**J. BASKARAN M.Sc., B.Ed. (C.S)**
**jbaskaran89@gmail.com**
**Puducherry.**

**J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)**
**jilakkia@gmail.com**
**Puducherry.**

# 5. PYTHON - VARIABLES AND OPERATORS

## Section – A

**Choose the best answer** **(1 Mark)**

1. Who developed Python ?

A) Ritche **B) Guido Van Rossum**
C) Bill Gates D) Sunder Pitchai

2. The Python prompt indicates that Interpreter is ready to accept instruction.

**A) >>>** B) <<< C) # D) <<

3. Which of the following shortcut is used to create new Python Program ?

A) Ctrl + C B) Ctrl + F C) Ctrl + B **D) Ctrl + N**

4. Which of the following character is used to give comments in Python Program ?

**A)** # B) & C) @ D) $

5. This symbol is used to print more than one item on a single line.

A) Semicolon(;) B) Dollor($) **C) comma(,)** D) Colon(:)

6. Which of the following is not a token ?

**A) Interpreter** B) Identifiers C) Keyword D) Operators

7. Which of the following is not a Keyword in Python ?

A) break B) while C) continue **D) operators**

8. Which operator is also called as Comparative operator?

A) Arithmetic **B) Relational** C) Logical D) Assignment

9. Which of the following is not Logical operator?

A) and B) or C) not **D) Assignment**

10. Which operator is also called as Conditional operator?

**A) Ternary** B) Relational C) Logical D) Assignment

## Section-B

**Answer the following questions** **(2 Marks)**

**1. What are the different modes that can be used to test Python Program ?**

- In Python, programs can be written in two ways namely **Interactive mode** and **Script mode.**
- **Interactive mode** allows us to write codes in Python command prompt **( >>> ).**
- **Script mode** is used to create and edit python source file with the extension **.py**

**2. Write short notes on Tokens.**

- Python breaks each logical line into a sequence of elementary lexical components known as **Tokens**.
- The normal token types are ,

  1) Identifiers,

  2) Keywords,

  3) Operators,

  4) Delimiters and

  5) Literals.

**3. What are the different operators that can be used in Python ?**

- **Operators are special symbols** which represent computations, conditional matching in programming.
- Operators are categorized as Arithmetic, Relational, Logical, Assignment and Conditional.

**4. What is a literal? Explain the types of literals ?**

- Literal is a raw data given in a variable or constant.
- In Python, there are various types of literals. They are,

  1) **Numeric Literals** consists of digits and are immutable
  2) **String literal** is a sequence of characters surrounded by quotes.
  3) **Boolean literal** can have any of the two values: True or False.

**5. Write short notes on Exponent data?**

- An Exponent data contains decimal digit part, decimal point, exponent part followed by one or more digits.
- **Example:** 12.E04, 24.e04

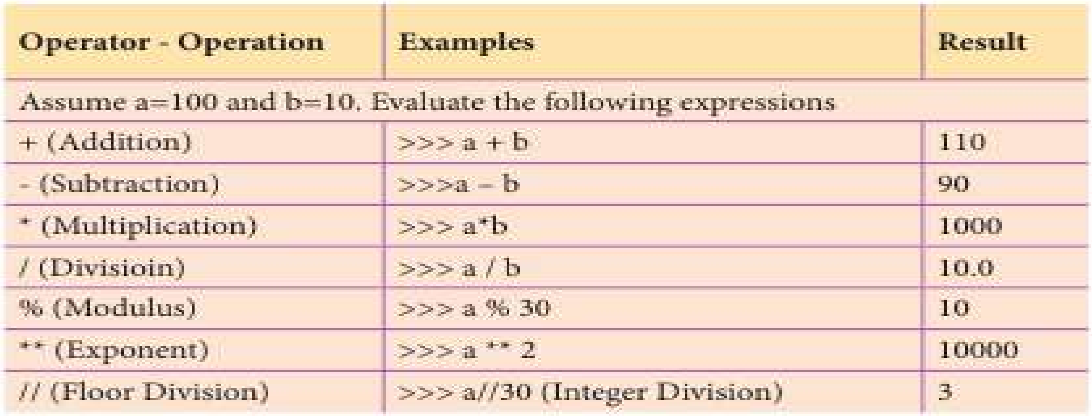
## Section-C

**Answer the following questions** **(3 Marks)**

**1. Write short notes on Arithmetic operator with examples.**

- An arithmetic operator is a mathematical operator used for simple arithmetic.

