

## UNIT III

### What is Statement in Python

A **Python statement** is an instruction that the Python interpreter can execute. There are different types of **statements in Python** language as Assignment statements, Conditional statements, Looping statements, etc. The token character NEWLINE is used to end a statement in Python. It signifies that each line of a Python script contains a statement. These all help the user to get the required output.

### Types of statements in Python?

The different types of Python statements are listed below:

- Multi-Line Statements
- Python Conditional and Loop Statements
  - Python If-else
  - Python for loop
  - Python while loop
  - Python try-except
  - Python with statement
- Python Expression statements
  - Python pass statement
  - Python del statement
  - Python return statement
  - Python import statement
  - Python continue and
  - Python break statement

### Example:

Statement in Python can be extended to one or more lines using parentheses (), braces {}, square brackets [], semi-colon (;), and continuation character slash (\). When the programmer needs to

do long calculations and cannot fit his statements into one line, one can make use of these characters.

Declared using Continuation Character (\):

```
s = 1 + 2 + 3 + \  
    4 + 5 + 6 + \  
    7 + 8 + 9
```

Declared using parentheses () :

```
n = (1 * 2 * 3 + 7 + 8 + 9)
```

Declared using square brackets [] :

```
footballer = ['MESSI',  
             'NEYMAR',  
             'SUAREZ']
```

Declared using braces {} :

```
x = {1 + 2 + 3 + 4 + 5 + 6 +  
     7 + 8 + 9}
```

Declared using semicolons(;):

```
flag = 2; ropes = 3; pole = 4
```

## **Keywords in Python**

**Python Keywords** are some predefined and reserved words in Python that have special meanings. Keywords are used to define the syntax of the coding. The keyword cannot be used as an identifier, function, or variable name. All the keywords in Python are written in lowercase except True and False. There are 35 keywords in Python 3.11.

In Python, there is an inbuilt keyword module that provides an iskeyword() function that can be used to check whether a given string is a valid keyword or not. Furthermore, we can check the name of the keywords in Python by using the kwlist attribute of the keyword module.

## **Identifiers in Python**

**Identifier** is a user-defined name given to a variable, function, class, module, etc. The identifier is a combination of character digits and an underscore. They are case-sensitive i.e., 'num' and 'Num' and 'NUM' are three different identifiers in python. It is a good programming practice to give meaningful names to identifiers to make the code understandable.

We can also use the Python string isidentifier() method to check whether a string is a valid identifier or not.

## **Rules for Naming Python Identifiers**

- It cannot be a reserved python keyword.
- It should not contain white space.
- It can be a combination of A-Z, a-z, 0-9, or underscore.
- It should start with an alphabet character or an underscore ( \_ ).
- It should not contain any special character other than an underscore ( \_ ).

## **Examples of Python Identifiers**

### **Valid identifiers:**

- var1
- \_var1
- \_1\_var

- var\_1

### **Invalid Identifiers**

- !var1
- 1var
- 1\_var
- var#1
- var 1

### **Python Keywords and Identifiers Examples**

**Example 1:** Example of and, or, not, True, False keywords.

```
print("example of True, False, and, or, not keywords")
```

```
# compare two operands using and operator  
print(True and True)
```

```
# compare two operands using or operator  
print(True or False)
```

```
# use of not operator  
print(not False)
```

### **Output**

```
example of True, False, and, or, not keywords
```

```
True
```

```
True
```

```
True
```

## Memory Allocation in Python

There are two parts of memory:

- stack memory
- heap memory

The methods/method calls and the references are stored in **stack memory** and all the values objects are stored in a **private heap**.

### Work of Stack Memory

The allocation happens on contiguous blocks of memory. We call it stack memory allocation because the allocation happens in the function call stack. The size of memory to be allocated is known to the compiler and whenever a function is called, its variables get memory allocated on the stack.

It is the memory that is only needed inside a particular function or method call. When a function is called, it is added onto the program's call stack. Any local memory assignments such as variable initializations inside the particular functions are stored temporarily on the function call stack, where it is deleted once the function returns, and the call stack moves on to the next task. This allocation onto a contiguous block of memory is handled by the compiler using predefined routines, and developers do not need to worry about it.

