

**Course  
&  
Test Series**

**Numbers and Strings in Python**

 **CBSE**

 **ICSE**

 **NTSE**

 **Banking & Insurance**

 **Central Govt. Service**

 **State Govt. Services**

 **LAW Entrance**

 **MBA Entrance**

 **Railways & Metro Services**

**...many more**

**abhyasonline.in**

**Numbers in Python**

**Numbers in Python**

Numbers in Python are used to store numeric values.

Python supports three main numeric data types:

- int - integers (whole numbers)
- float - numbers with decimal points
- complex - numbers with real and imaginary parts

**Basic Arithmetic:**

Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication, and division.

Four basic arithmetic operators exist in Python:

1. Addition (+) 2. Subtraction (-) 3. Multiplication (\*) 4. Division (/)

Applying arithmetic operators:

Assume that  $x = 7$ ,  $y = 20$ , and  $z = 2$ .

Given the following lines of code, what is the output of the code?

**Ques 1:**

`c = 0`

`c = x - z`

`c = c + 1`

`print(c)`

Options: a. 1 , b. 5 , c. 6

**Ques 2:** What is the value of a?

`a = 3.5 - 1.5`

Options: a. 2 , b. 2.0 , c. 2.5

**Ques 3:** What is the output of `print(x / z)`?

Options: a. 3 , b. 3.0 , c. 3.5

**Operator Precedence**

When a calculation has multiple operators, each operator is evaluated in order of precedence. Ex: 1

Operator	Description	Example	Result
()	Parentheses	<code>(1 + 2) * 3</code>	9
**	Exponentiation	<code>2 ** 4</code>	16
+, -	Positive, negative	<code>-math.pi</code>	-3.14159
*, /	Multiplication, division	<code>2 * 3</code>	6
+, -	Addition, subtraction	<code>1 + 2</code>	3

**Operator Precedence from Higher to Lower**

Course  
&  
Test Series

Numbers and Strings in Python

+ 2 \* 3 is 7 because multiplication takes precedence over addition. However, (1 + 2) \* 3 is 9 because parentheses take precedence over multiplication.

Ques 1: What is the value of  $4 * 3 ** 2 + 1$ ?  
Options: a. 37 , b. 40 , c. 145

Ques 2: Which part of  $(1 + 3) ** 2 / 4$  evaluates first?  
Options: a.  $4 ** 2$  , b.  $1 + 3$  , c.  $2 / 4$

Solved Example: Arithmetic Operations on Numbers

a = 10  
b = 3

```
print("Addition:", a + b)
print("Subtraction:", a - b)
print("Multiplication:", a * b)
print("Division:", a / b)
print("Floor Division:", a // b)
print("Modulus:", a % b)
print("Power:", a ** b)
```

Output:

```
Addition: 13
Subtraction: 7
Multiplication: 30
Division: 3.3333333333333335
Floor Division: 3
Modulus: 1
Power: 1000
```

Explanation:

- Python supports all basic arithmetic operators.
- / gives float result, // gives integer quotient, % gives remainder.

Assignment on Numbers in Python

Ques 1: Write a Python program to add two numbers: 12 and 8. Print the result.  
Ques 2: Write a Python program to check if the number 25 is even or odd.  
(Hint: Use modulus % operator.)

...many more  
abhyasonline.in

**Course  
&  
Test Series**



...many more

**abhyasonline.in**

**Numbers and Strings in Python**

Ques 3: Write a Python program to calculate the area of a circle with radius 10.  
(Hint: Formula =  $3.14 \times \text{radius}^2$ )

Strings in Python

String: A string is a sequence of characters enclosed in single quotes ' ', double quotes " ", or triple quotes ''' ''' / """" """".

Quote marks:

A string is a sequence of characters enclosed by matching single (') or double (") quotes.

Ex: "Happy birthday!" and '21' are both strings.

To include a single quote (') in a string, enclose the string with matching double quotes ("). Ex: "Won't this work?"

To include a double quote ("), enclose the string with matching single quotes ('). Ex: 'They said "Try it!", so I did

Valid string	Invalid string
"17" or '17'	17
"seventeen" or 'seventeen'	seventeen
"Where?" or 'Where?'	"Where?'
"I hope you aren't sad."	'I hope you aren't sad.'
'The teacher said "Correct!" '	"The teacher said "Correct!" "

Rules for Strings



Course  
&  
Test Series

Numbers and Strings in Python

Solved Example: Example of Strings

```
name = "Aarav"
greeting = 'Hello, World!'
paragraph = """This is
a multi-line
string."""
```

```
print(name)
print(greeting)
print(paragraph)
print(type(name))
```

Output:

```
Aarav
Hello, World!
This is
a multi-line
string.
<class 'str'>
```

Explanation:

- Strings are immutable (cannot be changed after creation).
- Triple quotes allow multi-line text.

