Python Programming

Course Curriculum

* What is Python?
* Python Features
* Python History
* Python Applications
* Python Install
* Variables
* Data Types
* Keywords
* Literals
* Operators
* Loops
* Conditional Statements
* Control Statements
* Comments
* Type casting
* Strings
* Data Structures
* Functions
* Arrays
* Numpy
* Pandas
* Matplotlib
* Seaborn
* Files
* Modules
* Oops Concepts
* HTML
* CSS
* JAVASCRIPT
* JQUERY
* Bootstrap
* Flask
* GUI Frameworks
* **What is Python?**
* Python is a general purpose, dynamic Typed, high-level and interpreted programming language. It supports object oriented programming approach to develop applications it is simple and easy to learn and provides lots of high-level Data structures.(LIST , TUPLE, DICTINARY set)
* Python is easy to learn yet powerful and versatile scripting language. Which makes it attractive for Application Development
* Python syntax and dynamic typing with its interpreted nature make it an ideal language for scripting and rapid application development.
* Python supports multiple programming pattern including object-oriented imperative and functional or procedural programming styles.

**Python Features**

1) **Easy to Learn and use:**

Python is easy to learn as compared to other programming languages. Its syntax is straight forward the same as the English language. There is no use of the semicolon or curly-bracket, the indentation defines the code block. It is the recommended programming language for beginners.

### 2) Expressive Language:

Python can perform complex tasks using a few lines of code. A simple example, the hello world program you simply type print **("Hello World")**. It will take only one line to execute, while Java or C takes multiple lines.

Main()

{

Printf(“hello world”);

}

Class Hello

{

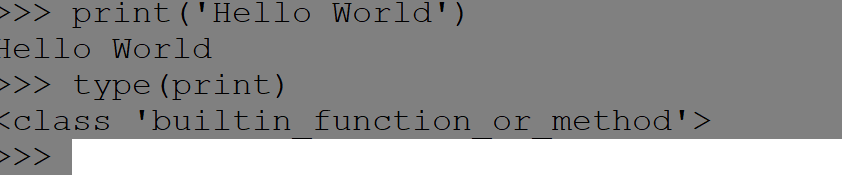
Public void main(String args[0, 1 2])

{

System.out.println(“hello”);

}

}



### 3) Interpreted Language:

Python is an interpreted language; it means the Python program is executed one line at a time. The advantage of being interpreted language, it makes debugging easy and portable.

### 4) Cross-platform Language:

Python can run equally on different platforms such as Windows, Linux, UNIX, and Macintosh, etc. So, we can say that Python is a portable language. It enables programmers to develop the software for several competing platforms by writing a program only once.

### 5) Free and Open Source:

Python is freely available for everyone. It is freely available on its official website [www.python.org](https://www.python.org/). It has a large community across the world that is dedicatedly working towards make new python modules and functions. Anyone can contribute to the Python community. The open-source means, "Anyone can download its source code without paying any penny."

C

Windows (TCS)

### 6) Object-Oriented Language (JAVA)

Python supports object-oriented language and concepts of classes and objects come into existence. It supports inheritance, polymorphism, and encapsulation, etc. The object-oriented procedure helps to programmer to write reusable code and develop applications in less code.

7) **Extensible: .c .java .py .html**

**Class A**

**{**

**Class B extends A**

**{**

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our Python code. It converts the program into byte code, and any platform can use that byte code.

1 BYTE =8 BITS

JVM OR JDK

### 8) Large Standard Library:

### DATA SCIENCE

### ML

### DL

### AI

### JS

It provides a vast range of libraries for the various fields such as machine learning, web developer, and also for the scripting. There are various machine learning libraries, such as Tensor flow, Pandas, Numpy, Keras, and Pytorch, et0c. Django, flask, pyramids are the popular framework for Python web development.

### 9) GUI Programming Support:

### GUI

Graphical User Interface is used for the developing Desktop application. PyQT5, Tkinter, Kivy are the libraries which are used for developing the web application.

### 10) Integrated:

It can be easily integrated with languages like C, C++, and JAVA, etc. Python runs code line by line like C, C++ Java. It makes easy to debug the code.

### 11) Embeddable:

The code of the other programming language can use in the Python source code. We can use Python source code in another programming language as well. It can embed other language into our code.

### Python History:

* Python laid its foundation in the late 1980s.
* The implementation of Python was started in December 1989 by **Guido Van Rossum** at CWI in Netherland.
* In February 1991, **Guido Van Rossum** published the code (labeled version 0.9.0) to alt.sources.
* In 1994, Python 1.0 was released with new features like lambda, map, filter, and reduce.
* Python 2.0 added new features such as list comprehensions, garbage collection systems.
* On December 3, 2008, Python 3.0 (also called "Py3K") was released. It was designed to rectify the fundamental flaw of the language.
* ABC programming language is said to be the predecessor of Python language, which was capable of Exception Handling and interfacing with the Amoeba Operating System.
* The following programming languages influence Python:
  + ABC language.
  + Modula-3

**Why the Name Python?**

There is a fact behind choosing the name [Python](https://www.javatpoint.com/python-tutorial)

. **Guido van Rossum** was reading the script of a popular BBC comedy series "**Monty Python's Flying Circus**". It was late on-air 1970s.

Van Rossum wanted to select a name which unique, sort, and little-bit mysterious. So he decided to select naming Python after the **"Monty Python's Flying Circus"** for their newly created programming language.

The comedy series was creative and well random. It talks about everything. Thus it is slow and unpredictable, which made it very interesting.

Python is also versatile and widely used in every technical field, such as [Machine Learning](https://www.javatpoint.com/machine-learning)

, [Artificial Intelligence](https://www.javatpoint.com/artificial-intelligence-tutorial)

, Web Development,

[Mobile Application](https://www.javatpoint.com/javatpoint.com/mobile-application-testing)

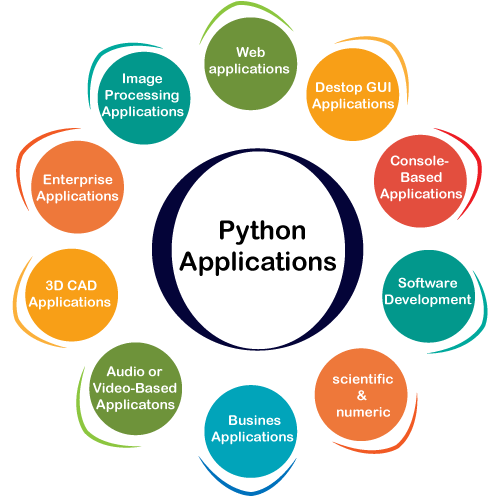
, Desktop Application,

Scientific Calculation, etc.

**PYTHON APPLICATIONS:**

Python is known for its general-purpose nature that makes it applicable in almost every domain of software development. Python makes its presence in every emerging field. It is the fastest-growing programming language and can develop any application.

Here, we are specifying application areas where Python can be applied.



