Chapter 2 : Control Structure

Python has three types of control structures:

- Sequential – default mode
- Selection – for decision making and branching
- Repetition – for looping/multiple calculations

2.1 Sequential

- Set of statements whose execution process happens in a sequence. The problem with sequential statements is that if the logic has broken in any one of the lines, then the complete source code execution will break

```
1  a = 10.0
2  b = 15.0
3
4  x = b-a
5
6  print('x=',x)
```

Example 2.1 : Sequence list

2.2 Selection

- Selection statements are also known as Decision control statements or branching statements.
- The selection statement allows a program to test several conditions and execute instructions based on which condition is true.
- Some decision control statements are:
    - if
    - if-else
    - nested if
    - if-elif-else
- This selection usually written in code block and can be written as a subprogram to test decision making control.
- When you write for example and if block, it should followed by : and the rest of the code will be indented meaning the selection rule is govern by this block.

2.2.1 if block

```
1  a = 1.0
2  b = 2.0
3  n = a*b
4
5  if n > 0:
6     print('n is larger than 0')
7
```

- This program using if block to test if our condition or decision is correct based on our mathematical operation.

2.2.2 if-else

```
a = 1.0
b = 2.0
n = a*b

if n > 0:
   print('n is larger than 0')
else:
   print('n is smaller than 0')

```

- For if-else loop, we can set to choose which condition is correct by setting two separate condition that fulfil our answer.

2.2.3 nested-if

```
a = 1.0
b = 2.0
c = 5.0
n = a*b

if n > 0:
   if a > c:
      print('The value of a is big')
   else:
      print('The value of a is small')
```

- In nested-if, there are two if block, one after the other and both need to fulfil the condition if the values are both correct. If not, it will go out from the nested if and go to the else block.

2.2.4 if-elif-else

```
a = -1.0
b = 2.0
c = 5.0
n = a*b

if n > 0:
   print('The of n is positive')
elif n< 0:
   print('The of n of a is negative')
else:
   print('The value of n is zero')
```

- For if-elif-else block, we can set three separate condition that fulfil our answer according to our input.

2.3 Repetition

- A repetition statement is used to repeat a group(block) of programming instructions.
- In Python, we generally have two loops/repetitive statements:
    - for loop
    - while loop

2.3.1 for loop

```
1  i = 0
2  n = 100
3
4  for i in range(n):
5      i=i+1
6      print(i)
```

- For loop is a used to compute a loop or iteration of calculation. We can set how many iteration or loop that we can using for and i is the initial condition to set our calculation. In many computing language, we use variables like i, j, k, l, m and n to set the initial condition.

2.3.2 while loop

```
1  i = 0
2  n = 100
3
4  while i < n:
5      print(i)
6      i=i+1
7  print('end')
```

- In while loop, we could set the condition, like for but here we use boolen or logical operator to set our condition.

2.4 Extended mathematical, logical and boolean operator

List of operator:

Operator	Meaning	Example
>	Greater than	x > y
<	Less than	x < y
>=	Greater and equal than	x >= y
=<	Less and equal than	x =< y
==	Equal	x == y
!=	Not equal to	x != y
and	Returns True if both statements are true	x < 2 and x < 4
or	Returns True if one of the statements is true	x < 2 or x < 4
not	Reverse the result, returns False if the result is true	not (x < 2 and x < 4)

2.5 Function

- A function is a block of code that performs a specific task.
- You can pass data, known as parameters, into a function.
- A function can return data as a result.
- How to create a function:

  In python, the function is defined by using def keyword

  Example

```
1 def my_function():
2     print('Creating text from function')
3 my_function()
4
```

Output:

```
Creating text from function
```

- Python function is useful if you wanted to create calculation where it can be like a subprogram of your main program.
- For example, we want to solve a polynomial, $f(x) = x^3 + 2x^2 + 5$ where we can change the input, x

```
1 #program to calculate polynomial
2
3 def f(x):
4     return x**3+2*x**2+5
5
6 #input
7 x = 5
8
9 print(x,f(x))
10
```

Output:

```
5 180
```