# PYTHON PROGRAMMING

Python is  a **text** based **programming language**. That can be used to create programs, games, applications and much more!

A **program** is a set of precise instructions, expressed in a **programming language**. **Translating** the programming language is necessary for a machine to be able to **execute** the instructions.

To execute a Python program, you need a **Python interpreter**.

This is a program that translates and executes your Python program.

A list is where values can be stored. This is a comma-separated list  of values (items) in square brackets.

```
flavours = ["strawberry", "chocolate", "mint",
"cherry","raspberry"]
```

This is an data structure  organised in a structure, each item has its own index indicating its position in the list.
NOTE: List item numbering starts from 0—zero based system

When this code is executed
print  (flavours[2])
Mint will be output as it is looking in the list flavours and  selecting index position 2 to output

**Arithmetic operators** + addition, - difference, * multiplication, / division, // integer division
% remainder of integer division, ** exponentiation  (to the power of)

## Useful snippets of code

| | |
|---|---|
| list.append(item) | Add an item to the end of a list |
| list.insert,index.item) | Inserts an item to a given index |
| list.pop(index) | Remove item at given index and return it |
| list.remove(item) | Remove the first item from the list with a particular value |
| list.index(item) | Search for the index of an item |
| list.count(item) | List the occurrences of the item |
| list.reverse() | Reverse the list |
| list.sort() | Sort the list |

Use an structure , a (**while**) when the program needs to **repeat** actions, while a **condition** is satisfied.

**for loops** are convenient for **iterating** over any sequence of elements

**Walk through** the program keeping track of what is happening  to lists and variables as the loops are executed.