String Concatenation (Joining Strings)

We can join two or more strings using the + operator.

```
str1 = "Hello"
str2 = "World"
result = str1 + " " + str2
print(result)
```

Output:

```
Hello World
```

String Repetition

We can repeat strings using the \* operator.

```
word = "Hi "
print(word * 3)
```

Output:

```
Hi Hi Hi
```

Indexing (Accessing Characters)

Each character in a string has an index (position). Index starts at 0.

...many more

abhyasonline.in

Course  
&  
Test Series

Numbers and Strings in Python

```
text = "Python"  
print(text[0]) # First character  
print(text[3]) # Fourth character
```

**Output:**

P  
h

**Negative indexing** is also allowed (from the end):

```
print(text[-1]) # Last character  
print(text[-2]) # Second last character
```

**Slicing (Extracting Substrings)**

We can extract part of a string using string[start:end].

(Note: end index is excluded)

```
text = "Programming"  
print(text[0:6]) # from index 0 to 5  
print(text[3:]) # from index 3 till end  
print(text[:5]) # from start till index 4
```

**Output:**

Progra  
gramming  
Progr

**String Length**

We can find the number of characters using len().

```
text = "Hello World"  
print(len(text))
```

**Output:**

11

**Changing Case in Python Strings**

Python provides built-in string methods to change the case of text.

1. **.upper()** → Converts all characters to UPPERCASE

```
text = "python programming"  
print(text.upper())
```

**Output:**

PYTHON PROGRAMMING

2. **.lower()** → Converts all characters to lowercase

```
text = "HELLO WORLD"  
print(text.lower())
```

...many more

abhyasonline.in

Course  
&  
Test Series

Numbers and Strings in Python

Output:  
hello world

3. `.capitalize()` → Capitalizes only the first letter of the string, rest lowercase  
`text = "welcome to python"`  
`print(text.capitalize())`

Output:  
Welcome to python

4. `.title()` → Capitalizes the first letter of every word  
`text = "python is fun to learn"`  
`print(text.title())`

Output:  
Python Is Fun To Learn

5. `.swapcase()` → Swaps UPPERCASE to lowercase and vice versa  
`text = "PyThOn ProGRaMming"`  
`print(text.swapcase())`

Output:  
pYtHoN pROgrAMMING

String Operations

`a = "Hello"`  
`b = "World"`

# Concatenation  
`print(a + " " + b)`

# Repetition  
`print(a * 3)`

# Length  
`print(len(a))`

# Membership  
`print("H" in a)`  
`print("z" not in a)`

Output:  
Hello World  
HelloHelloHello  
5  
True  
True

Banking &  
Insurance

Central Govt.  
Service

State Govt.  
Services

LAW  
Entrance

MBA  
Entrance

Railways & Metro  
Services

...many more

abhyasonline.in

Course  
&  
Test Series

Numbers and Strings in Python

**Explanation:**

- + joins strings.
- \* repeats them.
- len() gives length.
- in / not in check presence of characters.

**Reversing a String**

**Using String Slicing [::-1] (Most Common & Simple)**

```
text = "Python"  
print(text[::-1])
```

**Output:**

nohtyP

NOTE: Here, [::-1] means start from the end and step backward by -1.

**Solved Example Combining Numbers and Strings**

```
name = "Aarav"  
age = 18
```

```
print("Hello " + name + ", you are " + str(age) + " years old.")
```

**Output:**

Hello Aarav, you are 18 years old.

**Explanation:**

- str(age) converts integer 18 into string so it can be concatenated with other strings.

**Assignment on Strings**

Ques 1: Write a program that takes the string "PYTHON" and prints it in reverse order.

Ques 2: Write a program to join the words "Data" and "Analysis" with a space in between. Print the result.

 CBSE

 ICSE

 NTSE

 Banking & Insurance

 Central Govt. Service

 State Govt. Services

 LAW Entrance

 MBA Entrance

 Railways & Metro Services

...many more

abhyasonline.in