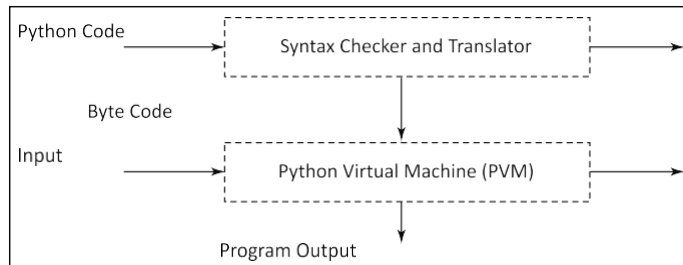


### Python Assignment No. 1

#### Q2.} Explain internal working of python with diagram.

**Ans.** When a programmer tries to run a Python code as a script or instructions in an interactive manner in a Python shell, then Python performs various operations internally.



The Python interpreter performs the following steps to execute a Python program or run a set of instructions in interactive mode.

**STEP 1:** The interpreter reads a Python code or instruction. Then it verifies that the instruction is well formatted, i.e. it checks the syntax of each line. If it encounters any error, it immediately halts the translation and shows an error message.

**STEP 2:** If there is no error, i.e. if the Python instruction or code is well formatted then the interpreter translates it into its equivalent form in low level language called “Byte Code”. Thus, after successful execution of Python script or code, it is completely translated into byte code.

**STEP 3:** Byte code is sent to the Python Virtual Machine (PVM). Here again the byte code is executed on PVM. If an error occurs during this execution then the execution is halted with an error message.

#### Q.3} Explain Python character set.

**Ans.** Any program written in Python contains words or statements which follow a sequence of characters. When these characters are submitted to the Python interpreter, they are interpreted or uniquely identified in various contexts, such as characters, identifiers, names or constants. Python uses the following character set:

Letters: Upper case and lower case letters

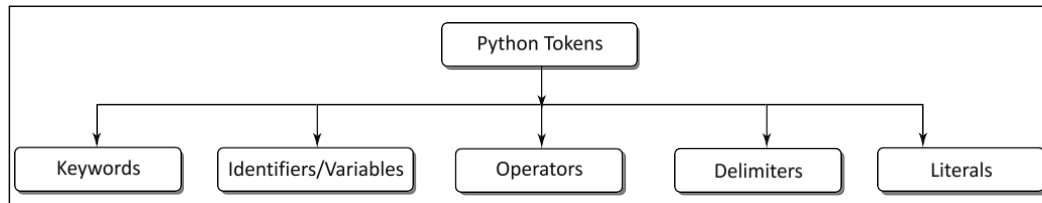
Digits: 0,1,2,3,4,5,6,7,8,9

Special Symbols: Underscore (`_`), (`()`), (`[]`), (`{}`), `+`, `-`, `*`, `&`, `^`, `%`, `$`, `#`, `!`, Single quote(`'`), Double quotes(`"`), Back slash(`\`), Colon(`:`), and Semi Colon (`;`)

White Spaces: (`'\t\n\r\b\c'`), Space, Tab.

### Q.4} Explain Python Tokens in details.

**Ans.** A program in Python contains a sequence of instructions. Python breaks each statement into a sequence of lexical components known as tokens. Each token corresponds to a substring of a statement. Python contains various types of tokens. Figure 2.1 shows the list of tokens supported by Python.



#### 1. Literal

Literals are numbers or strings or characters that appear directly in a program. A list of some literals in Python is as follows:

Example

78      #Integer Literal

21.98    #Floating Point Literal

‘Q’      #Character Literal

“Hello”      #String Literal

Python also contains other literals, such as lists, tuple and dictionary.

#### 2. Value and Type on Literals

Programming languages contain data in terms of input and output and any kind of data can be presented in terms of value. Here value can be of any form like literals containing numbers, characters and strings.

To know the exact type of any value, Python offers an in-built method called type.

The syntax to know the type of any value is `type (value)`

Example

```
>>> type('Hello World')
```

```
<class 'str'>
```

```
>>> type(123)
```

```
<class 'int'>
```

### 3. Keywords (any 5 to 6 keywords as example mhnun dee).

Keywords are reserved words with fixed meanings assigned to them. Keywords cannot be used as identifiers or variables.

and	del	from	None	True
as	elif	global	nonlocal	try
assert	else	if	not	while
break	except	import	or	with
class	False	in	pass	yield
continue	finally	is	raise	
def	for	lambda	return	

### 4. Operator

Python contains various operators, viz. arithmetic, relational, logical and bitwise operators.

<i>Operator Type</i>	<i>Operators</i>
+ - * / // % **	Arithmetic Operator
== != < > <= >=	Relational Operator
and not or	Logical Operator
&   ~ ^ << >>	Bitwise Operator

### 5. Delimiter

Delimiters are symbols that perform a special role in Python like grouping, punctuation and assignment. Python uses the following symbols and symbol combinations as delimiters.

() [] { }

, : . ' = ;

+= -= \*= /= //= %= &= |= ^= >>= <<= \*\*=

### 6. Identifier/Variable

Identifier is the name used to find a variable, function, class or other objects. All identifiers must obey the following rules.