- It takes two operands and performs a calculation on them.
- **Arithmetic Operators used in python:**



| Operator - Operation | Examples | Result |
|---|---|---|
| Assume a=100 and b=10. Evaluate the following expressions | | |
| + (Addition) | >>> a + b | 110 |
| - (Subtraction) | >>>a – b | 90 |
| * (Multiplication) | >>> a*b | 1000 |
| / (Divisioin) | >>> a / b | 10.0 |
| % (Modulus) | >>> a % 30 | 10 |
| ** (Exponent) | >>> a ** 2 | 10000 |
| // (Floor Division) | >>> a//30 (Integer Division) | 3 |

**2. What are the assignment operators that can be used in Python?**

- '=' is a simple **assignment operator** to assign values to variable.
- There are various **compound operators** in Python like **+=, -=, *=, /=, %=, **= and //=.**
- **Example:**

**a=5** # assigns the value 5 to a

**a,b=5,10** # assigns the value 5 to a and 10 to b

**a+=2** # **a=a+2**, add 2 to the value of 'a' and stores the result in 'a' (Left hand operator)

**3. Explain Ternary operator with examples.**

- Ternary operator is also known as **conditional operator** that evaluates something based on a condition being true or false.
- It simply allows testing a condition in a single line replacing the multiline if-else making the code compact.

**Syntax:**

***Variable Name = [on_true] if [Test expression] else [on_false]***

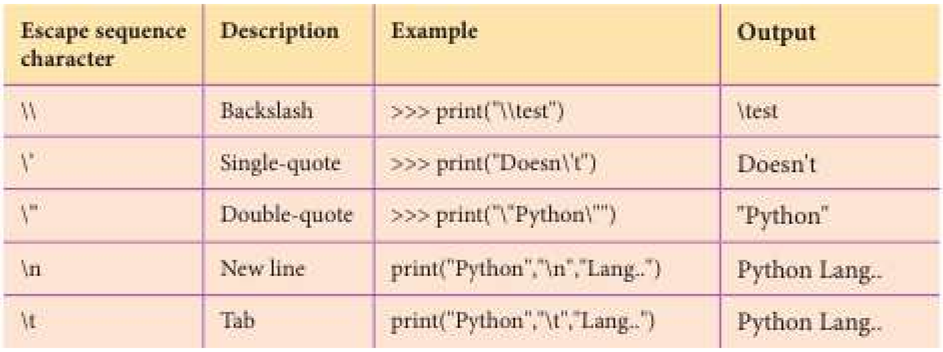
**Example :**

min = 50 if 49<50 else 70 # Output**: min = 50**

**4. Write short notes on Escape sequences with examples.**

- In Python strings, the backslash "\" is a special character, also called the "**escape**" character.

- It is used in representing certain whitespace characters.
- Python supports the following escape sequence characters.



| Escape sequence character | Description | Example | Output |
|---|---|---|---|
| \\ | Backslash | >>> print("\\test") | \test |
| \' | Single-quote | >>> print("Doesn\'t") | Doesn't |
| \" | Double-quote | >>> print("\"Python\"") | "Python" |
| \n | New line | print("Python","\n","Lang..") | Python Lang.. |
| \t | Tab | print("Python","\t","Lang..") | Python Lang.. |

**5. What are string literals? Explain.**

- In Python a string literal is a **sequence of characters** surrounded by **quotes**.
- Python supports **single, double and triple quotes** for a string.
- A character literal is a **single character** surrounded by **single or double quotes**.
- The value with **triple-quote "' '"** is used to give **multi-line** string literal.
- **Example:**

```
strings = "This is Python"
char = "C"
multiline_str = ''' This is a multiline string with more than one line code.'''
print (strings)
print (char)
print (multiline_str)
```

- **Output:**

```
This is Python
C
This is a multiline string with more than one line code.
```

## Section - D

**Answer the following questions:** **(5 Marks)**

**1. Describe in detail the procedure Script mode programming.**

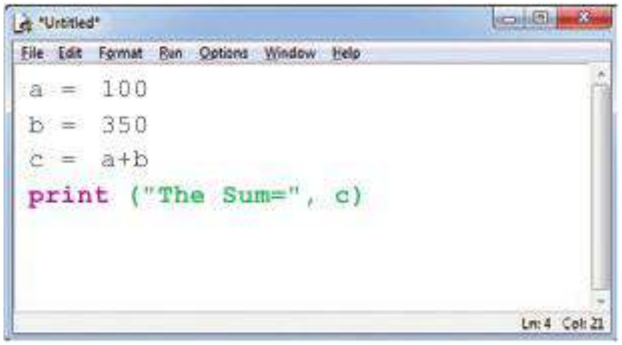
**SCRIPT MODE PROGRAMMING:**

- A script is a text file containing the Python statements.

- Once the Python Scripts is created, they are reusable , it can be executed again and again without retyping.
- The Scripts are editable.

