Making Decisions & Functions

if else statement

Syntax

The syntax of the *if...else* statement is −

```
if expression:
statement(s)
else:
statement(s)
```

Nested if else

Syntax of if...elif...else

```
if test condition:
Body of if
elif test condition :
Body of elif
else:
Body of else
```

for loop

Syntax of for Loop

```
for var_value in sequence:
Body of for loop
```

The range() function

We can generate a sequence of numbers using range() function. range(7) will generate numbers from 0 to 7.

We can also define the start, stop and step size as range(start,stop,step_size).

step size defaults to 1 if not provided.

```
print(range(10))

 # Output: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

while loop

Ex 1-

```
count = 0

while count < 5:

print(count)

count += 1
```

Ex 2-

```
count = 0

while count < 50:

    print(count)

    count += 5
```

Function

Functions are a convenient way to divide your code into useful blocks, allowing us to order our code, make it more readable, reuse it and save execution time.

The function definition does not execute the function body; this gets executed only when the function is called.

Syntax of function-

def function_name(parameters):

Statement1

Statement2

………………

Statement N

The Keyword def marks the start of function header.

1. Function naming follows the same rules of writing identifiers in Python.
2. Parameters (arguments) through which we pass values to a function are optional.
3. A colon (:) to mark the end of function header.
4. The return statement can be used to return a value from the function.

Global variables & local variables

Global variables: Variables that are not bound to any function but can be accessed inside as well as outside the function are called global variables.

Local variables: Variables which are declared inside a function are called local variables.