

UNIT 6

FUNDAMENTAL PROGRAMING

6.1. Program Flow Controls and Syntax in Python

A program flow control is the order in which the program's code executes. In Python, the flow control of the program is regulated by the implementation of three types of programming language constructs or program logic, namely sequential, branching, and loop.

6.1.1 Conditionals Program Flow Controls

In some cases, you may need a program to choose one out of two or more alternatives, depending on the input given.

Conditional Expression

Conditional expressions are statements that use Boolean operators (such as AND, OR, NOT) or comparison operators (such as $>$, $<$, $\neq$).

The comparisons in Table 6.1 are also called conditional expressions that correspond to the Boolean values - true or false.

Conditional or branching statements

Conditionals are statements in a program in which the program decides at runtime whether some parts of the code should or should not be executed.

In Python, a conditional statement is written using the if keyword. Like many other programming languages, Python provides various versions of branching statements that can be implemented using the if clause: simple if statement, if..else statement, and if..elif..else statement.

if statement: this is the simplest implementation of a conditional statement that decides only one part of the program to be executed if the condition is satisfied or ignored if the condition fails. if..else statement: The 'if..else statement' provides two alternative statements or blocks: one following the 'if expression' and another following the 'else clause'. 'if..else' allows us to specify two options: one which is executed if a condition is true (satisfied), and another which is executed if the condition is false (not satisfied).

if..elif..else statement: Assume a problem that could have many options to choose from or require several conditions.

Using 'and' and 'or' with if...statement: The 'and', and 'or' keywords are also used with 'if...statement' in Python. The 'and' and 'or' keywords are logical operators and are used to combine conditional statements.

Using 'or' operator with if... statement: The evaluation of the 'if' expression is True if one of these three conditions is satisfied. Otherwise, the expression evaluates to False, and in that case, it executes the statement under the else clause.

6.1.2 Loops or Iteration Program Flow Controls

Loops are set of statements that run repeatedly for a fixed number of times, or until a condition is satisfied. Loop statements control a block of code to be executed iteratively or until a certain condition fails.

Loops are a useful and frequently used feature in all modern programming languages. There are two types of loops that are built into Python:

for loops: The for loop in python is used to iterate over a sequence.

In Python for loop is used to iterate over the items of any sequence including the python list, string, tuple, dictionary, etc. It is also used to access elements from a container (e.g. list, string, tuple) using a built-in function range(). The general syntax of for loop is as follows.

Syntax: for variable_name in sequence :

statement_1

statement_2

[....]

For loop with range () functions:

range(x): generates a sequence of numbers from 0 to x, excluding x, incrementing by 1.

range(x, y): This generates a sequence of numbers from x to y excluding y, incrementing by 1.

range(x, y, z): This generates a sequence of numbers from x to y excluding y, incrementing by z.

break Statement with for: The term break is a keyword in python. With the break statement, you can stop the loop before it has looped through all the items.

continue keyword with for loop: The term continue is a keyword in python.

With the continue statement, you can stop the current iteration of the loop, and it continues with the next.

while Statement: The while statement in Python supports repeated execution of a statement or block of statements that is controlled by a conditional expression. The general syntax for the while statement is:

Syntax:

while expression:

Statement_1

Statement_2

[...]

6.2. Comments in Python

Comments are descriptive texts that exist in program source code and are ignored

by a compiler and interpreter. Comments are not executable statements or part of the program.

In python, comments are denoted by the hash symbol (#).

6.3. Python Interpreter

Python is an interpreted programming language, not a compiled one although compilation is a step in Python interpreter processes. Compilation is simply a translation that generates byte code that you will more in the next paragraph. The interpreter is the program that is responsible for executing each line of statements.

Python has a wide range of interpreters including IDLE, CPython, Jython, PyPy, and IronPython and online Python editors.