**Flask , django**

**WEB APPLICATIONS**

We can use Python to develop web applications. It provides libraries to handle internet protocols such as HTML and XML, JSON, Email processing, request, beautiful Soup, Feedparser, etc. One of Python web-framework named Django is used on **Instagram**. Python provides many useful frameworks, and these are given below:

* Django and Pyramid framework(Use for heavy applications)
* Flask and Bottle (Micro-framework)
* Plone and Django CMS (Advance Content management

**Desktop GUI Applications**

The GUI stands for the Graphical User Interface, which provides a smooth interaction to any application. Python provides a **Tk GUI library** to develop a user interface. Some popular GUI libraries are given below.

* Tkinter or Tk
* wxWidgetM
* Kivy (used for writing multitouch applications )
* PyQt or Pyside

### Console-based Application

### Console-based applications run from the command-line or shell. These applications are computer program which are used commands to execute. This kind of application was more popular in the old generation of computers. Python can develop this kind of application very effectively. It is famous for having REPL, which means **the Read-Eval-Print Loop** that makes it the most suitable language for the command-line applications.

Python provides many free library or module which helps to build the command-line apps. The necessary **IO** libraries are used to read and write. It helps to parse argument and create console help text out-of-the-box. There are also advance libraries that can develop independent console apps.

### Software Development:

Python is useful for the software development process. It works as a support language and can be used to build control and management, testing, etc.

* **SCons** is used to build control.
* **Buildbot** and **Apache** Gumps are used for automated continuous compilation and testing.
* **Round** or **Trac** for bug tracking and project management.

### Scientific and Numeric:

This is the era of Artificial intelligence where the machine can perform the task the same as the human. Python language is the most suitable language for Artificial intelligence or machine learning. It consists of many scientific and mathematical libraries, which makes easy to solve complex calculations.

numpy

Implementing machine learning algorithms require complex mathematical calculation. Python has many libraries for scientific and numeric such as Numpy, Pandas, Scipy, Scikit-learn, etc. If you have some basic knowledge of Python, you need to import libraries on the top of the code. Few popular frameworks of machine libraries are given below.

* SciPy
* Scikit-learn
* NumPy
* Pandas
* Matplotlib

### Business Applications:

Business Applications differ from standard applications. E-commerce and ERP are an example of a business application. This kind of application requires extensively, scalability and readability, and Python provides all these features.

Oddo is an example of the all-in-one Python-based application which offers a range of business applications. Python provides a **Tryton** platform which is used to develop the business application.

### Audio or Video-based Applications:

### Python is flexible to perform multiple tasks and can be used to create multimedia applications. Some multimedia applications which are made by using Python are ****TimPlayer, cplay,**** etc. The few multimedia libraries are given below.

* Gstreamer
* Pyglet
* QT Phonon

**3D CAD Applications:**

### The CAD (Computer-aided design) is used to design engineering related architecture. It is used to develop the 3D representation of a part of a system. Python can create a 3D CAD application by using the following functionalities.

* Fandango (Popular )
* CAMVOX
* HeeksCNC
* AnyCAD
* RCAM

### Enterprise Applications:

### Python can be used to create applications that can be used within an Enterprise or an Organization. Some real-time applications are OpenERP, Tryton, Picalo, etc.

### Image Processing Application

Python contains many libraries that are used to work with the image. The image can be manipulated according to our requirements. Some libraries of image processing are given below.

* OpenCV
* Pillow
* SimpleITK

In this topic, we have described all types of applications where Python plays an 11essential role in the development of these applications. In the next tutorial, we will learn more concepts about Python.

# **How to Install Python**

The first step is to learn how to install or update Python on a local machine or computer. In this tutorial, we will discuss the installation of Python on various operating systems.

## **Installation on Windows**

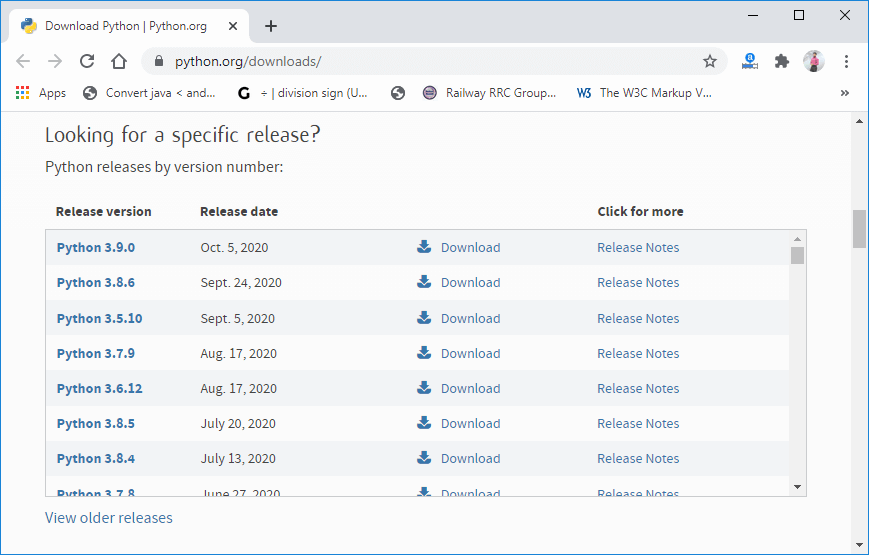
Visit the link [https://www.python.org/downloads/](https://www.python.org/downloads/" \t "_blank)

To download the latest release of [Python](https://www.javatpoint.com/python-tutorial)

. In this process, we will install Python 3.8.6 on our [Windows operating system](https://www.javatpoint.com/windows)

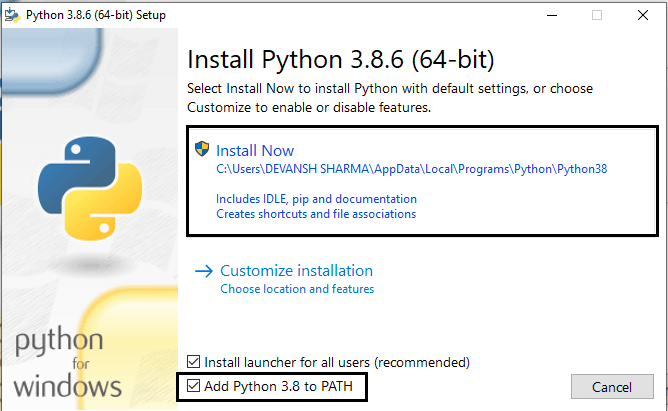
. When we click on the above link, it will bring us the following page.

**Step - 1: Select the Python's version to download.**

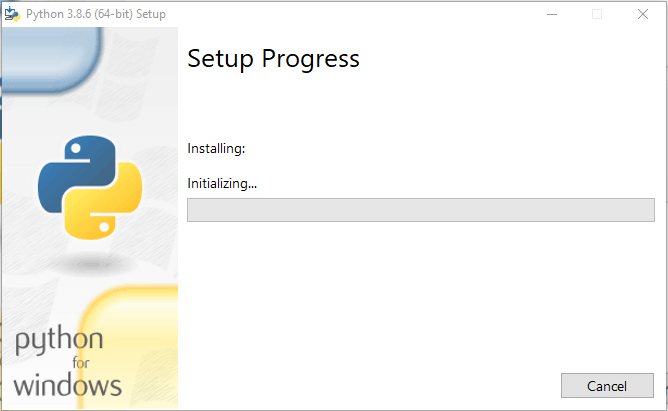


**Step - 2: Click on the Install Now**

Double-click the executable file, which is downloaded; the following window will open. Select Customize installation and proceed. Click on the Add Path check box, it will set the Python path automatically.



