

# QB365 Question Bank Software Study Materials

## Python - Variables and Operators Important 2, 3 & 5 Marks Questions With Answers (Book Back and Creative)

12th Standard

Computer Science

Total Marks : 75

### 2 Marks

10 x 2 = 20

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

**Answer :** (i) In Python, programs can be written in two namely Interactive mode and script mode.

(ii) **Interactive mode** allows us to write codes in Python command prompt ( > > > ).

(iii) **Script mode** is used to create and edit python source file.

2) Write short notes on Tokens.

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

(i) Identifiers,

(ii) Keywords,

(iii) Operators,

(iv) Delimiters and

(v) Literals.

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

**Answer :** The different operators that can be used in python are

(i) Arithmetic operators

(ii) Relational or Comparative operator

(iii) Logical operators

(iv) Assignment operators

(v) Conditional operator

4) What is a literal? Explain the types of literals?

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

#### **1) Numeric Literals:**

Numeric Literals consists of digits and are immutable.

#### **2) String Literals:**

In Python a string literal is a sequence of characters surrounded by quotes.

#### **3) Boolean Literals:**

A Boolean literal can have any of the two values: True or False.

5) Write short notes on Exponent data?

**Answer :** An Exponent data contains decimal digit part, decimal point, exponent part followed by one or more digits.

6) How the interactive mode of Python shell can be used as simple calculator?

**Answer :** In interactive mode Python code can be directly typed and the interpreter displays the result (s) immediately. The interactive mode can also be used as a simple calculator.

7) Why the following identifiers are invalid?

(i) 12 Name

(ii) name\$

(iii) physics-mark

(iv) break

**Answer :** (i) An identifies must start with an alphabet.

- (ii) No punctuation characters are allowed.
- (iii) Only underscore (-) allowed.
- (iv) Identifiers must not be a Python keyword.

8) Write a note on relational or comparative operator.

**Answer :** A Relational operator is also called as Comparative operator which checks the relationship between two operands. If the relation is true, it returns True otherwise it returns False.

9) What is input () function?

**Answer :** The input () function helps to enter data at run time by the user.

10) What is numeric Literals?

**Answer :** Numeric literals consist of digits and are immutable. Numeric literals can belong to 3 different numerical types Integer, Float, and complex.

**3 Marks**

10 x 3 = 30

11) Write short notes on Arithmetic operator with examples.

**Answer :** (i) An arithmetic operator is a mathematical operator that takes two operands and performs a calculation on them. They are used for simple arithmetic.

(ii) Most computer languages contain a set of such operators that can be used within equations to perform different types of sequential calculations.

(iii) Python supports the following Arithmetic operators.

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
/ (Division)	> > > a/b	10.0
% (Modulus)	> > > a%30	10
** (Exponent)	> > > a**2	10000
// (Floor Division)	> > > a// 30 (Integer Division)	3

12) What are the assignment operators that can be used in Python?

**Answer :** (i) In Python, is a simple assignment operator to assign values to variable. Let = 5 and = 10 assigns the value 5 to and 10 these two assignment statement can also be given a =5 that assigns the value 5 and 10 on the right to the variables a and b respectively.

(ii) There are various compound operators in Python like +=, -=, \*=, /=, %=, \*\*= and //= are also available.

Operator	Description	Example
Assume x = 10		
=	Assigns right side operands to left variable.	> > > X=10 > > > b="computer"
+=	Added and, assign back the result to left operand i.e, x = 30	> > > x + = 20# x = x + 20
-=	Subtracted and assign back the result to left operand i.e, x = 25	> > > x- = 5# x = x - 5
*=	Multiplied and assign back the result to left operand ie, x = 125	> > > x* = 5# x = x*5
/=	Divided and assign back the result to left operand i.e, x = 62.5	> > > x/ = 2# x = x/2
%=	Taken modulus (Remainder) using two operands and assign the result to left assign the result to left	> > > x% = 3 # x = x%3
**=	Performed exponential (power) calculation on operators and assign value to the left operand i.e, x = 6.25	> > > x**=2 #x = x**2
//=	Performed floor division on Operators and assign value to the left operand i.e, x= 2.0	> > > x// = 3

13) Explain Ternary operator with examples.

**Answer :** (i) Ternary operator is also known as conditional operator that evaluate something based on a condition being true or false.

