `type(B)`.
- Join these variables together by *adding* A and B: `A + B`
- What happens when you try `A - B`?  
**Note:** Not all operators are defined for all variable types. For example subtraction does not apply to strings.
- Notice how the string is missing a space between “Hello” and “world”. Combine A and B with a new space “ ” in the middle, and assign this new string to a new variable C.
- Print the length of this new string by entering `len(C)`.
- `len()` is a built-in function in Python which simply returns the *length* of something. What types of variables does `len()` work on?

**Hint:** Look at the Python 3 Documentation: <https://docs.python.org/3/library/functions.html>.

8. (\*) You can also check if a certain phrase or character is present in a string by using the keywords `in` or `not in`, which return a boolean.

Using the string `C` from above, try:

- `"w" in C`
- `"hello" in C`
- `"Hello" in C`
- `"world" not in C`
- `A in C`

**Exercise 4 - Booleans** Booleans are variables with the value `True` or `False`. When we compare to variables, for example `is A greater than B?`, written `A < B`, the outcome will be `True` if, for example `A = 5` and `B = 2` and the outcome will be `False` if, for example `A = 2` and `B = 4`.

1. Create two variables called “A” and “B”, and assign some numeric values to each of them.
2. Print the results of:
  - `A < B`
  - `A > B`
  - `A == B` (*Note: Two equals signs*)

You can also use `<=` for  $\leq$  and `>=` for  $\geq$  comparisons.

3. What happens if you do use `A=B` with only a single equals sign?
4. What happens if you write `not` in front of one of the lines above?
5. You can also set `A` and `B` to be Boolean truth values themselves, such as: `A = True` and `B = False`.

Now try some logical (Boolean) operations such as:

- `A and B`
- `A or B`
- `A and not B`

7. (\*) You can also use the `bool()` function to evaluate any value as either `True` or `False`. What numbers evaluate as `True`? Do any evaluate as `False`?

## Checklist

- Meet your TA on Friday - they will be mentoring you throughout the course, and will be there to answer question and provide feedback along the way.
- Check that you understand the basics: variables, different types of variables (numbers (integers, floats), Booleans, strings), the different built-in operators, and how these work with both numbers and strings.
- Practice with using Spyder.