**Step - 3 Installation in Process**



Now, try to run python on the command prompt. Type the command python -version in case of python3.

Variables:

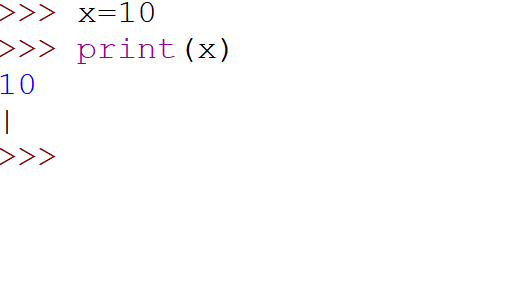
What is Python Variable?

* Python Variable is a name that is used to refer to memory location.
* Python variable is also known as an identifier and used to hold value.
* In Python, we don't need to specify the type of variable because Python is a infer language and smart enough to get variable type.
* Variable names can be a group of both the letters and digits, but they have to begin with a letter or an underscore.
* It is recommended to use lowercase letters for the variable name.

Syntax for Variable:

Variable\_name=value

Example Code:



## **Identifier Naming:**

## An Identifier is used to identify the literals used in the program. The rules to name an identifier are given below.

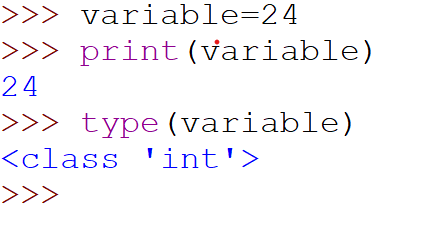
* The first character of the variable must be an alphabet or underscore ( \_ ).
* All the characters except the first character may be an alphabet of lower-case(a-z), upper-case (A-Z), underscore, or digit (0-9).
* Identifier name must not contain any white-space, or special character (!, @, #, %, ^, &, \*).
* Identifier name must not be similar to any keyword defined in the language.
* Identifier names are case sensitive; for example, my name, and MyName is not the same.
* Examples of valid identifiers: a123, \_n, n\_9, etc.
* Examples of invalid identifiers: 1a, n%4, n 9, etc.

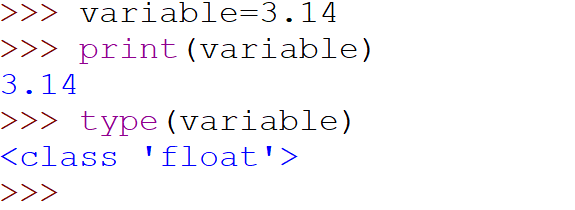
## **Declaring Variable and Assigning Values:**

## Python does not bind us to declare a variable before using it in the application. It allows us to create a variable at the required time.

* We don’t need to declare explicitly variable in Python
* The equal operator is used to assign value to a variable

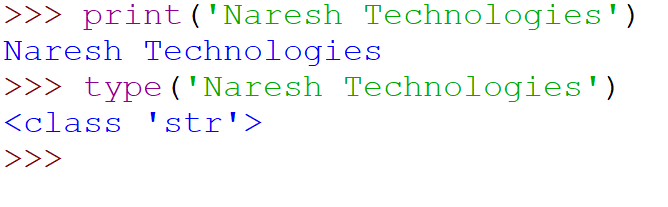
Example:

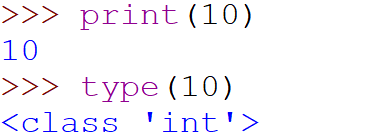




Object References:

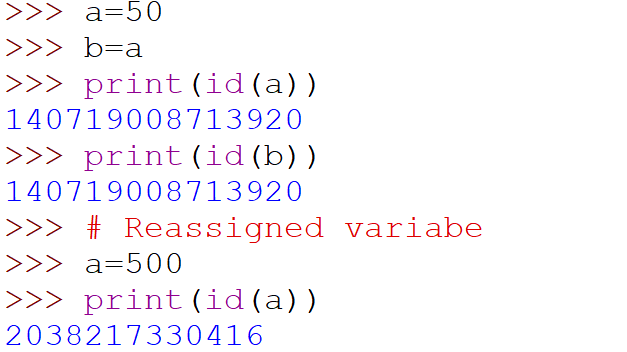
python is the highly object programing language, that’s why every data item belongs to a specific type





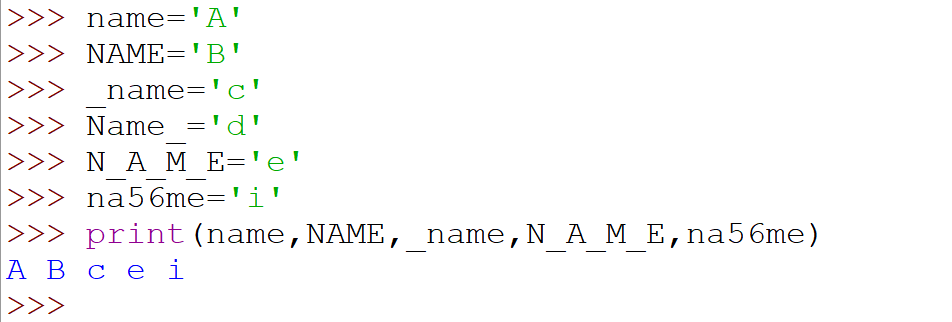
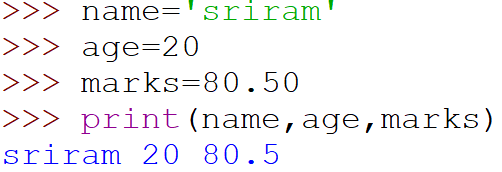
Object Identity:

* In python every created object identifies uniquely in python.
* Python provides the guaranteed that non two objects will have the same identifier.
* The built in function is used to identify the object identifier



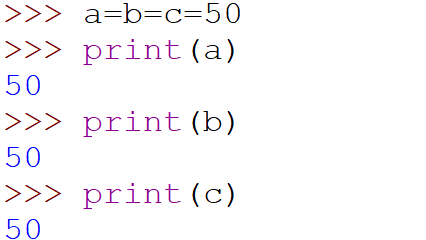
Variable Names:

Variable names can be any length can have uppercase, lowercase (A to Z, ato z), The digit (0-9), and underscore character(\_).