(ii) It simply allows testing a condition in a single line replacing the multiline if-else making the code compact. The syntax for conditional operator is,

Variable Name = [on\_true] if [Test expression]  
else [on\_false]

(iii) Example:

min = 50 if 49 < 50 else 70 # min = 50

min = 50 if 49 > 50 else 70 # min = 70

14) Write short notes on Escape sequences with examples.

**Answer :** (i) In Python strings, the backslash "\" is a special character, also called the "escape" character.

(ii) It is used in representing certain whitespace characters: "\t" is a tab, "\n" is a newline, and "\r" is a carriage return.

(iii) For example to print the message "It's raining", the Python command is

```
> > > print ("It \ 's raining")
```

It's raining

Python supports the following escape sequence characters.

Escape sequence characters	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..

15) What are string literals? Explain.

**Answer :** (i) In Python, a string literal is a sequence of characters surrounded by quotes. Python supports Single, double and triple quotes for a string.

(ii) A character literal is a single character surrounded by single or double quotes. The value with triple-quote u ''' ''' is used to give multi-line string literal.

**Program:**

To test String Literals:

```
# Demo Program to test String Literals
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)
# End of the Program
```

**Output:**

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

16) Differentiate two ways in which Python programs can be written.

**Answer :** (i) In Python, programs can be written in two ways namely Interactive mode and Script mode.

(ii) The Interactive mode allows us to write codes in Python command prompt > > > whereas in script mode programs can be written and stored as separate file with the extension.

(iii) .py and executed. Script mode is used to create and edit python source file.

17) Fill up the blanks to get the following output from Python code given.

**Output:**

```
Enter Number 1: 34
Enter Number 2: 70
The Sum = 104
```

**Code:**

```
X = (1) (input ("Enter Number 1:"))
Y = (2) ( (3) ("Enter Number 2 :"))
(4) ("The sum =", (5) )
```

**Answer :** (1) int

- (2) int
- (3) input
- (4) print
- (5) x + y

18) Write a note on statement which are ignored by the Python interpreter.

**Answer :** (i) In Python, comments begin with hash symbol (#). The lines that begins with # are considered as comments and ignored by the Python interpreter.

(ii) Comments may be single line or no multilines. The multiline comments should be enclosed within a set of # as given below.

# It is Single line Comment

# It is multiline comment

which contains more than one line #

19) What is numeric Literals? What are they?

**Answer :** Numeric Literals consists of digits and are immutable. Numeric Literals can belong to 3 different numerical types Integer, Float and complex.

a = 0b1010 #Binary literals

b = 100 #Decimal literal

c = 0o310 #Octal literal

d = 0X12C #Hexa decimal literal.

20) Write a program to demonstrate numeric literals.

**Answer : #Program to demonstrate Numeric literals.**

a = 0b1010 #Binary literals

b = 100 #Decimal literal

c = 0o310 #Octal literal

d = 0X12c #Hexadecimal literal

print ("Integer literals:", a,b,c,d)

#float literals

float\_1 = 10.5

float\_2 = 1.5e2

print("Float literals:", float\_1,float\_2)

#Complex literal

x = 1 + 3.14j

Print("complex literals:",x)

print("x = ",x, Imaginary part of x =" x.imag, "Real part of x = ", x.real"

#End of the program

**Output:**

Integer literals: 10 100 200 300

Float literals: 10.5 150.0

Complex literals:

x=(1.3.14) Imaginary part of x = 3.14

Real part of x = 1.0

**5 Marks**

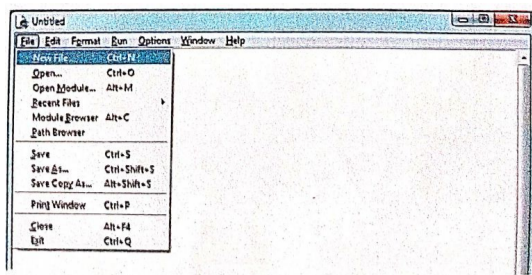
5 x 5 = 25

21) Describe in detail the procedure Script mode programming.