**<u>(i) Creating Scripts in Python</u>**

1. Choose **File → New File** or press **Ctrl + N** in Python shell window.
2. An **untitled** blank script text editor will be displayed on screen.
3. Type the code in Script editor as given below,



```
a = 100
b = 350
c = a+b
print ("The Sum=", c)
```

**<u>(ii) Saving Python Script</u>**

(1) Choose **File → Save** or Press **Ctrl + S**

(2) Now, **Save As** dialog box appears on the screen.

(3) In the **Save As** dialog box

- Select the location to save your Python code.
- Type the file name in **File Name** box.
- Python files are by default saved with extension **.py.**
- So, while creating scripts using Python Script editor, no need to specify the file extension.

(4) Finally, click **Save** button to save your Python script.

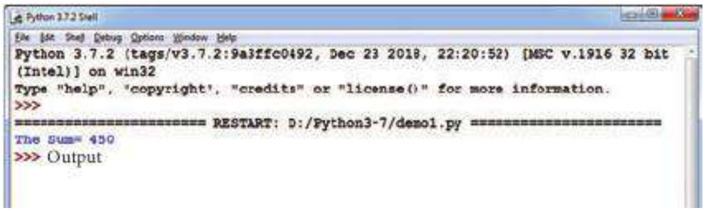
**<u>(iii) Executing Python Script</u>**

(1) Choose **Run → Run Module** or Press **F5**

(2) If your code has any error, it will be shown in red color in the IDLE window, and Python describes the type of error occurred.

- To correct the errors, go back to Script editor, make corrections, save the file and execute it again.

(3) For all error free code, the output will appear in the IDLE window of Python as shown in **Figure.**



```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
====================== RESTART: D:/Python3-7/demo1.py ======================
The Sum= 450
>>> Output
```

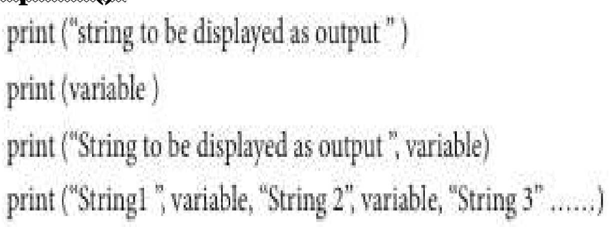
**2. Explain input() and print() functions with examples.**

**Input and Output Functions**

- A program needs to interact with the user to accomplish the desired task; this can be achieved using **Input-Output functions**.
- The **input()** function helps to enter data at run time by the user
- The output function **print()** is used to display the result of the program on the screen after execution.

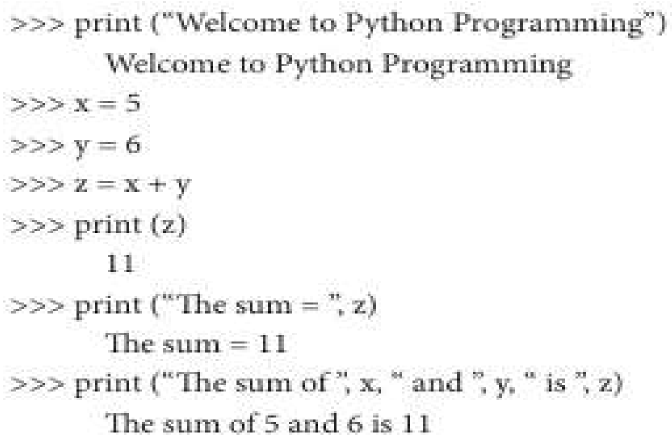
**1) print() function**

- In Python, the **print()** function is used to display result on the screen.
- **Syntax for print():**



```
print ("string to be displayed as output " )
print (variable )
print ("String to be displayed as output ", variable)
print ("String1 ", variable, "String 2", variable, "String 3" ......)
```

- **Example:**

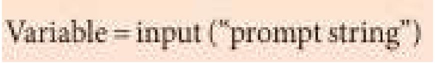


```
>>> print ("Welcome to Python Programming")
          Welcome to Python Programming
>>> x = 5
>>> y = 6
>>> z = x + y
>>> print (z)
          11
>>> print ("The sum = ", z)
          The sum = 11
>>> print ("The sum of ", x, " and ", y, " is ", z)
          The sum of 5 and 6 is 11
```

- The **print ( )** evaluates the expression before printing it on the monitor.
- The print () displays an entire statement which is specified within print ( ).
- **Comma ( , )** is used as a separator in **print ( )** to print more than one item.

**2) input() function**

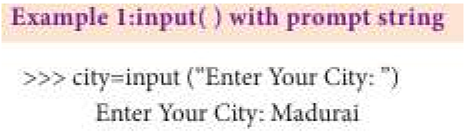
- In Python, **input( )** function is used to accept data as input at run time.
- The syntax for **input()** function is,



```
Variable = input ("prompt string")
```

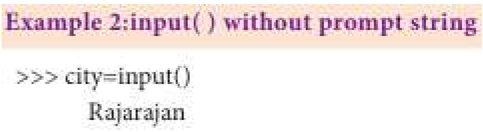
- **"Prompt string"** in the syntax is a message to the user, to know what input can be given.