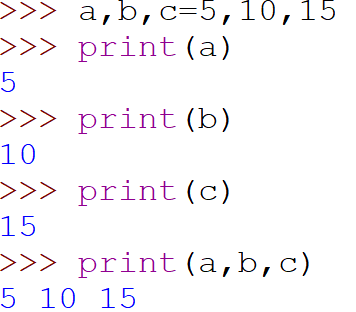
 

Multiple Assignment:

* Python allows us to assign a value to multiple variables in a single statement, which is also known as multiple assignments
* Multiple assignment in two ways, either by assigning a single value to multiple variables or assigning multiple values
* Assigning single value to multiple variables



* Assigning multiple values to multiple variables

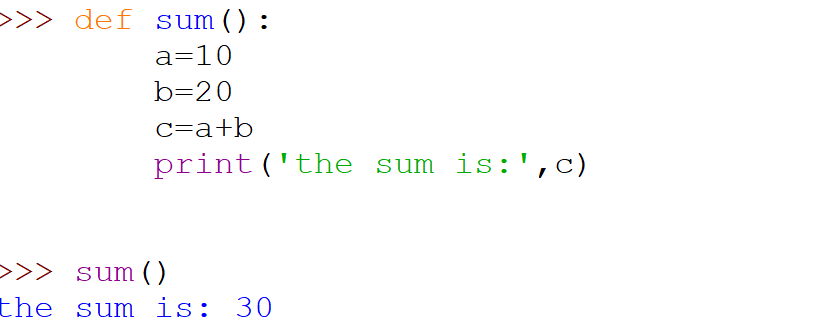


Variable Types:

1) Local Variables:

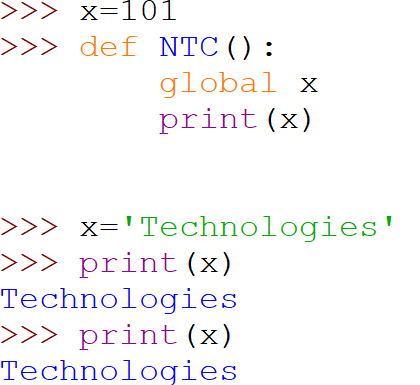
Localvariablesarethevariablesthatdeciaredinsidethefunctionandhaves

Scope within the function



2) Global Variables:

Global variables are the variable that deciared outside the function and have scope is in the entire progaram



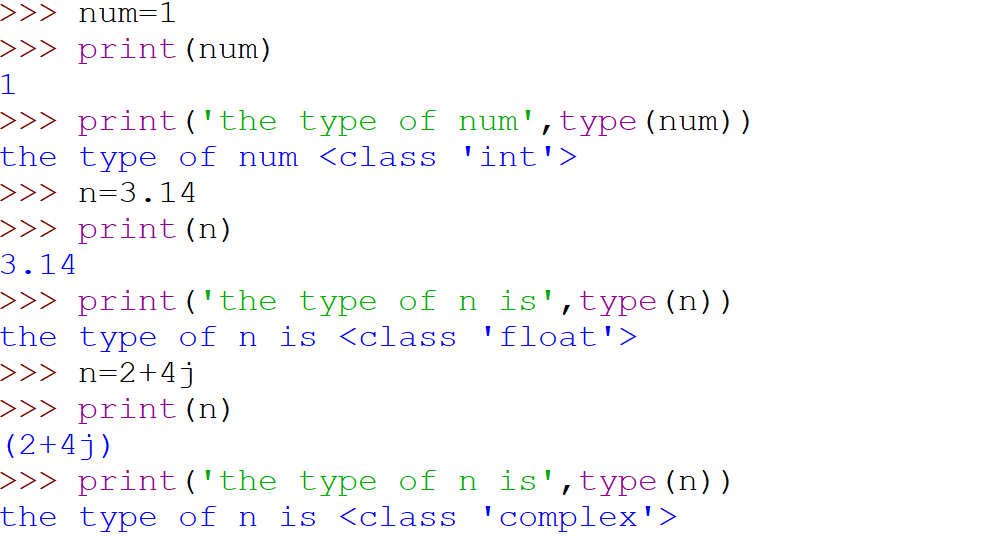
Int float char double longdouble

* Data Types:



Numbers:

Number stores numeric values. The integer, float, and complex values belong to a Python Numbers data-type. Python provides the **type()** function to know the data-type of the variable. Similarly, the is instance () function is used to check an object belongs to a particular class.



### Sequence Type:

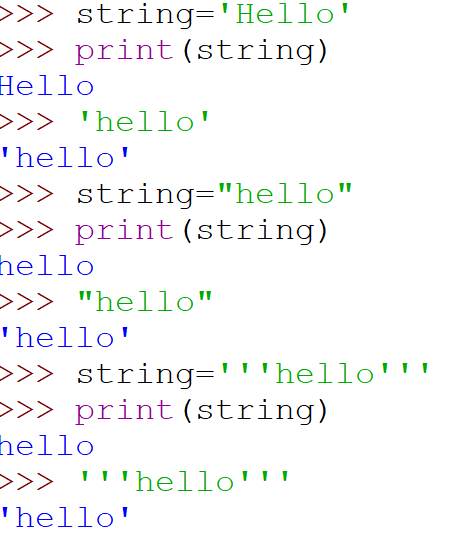
### String

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.

String handling in Python is a straightforward task since Python provides built-in functions and operators to perform operations in the string.

In the case of string handling, the operator + is used to concatenate two strings as the operation "hello"+" python" returns "hello python".

The operator \* is known as a repetition operator as the operation "Python" \*2 returns 'Python'.

