Session 5
Functions

- ✓ Introduction
- ✓ Built in Functions
- ✓ User Defined Functions

Introduction

So far in this course we have discussed features of the language such as types, operators and containers - these are primarily ways of storing data but a program is not very useful if it can't actually work on the data. **Functions** are the name of specialised pieces of code that take an input perform one kind of operation and return an output, for example you may have a function that returns the remainder from a division or logs a user in. Most programs will be made up of many functions and it may be easier to think of functions as miniature programs. Imagine if every programmer had to write their own functions for basic things?

From the examples of previous sessions you may remember seeing some functions, for example the sort function. There are many functions available for common things such as sorting and modifying strings - these are available so that programmers do not need to rewrite common functions as this would lead to slow and unmaintainable code.

In this session we will look at some useful functions available in the Python programming language and how to write our own functions. Writing our own functions is great for making easily readable and maintainable code, additionally it allows us to break down our problem into smaller chunks.

Built-In Functions

In the previous sessions you will have used some of the built in functions such as the sort function and the string format function. This is just a small sample of the functions available in Python.

Note

Functions available in Python may not be available in other languages such as C++ or Java, you should always check the online documentation or a programming reference manual when using a new language.

A list of built in Python functions can be found here - https://docs.python.org/2/library/functions.html. There are descriptions provided for each one and these are the functions that you should try and remember. Some of the built in functions can be used for modifying types for example it is possible to convert a string to an integer using the **int** function; or to convert an integer into a string.

Python contains a lot more functions than those listed on the built in functions page, all of this extra functionality can be access through modules such as the time module or the system module.

```
from time import time
import math

print abs(-100)
print int('345')
print str(123)
print time()
print math.pi
```

In the above example we have used various built in functions as well as using functions from imported modules.

Exercise

To get used to using the built in functions as well as modules, specifically the math module, you will write an application that can take any integer as a user input and perform the following functions and print a value -

- *return N to the power of 2*

- *return the square root of N*
- *divide N by 2 and return the remainder*

Save your application as Exercise_9.py.

User Defined Functions

As well as the built in functions a programmer can write their own functions, this may be to reduce duplicated code or to break down a problem into smaller problems.

Functions in Python are declared as follows -

```
def multiplyNumber(number, multiplier):
    x = number * multiplier
    return x
```

In Python the **def** keyword designates a user defined function, the word after the def is the name of the function followed by the arguments of the function. The arguments can be named anything you like but should try to describe the desired input, for example number and multiplier. After the function name and arguments comes the body of the function, this is where the actual action happens, in this case we multiply two numbers, and finally is the **return** keyword; the return keyword returns the contents of the variable for use by whatever called the function.

Generally user written functions should be small and focused, if they start to become so long that you have to scroll up and down to read it all then that is a good sign that your function is doing too much. Almost as important as the content of the function is the name of the function, typically it will consist of two words with the first word being a verb and the second being a noun. For example we could name a function **loginUser** with login being the verb and user being the noun.

 *Exercise*

Exercise 10 is a big one and will draw on all you have learnt so far from loops to functions.

You will be programming a calculator program that gives the user a set of options that contain the basic mathematical operations and asks the user to choose the operation by selecting a number, after a number has been selected the program must ask for two numbers and perform the operation and give the result.

For example your menu may look like this -

1. Add
2. Subtract
3. Multiply
4. Divide
Your choice:

For extra points you could make the calculator repeat its menu after each computation or ada more complication options to the calculator. Save your program as Exercise_10.py.