**Answer :** A script is a text file containing the Python statements. Python Scripts are reusable code. Once the script is created, it can be executed again and again without retyping. The Scripts are editable

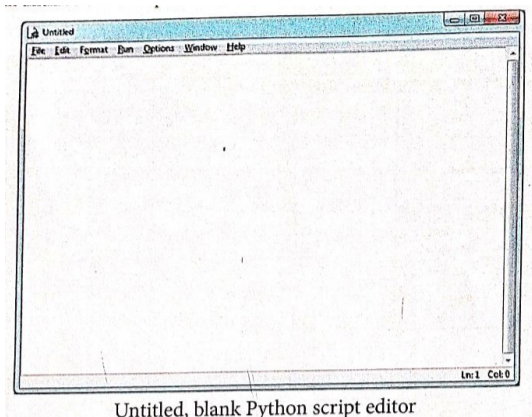
### i) Creating Scripts in Python:

1. Choose File → New File or press Ctrl + N in Python shell window.



To create new File

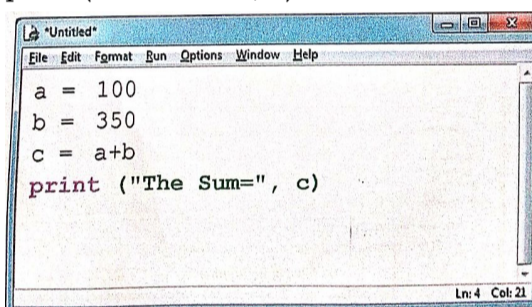
2. An untitled blank script text editor will be displayed on screen



Untitled, blank Python script editor

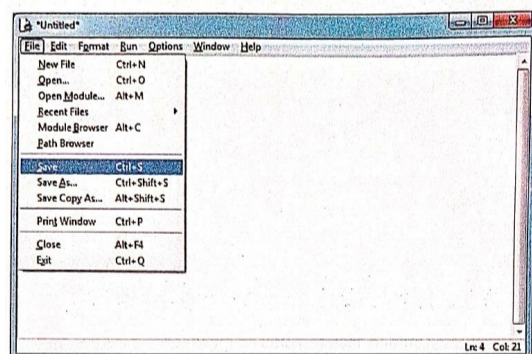
3. Type the following code in script editor.

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

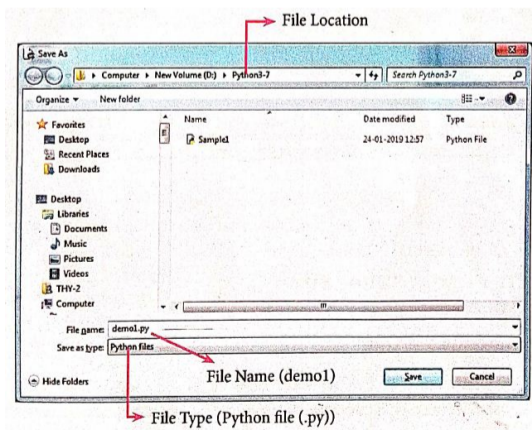


### ii) Saving Python script:

1. Choose File → Save or press Ctrl+S.



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

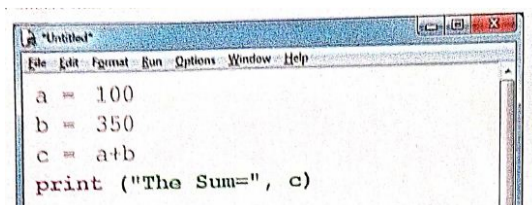


3. In the Save As dialog box, select the location where you want to save your Python code, and type the File name in File Name box. Python files are by default saved with extension .py. Thus, while creating Python Scripts using Python's script editor, no need to specify the file extension.

4. Finally click save button to save your python script.

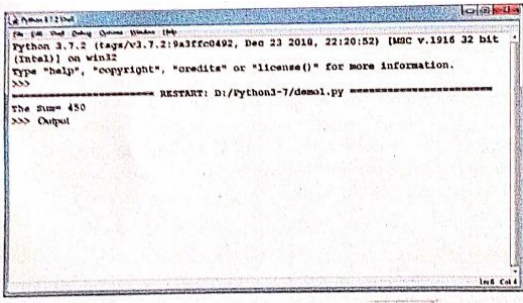
### iii) Executing Python Script:

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 using Ctrl +s or File → Save and execute it again.
3. For all error line code, the output will appear in the IDLL window of Python.



22) Explain input() and print() functions with examples.