An identifier:

- Is a sequence of characters that consists of letters, digits and underscore.
- Can be of any length
- Starts with a letter which can be either lower or upper case

- Can start with an underscore ‘\_’
- Cannot start with a digit
- Cannot be a keyword.

Some examples of valid identifiers are Name, Roll\_NO, A1, \_Address etc.

Python gives a syntax error if a programmer writes an invalid identifier. Some examples of invalid identifiers are First Name, 12Name, for, Salary@

### **Q.5} Define Print() function. Explain it with suitable example.**

**Ans.** In Python, a function is a group of statements that are put together to perform a specific task. The task of print function is to display the contents on the screen. The syntax of print function is:

Syntax of print() function:

```
print(argument)
```

The argument of the print function can be a value of any type int, str, float etc. It can also be a value stored in a variable.

```
>>> print('Hello Welcome to Python Programming')Hello Welcome to Python Programming
```

### **Q.6} Write down a simple python program with step wise instructions along with the algorithm to calculate area of circle.**

**Ans.**

**STEP 1:** Design an algorithm for the given problem.

An algorithm describes how a problem is to be solved by listing all the actions that need to be taken. An algorithm helps a programmer to plan for the program before actually writing it in a programming language. Algorithms are written in simple English language along with some programming code.

**STEP 2:** Translate an algorithm to programming instructions or code. Let us now write an algorithm to calculate the area of a rectangle.

- Algorithm to Calculate the Area of a Circle.
  - a) Get the radius of the circle from the user.
  - b) Use the relevant formula to calculate the area  
$$\text{Area} = \pi * \text{radius} * \text{radius}$$
  - c) Finally display the area of the circle.

```
radius = int(input('Please Enter the Radius of Circle: '))
print(' Radius = ', radius)           #Print Radius
PI = 3.1428                           #Initialize value of PI
Area = PI * radius * radius           #Calculate Area
print(' Area of Circle is: ',Area)    #Print Area
```

**Output**

```
Please Enter the Radius of Circle: 4
Radius = 4
Area of Circle is: 50.2848
```

**Q.7} Explain input() function in details.****Ans.**

The input() function is used to accept an input from a user. A programmer can ask a user to input a value by making use of input().

input() function is used to assign a value to a variable.

**Syntax:**

Variable\_Name = input()

OR

Variable\_Name = input('String')

```
Str1 = input('Enter String1:')
Str2 = input('Enter String2: ')
print(' String1 = ',Str1)
print(' String2 = ',Str2)
```

**Output**

```
Enter String1:Hello
Enter String2: Welcome to Python Programming
String1 = Hello
String2 = Welcome to Python Programming
```

The input() function is used to read the string from the user. The string values entered from the user are stored in two separate variables, viz. Str1 and Str2. Finally all the values are printed by making use of print() function.

### Q.8} How to translate mathematical formula into equivalent Python expression. Explain with example.

**Ans.**

Consider the following quadratic equation written in normal arithmetic manner.

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

The steps required to convert this quadratic equation into its equivalent Python expression are given as follows:

- **STEP 1:** The numerator and denominator are computed first to find the roots of the quadratic equation. Division between the numerator and denominator is performed as the last step. Hence, we can write the above expression as:

$$\text{Numerator/Denominator}$$

- **STEP 2:** The denominator is just 2a, so we can rewrite the formula as:

$$\text{Numerator}/((2 * a))$$

- **STEP 3:** Now we can split the numerator into two parts, i.e. left and right as follows:  $(\text{Left} + \text{Right}) / ((2 * a))$

- **STEP 4:** Substitute -b for left. There is no need to put parenthesis for -b because unary operator has higher precedence than binary addition. Hence, the above equation becomes:

$$(-b + \text{Right}) / ((2 * a))$$

- **STEP 5:** The right contains the expression inside the square root. Therefore, the above equation can be rewritten as:

$$(-b + \text{sqrt}(\text{expression})) / ((2 * a))$$

- **STEP 6:** But the expression inside the square root contains two parts left and right. Hence, the above equation is further rewritten as

$$(-b + \text{sqrt}(\text{left} - \text{right})) / ((2 * a))$$

- **STEP 7:** Now the left part contains the expression  $b^2$  and the right part contains the expression  $4*a*c$ . There is no need to put parenthesis for  $b^2$  because the exponent operator has higher precedence than the  $*$  operator since the expression  $4*a*c$  is present on the right side. The above equation can be rewritten as

$$(-b + \text{sqrt}(b^2 - 4*a*c)) / ((2 * a))$$

Thus, we have converted the mathematical expression into a Python expression. While converting an equation into a Python expression, one needs to only remember the rules of operator precedence and associativity.

```
P = 4
Q = 2
Z = (2 + 8 * P) / 2 - ((P-Q)*(P+Q))/2 + 4 * ((P+Q)/2)
print('( 2 + 8 * P) / 2 - ((P-Q)*(P+Q))/2 + 4 * ((P+Q)/2)')
print(' where P = ',P,' and Q = ', Q)
print(' Answer of above expression = ',Z)
```

#### Output

```
(2 + 8 * P) / 2 - ((P-Q)*(P+Q))/2 + 4 * ((P+Q)/2)
where P = 4 and Q = 2
Answer of above expression = 23.0
```