

# Python Basics



# Content

- Comments in Python
- Variable Declaration in Python
- Data Types in Python
- Type Conversion in Python
- Operators in Python

# Comments in Python

---

- In general, Comments are used in a programming language to describe the program or to hide the some part of code from the interpreter.
- Comments in Python can be used to explain any program code. It can also be used to hide the code as well.
- Comment is not a part of the program, but it enhances the interactivity of the program and makes the program readable.

Python supports two types of comments:

- Single Line Comment
- Multi Line Comment

# Comments in Python

Cont..

## Single Line Comment:

In case user wants to specify a single line comment, then comment must start with '#'

### Example:

```
# This is single line comment  
print "Hello Python"
```

### Output:

Hello Python

## Multi Line Comment:

Multi lined comment can be given inside triple quotes.

### Example:

```
"""This is  
    Multiline  
    Comment"""  
print "Hello Python"
```

### Output:

Hello Python

# Variable Declaration in Python

---

- A variable is a named memory location in which we can store values for the particular program.
- In other words, Variable is a name which is used to refer memory location. Variable also known as identifier and used to hold value.
- In Python, We don't need to declare explicitly variable in Python. When we assign any value to the variable that variable is declared automatically.
- In Python, We don't need to specify the type of variable because Python is a loosely typed language.

# Variable Declaration in Python

Cont..

- In loosely typed language no need to specify the type of variable because the variable automatically changes its data type based on assigned value.

## Rules for naming variable:

- Variable names can be a group of both letters and digits, but they have to begin with a letter or an underscore.
- It is recommended to use lowercase letters for variable name. 'SUM' and 'sum' both are two different variables.

### Example: **Vardemo.py**

```
a=10 #integer
b="StudyGlance" #string
c=12.5 #float
print(a)
print(b)
print(c)
```

### output:

```
$python3 Vardemo.py
10
StudyGlance
12.5
```

# Variable Declaration in Python

Cont..

- Python allows us to assign a value to multiple variables and multiple values to multiple variables in a single statement which is also known as multiple assignment.
- Assign single value to multiple variables :

**Example:** Vardemo1.py

```
x=y=z=50  
print x  
print y  
print z
```

**output:**

```
$python3 Vardemo1.py  
50  
50  
50
```

- Assign multiple values to multiple variables :

**Example:** Vardemo2.py

```
a,b,c=5,10,15  
print a  
print b  
print c
```

**output:**

```
$python3 Vardemo2.py  
5  
10  
15
```

# Data Types in Python

---

- In general, Data Types specifies what type of data will be stored in variables. Variables can hold values of different data types.
- Python is a dynamically typed or loosely typed language, hence we need not define the type of the variable while declaring it.
- The interpreter implicitly binds the value with its type.
- Python provides us the **type ()** function which enables us to check the type of the variable.



# Data Types in Python

Cont..

- Python provides following **standard data types**, those are
  - ✓ **Numbers**
  - ✓ **String**

## Numbers:

- Number stores numeric values. Python creates Number type variable when a number is assigned to a variable.

There are three numeric types in Python:

1. int
2. float
3. Complex

# Data Types in Python

Cont..

## 1. int:

Int, or integer, is a whole number, positive or negative, without decimals, of unlimited length.

### Example:

```
a=10  
b=-12  
c=123456789
```

## 2. float:

Float or "floating point number" is a number, positive or negative, containing one or more decimals.

### Example:

```
X=1.0  
Y=12.3  
Z=-13.4
```

## 3. complex:

Complex numbers are written with a "j" as the imaginary part.

### Example:

```
A=2+5j  
B=-3+4j  
C=-6j
```

## String:

- The string can be defined as the sequence of characters represented in the quotation marks. In python, we can use single, double, or triple quotes to define a string.
- In the case of string handling, the operator + is used to concatenate two strings as the operation *"hello"+"python"* returns *"hello python"*.

### Example:

```
S1='Welcome' #using single quotes
```

```
S2="To" #using double quotes
```

```
S3="''Python'' #using triple quotes
```

# Data Types in Python

Cont..

## Example:

**“datatypesdemo.py”**

```
a=10
```

```
b="Python"
```

```
c = 10.5
```

```
d=2.14j
```

```
print("Data type of Variable a :",type(a))
```

```
print("Data type of Variable b :",type(b))
```

```
print("Data type of Variable c :",type(c))
```

```
print("Data type of Variable d :",type(d))
```

## Output:

**python3 datatypesdemo.py**

Datatype of Variable a : <class 'int'>

Datatype of Variable b : <class 'str'>

Datatype of Variable c : <class 'float'>

Datatype of Variable d : <class 'complex'>

# Type Conversion in Python

- Python provides Explicit type conversion functions to directly convert one data type to another. It is also called as **Type Casting** in Python
- Python supports following functions
  1. **int ()** : This function converts **any data type to integer**.
  2. **float()** : This function is used to convert **any data type to a floating point number**.
  3. **str()** : This function is used to convert **any data type to a string**.

## Example: "Typeconversiondemo.py"

```
x = int(2.8)
y = int("3")
z = float(2)
s = str(10)
print(x);print(y)
print(z); print(s)
```

## Output:

```
python3 typeconversiondemo.py
2
3
2
10
```

# Operators in Python

- The operator can be defined as a symbol which is responsible for a particular operation between two operands.
- Python provides a variety of operators described as follows.

## Arithmetic operators :

<b>+</b> (addition)	<b>eg:</b> a=20; b=10 then a + b=30
<b>-</b> (subtraction)	<b>eg:</b> a=20; b=10 then a - b=10
<b>*</b> (multiplication)	<b>eg:</b> a=20; b=10 then a * b=200
<b>/</b> (divide)	<b>eg:</b> a=20; b=10 then a / b=2
<b>%</b> ( remainder)	<b>eg:</b> a=20; b=10 then a % b=0
<b>//</b> (floor division)	<b>eg:</b> a=24; b=7 then a // b=3
<b>**</b> (exponent)	<b>eg:</b> a=2; b=3 then a ** b=8

## Membership operators :

**in** (True, If the value is present in the data structure)

**not in** (True, If the value is not present in the data structure)

# Operators in Python

Cont...

## Comparison operators :

**==** (Equal to)  
**!=** (Not equal to)  
**<=** (Less than or equal)  
**>=** (Greater than or equal)  
**<** (Less than)  
**>** (Greater than)

## Bitwise operators :

**&** (binary and)  
**|** (binary or)  
**^** (binary xor)  
**~** (negation)  
**<<** (left shift)  
**>>** (right shift)

## Assignment operators :

**=** (Assigns to)  
**+=** (Assignment after Addition)  
**-=** (Assignment after Subtraction)  
**\*=** (Assignment after Multiplication)  
**/=** (Assignment after Division)  
**%=** (Assignment after Modulus)  
**\*\*=** (Assignment after Exponent)  
**//=** (Assignment after floor division)

## Logical operators :

**and** (logical and)  
**or** (logical or)  
**not** (logical not)

## Identity operators :

**is** (Returns true if both variables are the same object)  
  
**is not** (Returns true if both variables are not the same object)