- If a prompt string is used, it is displayed on the monitor; the user can provide expected data from the input device.
- The **input( )** takes typed data from the keyboard and stores in the given variable.
- If prompt string is not given in **input( )**, the user will not know what is to be typed as input.
- **Example:**



Example 1:input( ) with prompt string

```
>>> city=input ("Enter Your City: ")
         Enter Your City: Madurai
```



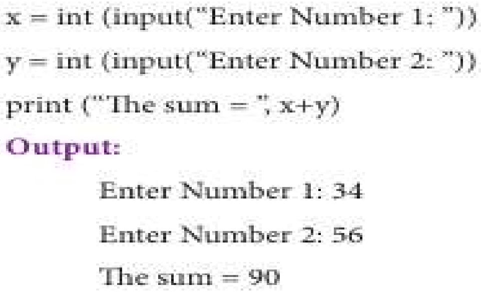
Example 2:input( ) without prompt string

```
>>> city=input()
         Rajarajan
```

- In **Example 1** input() using prompt string takes proper input and produce relevant output.
- In **Example 2** input() without using prompt string takes irrelevant input and produce unexpected output.
- So, to make your program more interactive, provide prompt string with **input( ).**

**Input() using Numerical values:**

- The **input ( )** accepts all data as string or characters but not as numbers.
- The **int( )** function is used to convert string data as integer data explicitly.
- **Example:**



```
x = int (input("Enter Number 1: "))
y = int (input("Enter Number 2: "))
print ("The sum = ", x+y)
Output:
            Enter Number 1: 34
            Enter Number 2: 56
            The sum = 90
```

**3. Discuss in detail about Tokens in Python.**

**Tokens**

- Python breaks each logical line into a sequence of elementary lexical components known as **Tokens**.
- The normal token types are,

1) Identifiers,
2) Keywords,
3) Operators,
4) Delimiters and
5) Literals.

- Whitespace separation is necessary between tokens, identifiers or keywords.

**<u>1) Identifiers</u>**

- An Identifier is a name used to identify a variable, function, class, module or object.
- An identifier must start with an alphabet (A..Z or a..z) or underscore ( _ ).
- Identifiers may contain digits (0 .. 9)
- Python identifiers are case sensitive i.e. uppercase and lowercase letters are distinct.
- Identifiers must not be a **python** keyword.
- Python does not allow punctuation character such as %,$, @ etc., within identifiers.
- **Example of valid identifiers:** Sum, total_marks, regno, num1
- **Example of invalid identifiers:** 12Name, name$, total-mark, continue

**<u>2) Keywords</u>**

- Keywords are special words used by Python interpreter to recognize the structure of program.
- Keywords have **specific meaning for interpreter**, they cannot be used for any other purpose.
- **Python Keywords:** false, class, If, elif, else, pass, break etc.

**<u>3) Operators</u>**

- **Operators are special symbols** which represent computations, conditional matching in programming.
- Operators are categorized as Arithmetic, Relational, Logical, Assignment and Conditional.
- Value and variables when used with operator are known as **operands**.
- **<u>Example:</u>**

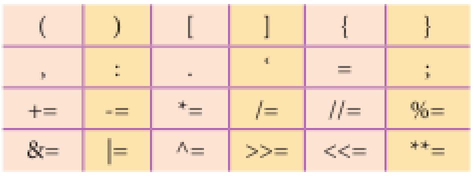
```
a=100
b=10
print ("The Sum = ",a+b)
print ("The a > b = ",a>b)
print ("The a > b or a == b = ",a>b or a==b)
a+=10
print(“The a+=10 is =”, a)
```

- **<u>Output:</u>**

```
The Sum = 110
The a>b = True
The a > b or a == b = True
The a+=10 is= 110
```

**<u>4) Delimiters</u>**

- Python uses the symbols and symbol combinations as delimiters in expressions, lists, dictionaries and strings.
- Following are the delimiters.



| | | | | | |
|---|---|---|---|---|---|
| ( | ) | [ | ] | { | } |
| , | : | . | ‘ | = | ; |
| += | -= | *= | /= | //= | %= |
| &= | \|= | ^= | >>= | <<= | **= |

## 5) Literals

- Literal is a raw data given in a variable or constant.
- In Python, there are various types of literals. They are,

  1) **Numeric Literals** consists of digits and are immutable

  2) **String literal** is a sequence of characters surrounded by quotes.

  3) **Boolean literal** can have any of the two values: True or False.

**PREPARED BY**

**J. BASKARAN M.Sc., B.Ed. (C.S)**
**jbaskaran89@gmail.com**

**J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)**
**jilakkia@gmail.com**