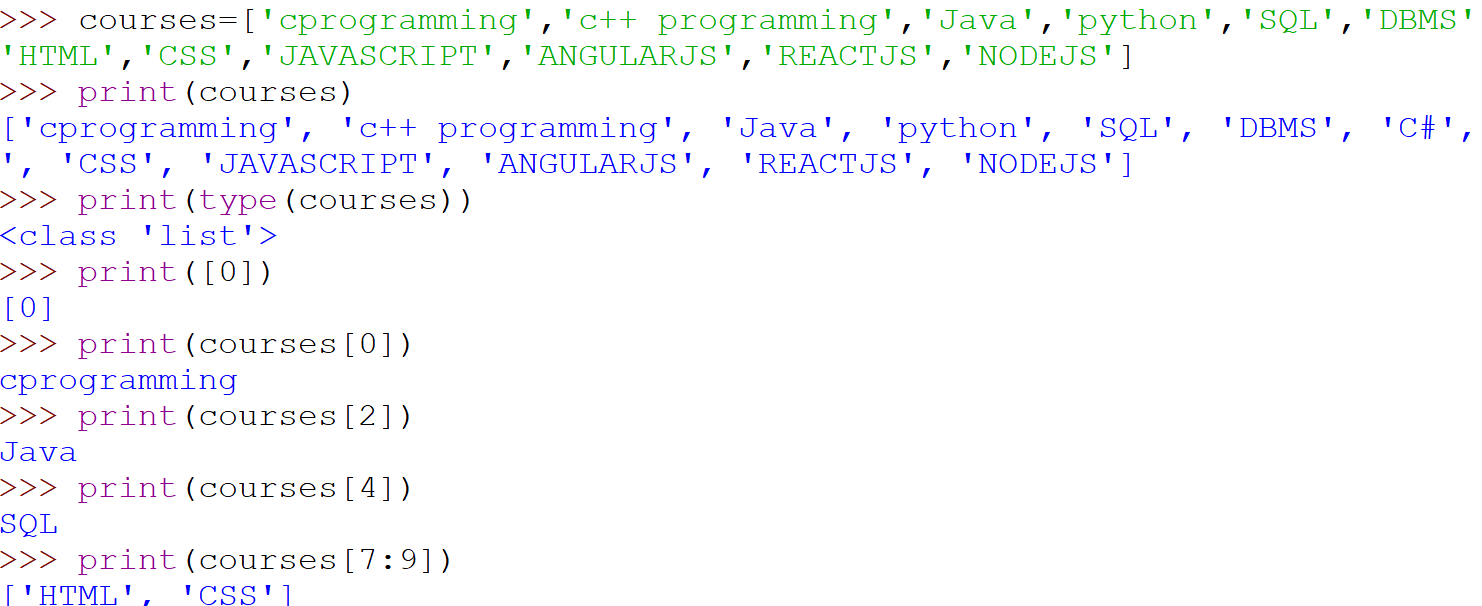


### List

### Int a[5];

Python Lists are similar to arrays in C. However, the list can contain data of different types. The items stored in the list are separated with a comma (,) and enclosed within square brackets [].

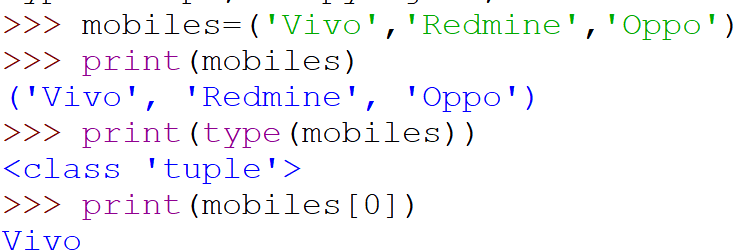
We can use slice [:] operators to access the data of the list. The concatenation operator (+) and repetition operator (\*) works with the list in the same way as they were working with the strings.



### Tuple

A tuple is similar to the list in many ways. Like lists, tuples also contain the collection of the items of different data types. The items of the tuple are separated with a comma (,) and enclosed in parentheses ().

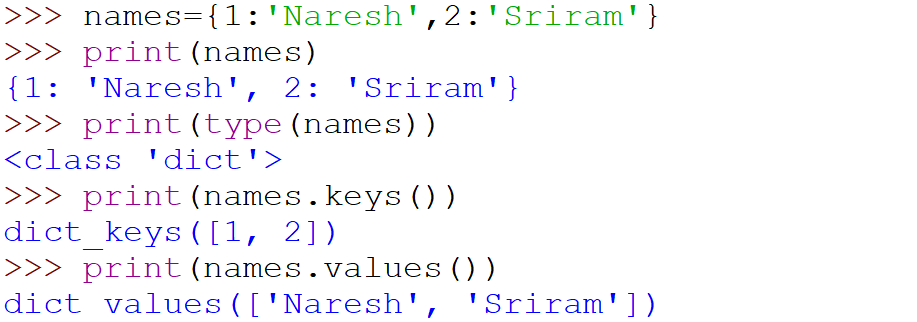
A tuple is a read-only data structure as we can't modify the size and value of the items of a tuple.



### Dictionary

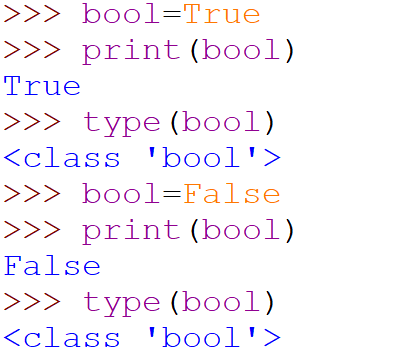
Dictionary is an unordered set of a key-value pair of items. It is like an associative array or a hash table where each key stores a specific value. Key can hold any primitive data type, whereas value is an arbitrary Python object.

The items in the dictionary are separated with the comma (,) and enclosed in the curly braces {}.

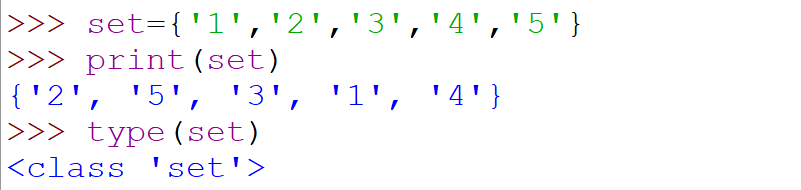


### Boolean

Boolean type provides two built-in values, True and False. These values are used to determine the given statement true or false. It denotes by the class bool. True can be represented by any non-zero value or 'T' whereas false can be represented by the 0 or 'F'. Consider the following example.

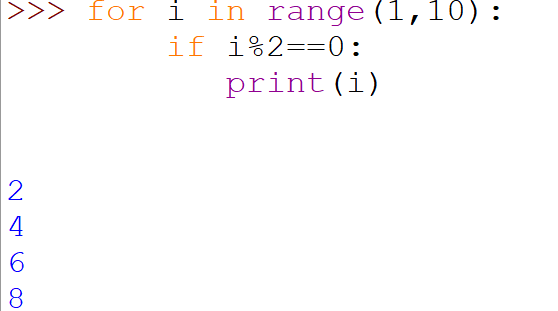


* SETS:
* Python Set is the unordered collection of the data type. It is iterable, mutable (can modify after creation), and has unique elements. In set, the order of the elements is undefined; it may return the changed sequence of the element. The set is created by using a built-in function  **set(),** or a sequence of elements is passed in the curly braces and separated by the comma. It can contain various types of values. Consider the following example.



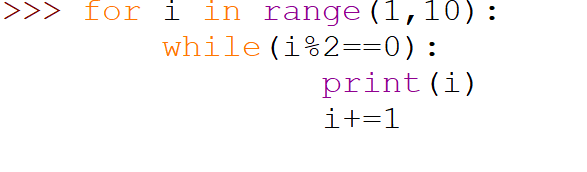
* Keywords
* Every scripting language has designated words or keywords, with particular definitions and usage guidelines
* Python is no exception. The fundamental constituent elements of any python program are python keywords
* Python keywords are unique words reserved with defined meanings and functions that we can only apply for these functions
* For keyword:

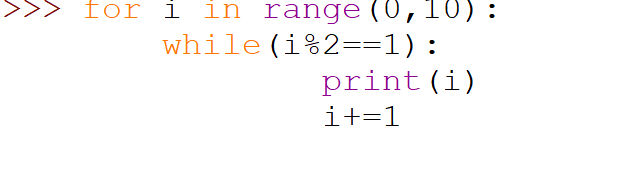
For is a looping statement in python that mainly used for number of iterative Statements



* While Keyword:

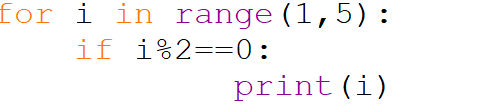
While is a looping statement in python that mainly used for set of statements as long as a condition is true