**Answer : 1. The print () Function:**

In Python, the print() function is used to display result on the screen. The syntax for print ( ) is as follows:

**Example:**

```
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 Programoming
> > > x = 5
> > > y = 6
> > > z= X+Y
> > > print (z)
> > > 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.

**input () function:**

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

**Syntax:**

```
Variable = input ("prompt String')
```

Where, prompt string in the Syntax is a statement or 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 whatever is typed from the Keyboard and stores the entered data in the given Variable. If prompt string is not given in input() no message is displayed on the screen, then, the user will not know what is to be typed as input.

**Example 1:**

```
input () with prompt String
> > > City=input ("Enter your city:")
Enter your city: Madurai
> > > print ("I am from", city)
I am from Madurai
```

**Example 2:**

```
input() without prompt string
> > > city = input ()
Rajarajan
> > > print ("I am from", city)
I am from Rajarajan
```

in Example 2, the input () is not having any prompt string, thus the user will not know what is to be typed as input. If the user inputs irrelevant data as given in the above example then the output will be unexpected. So, to make your program more interactive, provide prompt string with input (). The input () accepts all data as string or characters but not as numbers. If a numerical value is entered, the input values should be explicitly converted into numeric data type. The int() function is used to convert string data as integer data explicitly.

**Example 3:**

```
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.
```



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

- (i) Identifiers,
- (ii) Keywords,
- (iii) Operators,
- (iv) Delimiters and
- (v) Literals.

**(i) Identifiers:**

1. An Identifier is a name used to identify a variable, function, class, module or object.
2. An identifier must start with an alphabet (A..Z or a..z) or underscore (-).
3. Identifiers may contain digits (0 .. 9).
4. Python identifiers are case sensitive i.e. uppercase and lowercase letters are distinct.
5. Identifiers must not be a python keyword.
6. Python does not allow punctuation character such as %, \$, @ etc., within identifiers.

**Example of valid identifiers.**

Sum, total\_marks, regno, num1.

**Example of invalid identifiers**

12 Name, name \$, total-mark, continue.

**Keywords:**

Keywords are special words used by Python interpreter to recognize the structure of program. As these words have specific meaning for interpreter, they cannot be used for any other purpose.

false	Class	finally	is	return
none	continue	for	lambda	try
true	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	expect	in	raise	

**Operators:**

In computer programming languages operators are special symbols which represent computations, conditional matching etc. The value of an operator used is called operands. Operators are categorized as Arithmetic, Relational, Logical, Assignment etc. Value and variables when used with operator are known as operands.

**i) Arithmetic operators**

An arithmetic operator is a mathematical operator that takes two operands and performs a calculation on them. They are used for simple arithmetic. Most computer languages contain a set of such operators that can be used within equations to perform different types of sequential calculations.

Python supports the following Arithmetic operators.

Operator-operation	Examples	Result
Assume a=100 and b=10. Evaluate the following expression		
+ (addition)	>>>a+b	110
- subtraction)	>>>a-b	90
* (multiplication)	>>>a*b	1000
/ (Division)	>>>a/b	10.0
% (Modulus)	>>>a%30	10
** (Exponent)	>>>a**2	10000
// (Floor Division)	>>>a//30 (Integer Division)	3

**ii) Relational or Comparative Operators**

A Relational operator is also called as comparative operator which checks the relationship between two operands. If the relation is true, it returns True; otherwise it returns False.

Python supports following relational operators.

Operator- operation	Examples	Result
Assume the value of a =100 and b=35, Evaluate the following expressions.		
== (is Equal)	>>>a==b	False
>(Greater than)	>>> a>b	True
<(Less than)	>>>a<b	False
>= (Greater than or Equal to)	>>>a>=b	True
<= (Less than or Equal to)	>>>a<=b	False
!= (Not equal to)	>>>a!=b	True

**ii) Logical Operators**

In Python, Logical operators are used to perform logical operations on the given relational expressions. There are three logical operators they are and, or and not.

Operator- operation	Examples	Result
Assume a =97 and b=35, Evaluate the following Logical expressions.		
or	>>> a>b or a==b	True
and	>>> a>b and a==b	False
not	>>> not a>b	False i.e, Not True.

#### iv) Assignment Operators

In Python, = is a simple assignment operator to assign values to variables. Let a=5 and b=10 assigns the value 5 to a and 10 to b. These two assignment statements can also be given as a,b=5,10 that assigns the value 5 and 10 on the right to the variables a and b respectively.

There are various compound operators in Python. Like +=, -=, \*=, /=, %=, \*\*= and //= are also available.

Operator	Description	Example
=	Assumes right side operands to left variable.	> > > X=10 > > > b="computer"
+=	Added and, assign back the result to left operand i.e, x = 30	> > > x + = 20# x = x + 20
-=	Subtracted and assign back the result to left operand i.e, x = 25	> > > x - = 5# x = x - 5
*=	Multiplied and assign back the result to left operand i.e, x = 125	> > > x * = 5# x = x*5
/=	Divided and assign back the result to left operand i.e, x = 62.5	> > > x / = 2# x = x/2
%=	Taken modulus (Remainder) using two operands and assign the result to left assign the result to left	> > > x % = 3 # x = x%3
**=	Performed exponential (power) calculation on operators and assign value to the left operand i.e, x = 6.25	> > > x ** = 2 #x = x**2
//=	Performed floor division on Operators and assign value to the left operand i.e, x= 2.0	> > > x // = 3

#### v) Conditional Operator

Ternary operator is also known as Conditional operator that evaluate 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. The syntax conditional operator is

Variable Name = [on\_true] if [Test expression] else [on\_false]

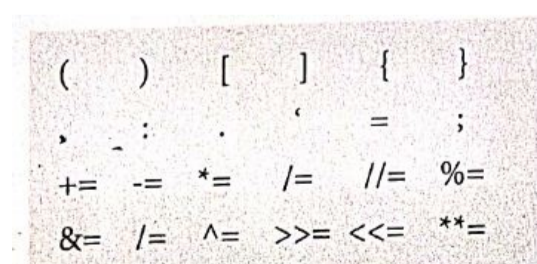
##### Example:

min = 50 if 49 < 50 else 70 # min = 50

min = 50 if 49 > 50 else 70 # min = 70

##### Delimiters:

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



#### Literals

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

- 1) Numeric
- 2) String
- 3) Boolean

##### 1) Numeric Literals

Numeric literals consist of digits and are immutable (unchangeable). Numeric literals can belong to 3 different numerical types

Integer, float and complex.

## 2) String literals

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.

## 3) Boolean literals

A Boolean literal can have any of the two values: True or False.

## 4) Escape sequences

In python strings, the backslash "\" is a special character, also called the "escape" character. It is used in representing certain white space character: "\t" is a tab, "\n" is a newline, and "\r" is a carriage return. For Example to print the message "It's raining"

The python command is