### Example:

```
def func():
```

```
    # All these variables get memory
```

```
    # allocated on stack
```

```
    a = 20
```

```
    b = []
```

```
    c = ""
```

## Work of Heap Memory

The memory is allocated during the execution of instructions written by programmers. Note that the name heap has nothing to do with the heap data structure. It is called heap because it is a pile of memory space available to programmers to allocate and de-allocate. The variables are needed outside of method or function calls or are shared within multiple functions globally are stored in Heap memory.

### Example:

```
# This memory for 10 integers  
# is allocated on heap.  
a = [0]*10
```

## Built-in Data Types in Python

There are different types of data types in Python. Some built-in Python data types are –

- Numeric data types – int, float, complex
- String data types – str
- Sequence types – list, tuple, range
- Binary types – bytes, bytearray, memoryview
- Mapping data type – dict
- Boolean type – bool
- Set data types – set, frozenset

## Python Numeric Data types

In Python, the numeric data type is used to hold numeric values.

Integers, floating-point, and complex numbers fall under the Python numbers category. They are defined as int, float, and complex classes in Python.

- **int** – holds signed integers of non-limited length.
- **float** – holds floating decimal points, and it's accurate up to 15 decimal places.
- **complex** – holds complex numbers.

### **Python String Data type**

A string is a collection of Unicode symbols. The name for String in Python is str. Single or double quotations are used to represent strings. The use of triple quotes `"""` or `'''` to indicate multiple strings is acceptable. Between the quotations, every character is a part of the string.

The only restriction is the machine system's memory resources, which one may use as many characters as they like. In Python programming, deleting or updating a string will result in an error. As a result, the Python programming language does not permit the alteration of strings.

### **Python Sequence Data types**

- **List** – The list is a flexible data type only available in Python. It resembles the array in C/C++ in certain ways. However, the list in Python is noteworthy because it can store many sorts of data simultaneously. A list is an ordered collection of information expressed using commas and square brackets (`[]`). (`.`).
- **Tuple** – The list and a tuple are comparable in many respects. Tuples hold a collection of elements of various data kinds, much like lists do. The tuple's components are separated by commas (`,`) and parenthesized (`()`). Due to the inability to change the elements' size and value, tuples are read-only data structures.
- **Range** – The `range()` method in Python returns a list of integers that fall inside a specified range. It is most frequently used to iterate over a series of integers using Python loops.

### **Python Data Binary types**

- **bytes** – A bytes object results from the `bytes()` function. It can produce empty byte objects of the desired size or transform items into byte objects. `Bytes()` and `bytearray()`

return different types of objects: bytes() returns an immutable object, whereas bytearray() returns an alterable object.

- **bytearray** – The bytearray object, an array of the specified bytes, is returned by the bytearray() function. A modifiable series of numbers from 0 to x to 256 is provided.
- **memoryview** – Python programs may access an object's internal data that implements the buffer protocol using memoryview objects without copying. The byte-oriented data of an object may be read and written directly without copying it using the memoryview() method.

### Python Mapping Data type

- **dict** – A dictionary in Python is a collection of data items that are stored in an unordered fashion, much like a map. Dictionaries are made up of key-value pairs, as contrast to other data types, which can only contain a single value. Key-value pairs are included in the dictionary to increase its efficiency. A comma "separates each key," whereas each key-value pair in the representation of a dictionary data type is separated by a colon.

### Python Boolean Data type

- **bool** – True and False are the two pre-built values the boolean type offers. The provided statement's truth or falsity is determined using these values. It's identified by the bool class. Any non-zero integer or the letter "T" can be used to denote truth, while the number "0" or the letter "F" can denote falsehood.

### Python Set Data types

- **set** – The data type's unordered collection is called a Python Set. It has components that are unique, iterable, and changeable (may change after creation). The order of the items in a set is ambiguous; it can yield the element's modified sequence. Use the built-in method set() to build the set, or give a list of elements enclosed in curly braces and separated by commas. It may include several kinds of values.
- **frozenset** – The frozenset() method returns an immutable frozenset object whose initial elements are taken from the supplied iterable. A frozen set is an immutable version of a



Python set object. The elements of a set can be altered at any time, but once a frozen set has been created, its elements cannot be altered.