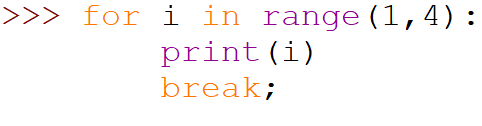
* If keyword:

If keyword is used to create conditional statements and allows you to execute a block of code only if a condition is true



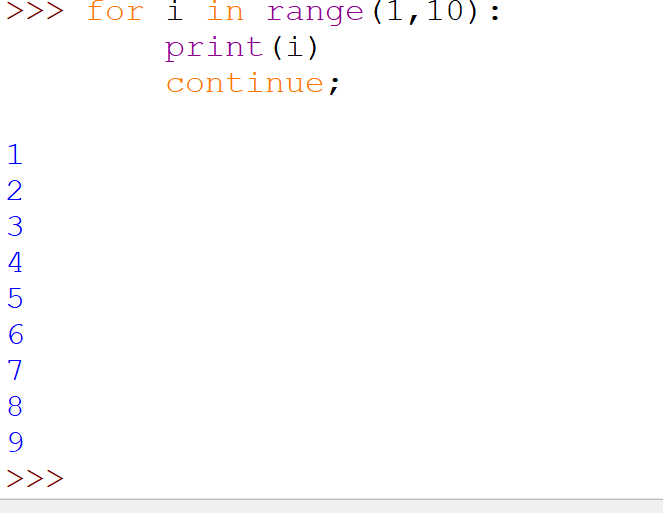
* Break:

The break keyword is used to break out a for loop, or a while loop

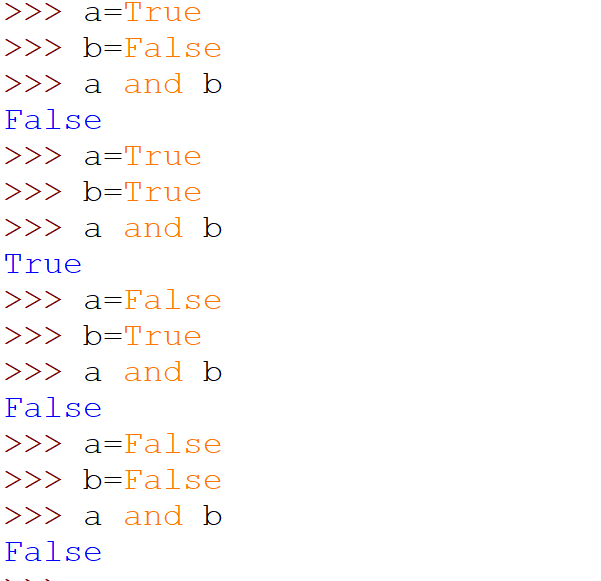


* Continue:

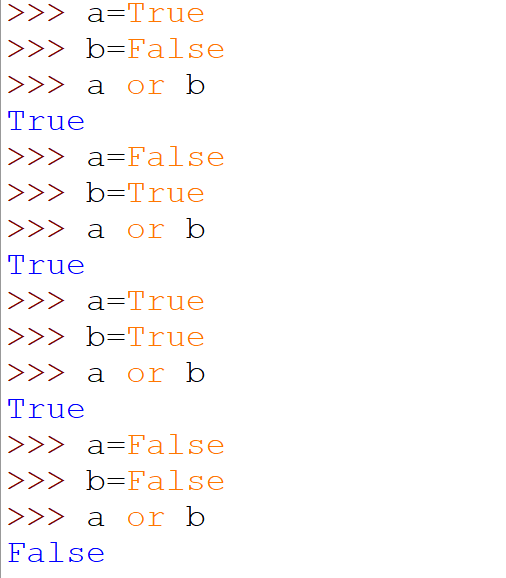
The continue keyword is used to end the current iteration in a for loop or a while loop, and continue to the next iteration



* And Keyword
  + The and keyword is a logical operator.
  + Logical operators are used to combine conditional statements
  + If both statements return true, otherwise it will return false



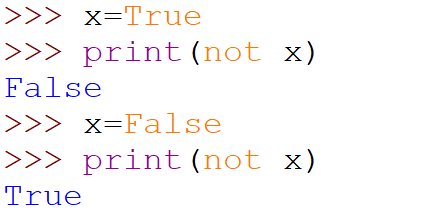
* Or keyword:
* The or keyword is a logical operator
* Logical operators are used to combine conditional statements
* The return value will be true if one of the statements return true otherwise it will return false



Not keyword:

The not keyword is a logical operator

The return value will be true if the statements are not true otherwise it will return false



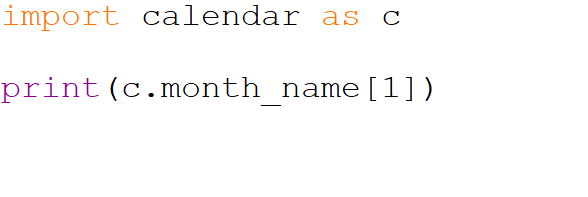
Except Keyword:

The except keyword is used in try...except blocks. It defines a block of code to run if the try block raises an error.

You can define different blocks for different error types, and blocks to execute if nothing went wrong,

AS Keyword:

The as keyword is used to create an alias.

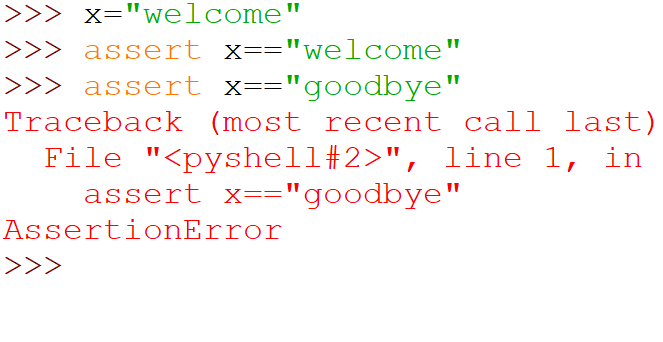


Assert keyword

The assert keyword is used when debugging code.

The assert keyword lets you test if a condition in your code returns True, if not, the program will raise an Assertion Error.

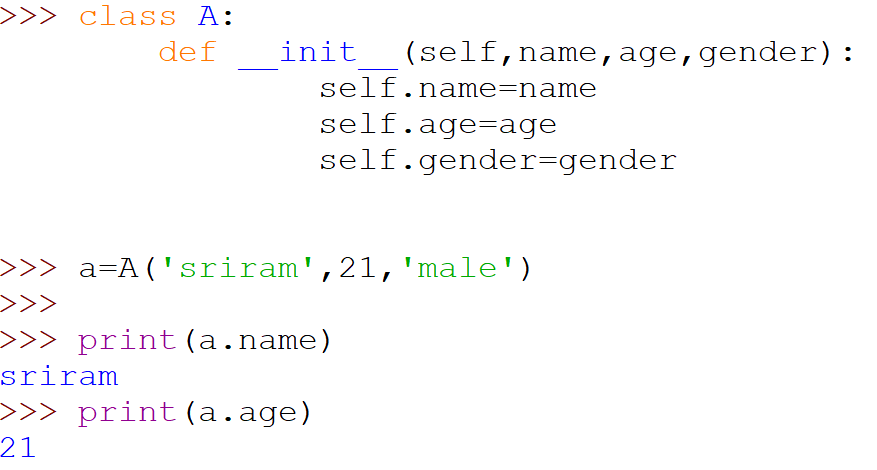
You can write a message to be written if the code returns False, check the example below.