```
> > print ("It \'s raining")
```

It's raining

Python supports the following escape sequence characters.

Escape sequence characters	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

24) Write the output for the following python code..

```
x=10
x+=20
print ("The x + = 20 is =",x)
x-=5
print ("The x -= 5 is =",x)
x*=5
print ("The x *= 5 is =",x)
x/=2
print ("The x / = 2 is =",x)
x%=3
print ("The x %= 3 is =",x)
x**=2
print ("The x **= 2 is =",X)
x//=3
print ("The x /= 3 is =",x)
```

**Answer :** Type a Value for X : 10

The x + = 20 is = 30

The x - = 5 is = 25

The x \* = 5 is = 125

The x / = 2 is = 62.5

The x % = 3 is = 2.5

The x \*\* = 2 is = 6.25

The x // = 3 is = 2.0

25) Explain briefly about Python data types with an example.

**Answer :** All data values in Python are objects and each object or value has a type. Python has Built-in or fundamental data types such as Number, String, Boolean, tuples, lists, and dictionaries.

### 1. Number data types

The built-in number objects in Python supports integers, floating-point numbers and complex numbers.

Integer Data can be decimal, octal or hexadecimal. Octal integer use O (both upper and lower case) to denote octal digits and hexadecimal integer use Ox (both upper and lower case) and L (only upper case) to denote long integer.

#### Example

```
702,4567,567          #Decimal integers
O102, 0876, O432      #Octal integers
Ox102, Ox876, Ox432   #Hexadecimal integers
34L, 523 L           #Longdecimal integer
```

A floating-point data is represented by a sequence of decimal digits that includes a decimal point. An Exponent data contains a decimal point, exponent part followed by one or more digits.

#### Example

```
123.34,456.23,156.223 #Floating point data
12. E04, 21.e04       #Exponent data
```

A complex number is made up of two floating-point values, one each for the real and imaginary parts.

### 2. Boolean Data type

A Boolean data can have any of the two values: True or False.

#### Example

```
Bool_var 1 = True
Bool_var 2 = False
```

### 3. String Data type

String data can be enclosed with a single quote or double quote or triple quote.

#### Example

```
char-data = 'A'
srting_data = " Computer Science"
Multi_data = """ String data can be enclosed with a single quote or double quote or triple quote."""
```