Class:

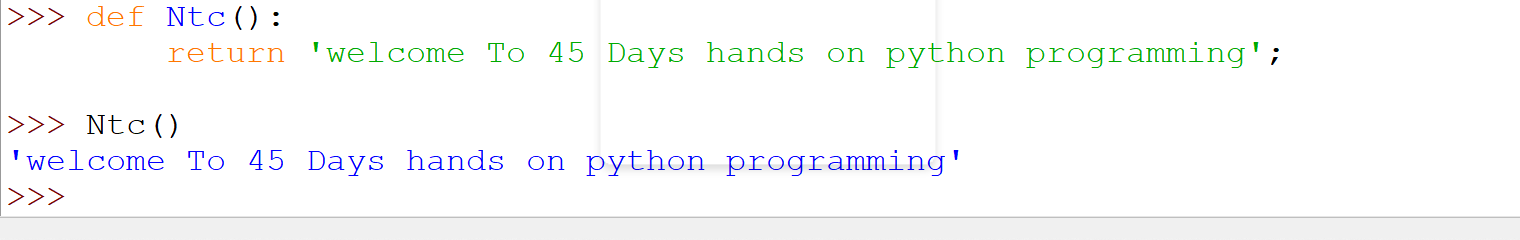
The class keyword is used to create a class.

A class is like an object constructor. See the example below to see how we can use it to create an object.



Def :

The def keyword is used to create, (or define) a function.



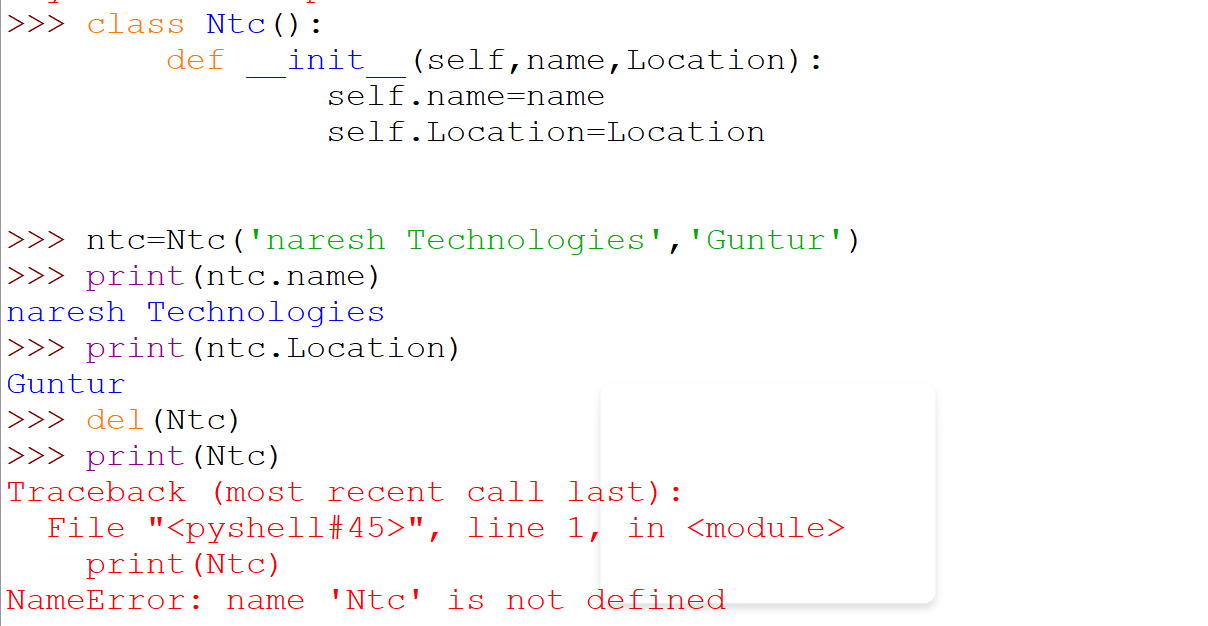
Void sum(int a, int b)

Main()

{

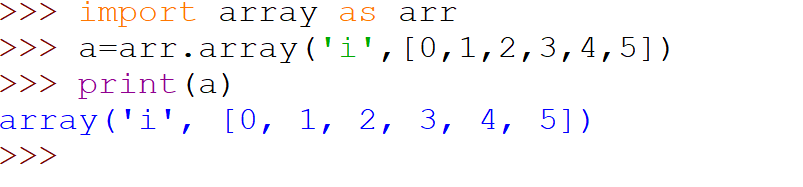
Del:

The del keyword is used to delete objects. In Python everything is an object, so the del keyword can also be used to delete variables, lists, or parts of a list etc.



* Import Keyword:

The import keyword is used to import modules.



* FROM Keyword:

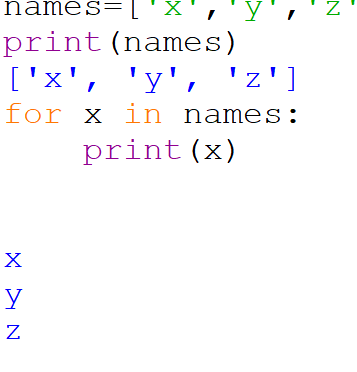
The from keyword is used to import only a specified section from a module.

* In

The in keyword has two purposes:

The in keyword is used to check if a value is present in a sequence (list, range, string etc.).

The in keyword is also used to iterate through a sequence in a for loop:

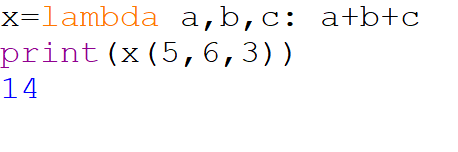


* **Lambda**

The lambda keyword is used to create small anonymous functions

A lambda function can take any number of arguments, but can only have one expression

The expression is evaluated and the result is returned.



# **Literals**

Python Literals can be defined as data that is given in a variable or constant.

### String literals:

### String literals can be formed by enclosing a text in the quotes. We can use both single as well as double quotes to create a string.

### Example program

### 

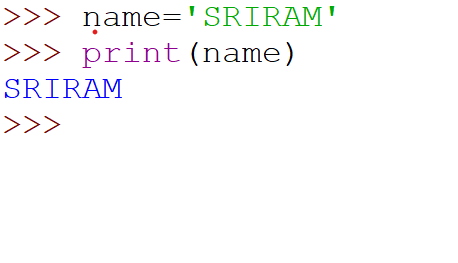
### 

**Types of Strings:**

There are two types of Strings supported in Python:

**a) Single-line String**-

Strings that are terminated within a single-line are known as Single line Strings.



**b) Multi- line String:**

A piece of text that is written in multiple lines is known as multiple lines string



### c) Using Triple Quotation:

### 

### Numeric literals:

### Numeric literals are immutable

### Different Types of Numerical types

### Int (signed integers)

### Numbers can be both positive and negative with no fractional

### Long (long integers):

### Intergers of size followed by lowercase or uppercase

### Float:

### Real numbers with both integer and factional

### Complex:

### In the form of a+bj where a forms the a is real part and b forms the imaginary part

### Boolean Literals:

### Boolean literals means either true or false

### List literals:

### List contains collection of items of different data types

### List are mutable (modifiable)

### Dictionary literals

### In Dictionary store the data in the key-value pair

### Tuple literals:

### Tuple is a collection of different data types

### It is immutable which means it cannot be modified after creation

### Set literals:

### Set is a collection of unordered dataset

### Operators

### What is Operators?

### The operator can be defined as a symbol which is responsible for a particular operation between two operands

### a+b

### Python provides a various of operators

### Arithmetic Operators

### Comparison Operators

### Assignment Operators

### Logical Operators

### Bitwise Operators

### Membership Operators

### Identity Operators

### Arithmetic Operators:

### Arithmetic Operators are used to perform arithmetic operations between two operands. It includes + (addition), - (Subtraction), \* (Multiplication), / (divide), % (reminder), // (floor division) and exponent (\*\*) operators

### 

### Comparison Operators:

### Comparison Operators are used to comparing the value of the two operands and return Boolean true or false

### 

### 

### 

### 

### 

### 

### Assignment Operators:

### The assignment operators are used to assign the value of the right expression to the left operand

### 

### 

### 

### 

### 

### 

### 

### BitWise Operators:

### The bitwise operators perform bit by bit operation on the values of the two operands

### A+b

### A b a&b

### T T T

### T F F

### F T F

### F F F

### Bitwise &:

### 

### Bitwise Or |

### 

### X y x or y

### T T T

### F T T

### F F F

### T F T

### Bitwise xor:

### 

### ~(negation):

### 

### 

### 

### Logical Operators:

### The logical operators are used primarily in the expression evaluation

### 

### Membership Operators:

### Python Membership operators are used to check the membership of value inside a python data structure

### In or not in

### 

### Identity operators:

### The identity operators are used to decide whether an element certain class or type

### 

### *Looping Statements(For , while , do while)*

### *What is for loop?*

### *In python, the for loop is used to iterate over a sequence such as a list, string, tuple other iterable objects such as range*

### *for loop in Python*

### *Syntax for for loop*

### *For I in range/sequence:*

### *Statement 1*

### *Statement 2*

### *Statement n*

### 

### For loop with range ()

### The range() function returns a sequence of numbers starting from 0 (by default)

### If the initial limit is not specified and it increments by 1

### The range() function is used with a loop to specify the range how many times the code block will be executed

### 

### 

### 

### 

### 

### 

### 

### 

### 

### While

### The Python while loop iteration of a code block is executed as long as the given condition, i.e., conditional\_ expression, is true.

### If we don't know how many times we'll execute the iteration ahead of time, we can write an indefinite loop.

### ****Syntax of Python****

### while conditional expression:

### Code block of while

### 

### 

### 

### #Python program to find the LCM of Two Numbers

### n1=int (input ("enter n1 value ::"))

### n2=int (input ("enter n2 value ::"))

### If (n1>n2):

### m=n1

### else:

### m=n2

### while(True):

### if (m%n1==0 and m%n2==0):

### print(m)

### break;

### m=m+1

### #Python Program To Find Factorial of Given Number

### n=int (input ("enter a number ::"))

### f=1

### i=1

### While i<=n:

### f=f\*i

### i=i+1

### print ("factorial of ", n, " is",f)

### n=15

### # initializing summation and a counter for iteration

### Summation=0

### Count=1

### # specifying the condition of the loop

### Summation=count\*\*2+summation

### # Incrementing the counter

### Count=count+1

### # print the final sum

### Print ("The sum of squares is”, summation)

### n=[34,12,54,23,75,34,11]

### def prime number(n):

### c=0

### i=2

### While i<=n/2:

### if n % i==0:

### c=1

### break

### i=i+1

### if c==0:

### print (f"{n} is a prime number")

### else:

### Print(f"{n} is not a prime number")

### 

### 

### For i in n:

### Prime number(i)

### # Multiplication Table using while loop

### n=21

### c=1

### print ("The Multiplication Table of:",n)

### While c<=10:

### ans=n\*c

### print(n, 'x', c,'=', ans)

### c+=1

### n=21

### c=1

### Print ("The Multiplication Table of:",n)

### While c<=10:

### Ans=n\*c

### Print(n, 'x', c,'=', ans)

### c+=1

### c=1

### While c<=5:

### Print ("Naresh Technologies::")

### c=c+1

### i=5

### While i>=1:

### Print ("Python", i)

### i=i-1

### i=1

### j=1

### While i<=5:

### Print ("Naresh")

### While j<=4:

### Print ("Rocks")

### j=j+1

### i=i-1

### i=1

### while i<=5:

### Print (i)

### i=i+1

**Conductional Statements (if, elif, else if)**

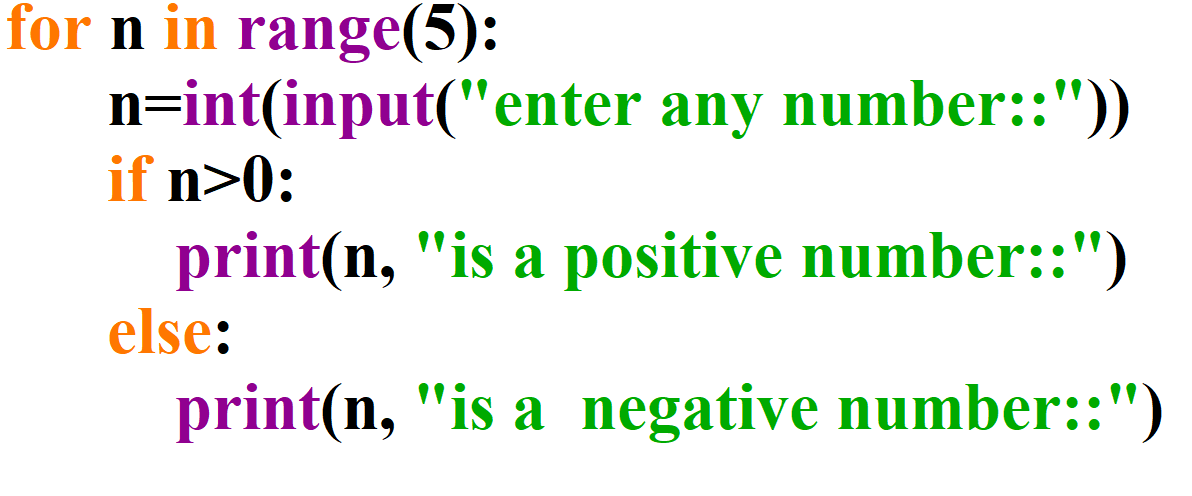
If Statement:

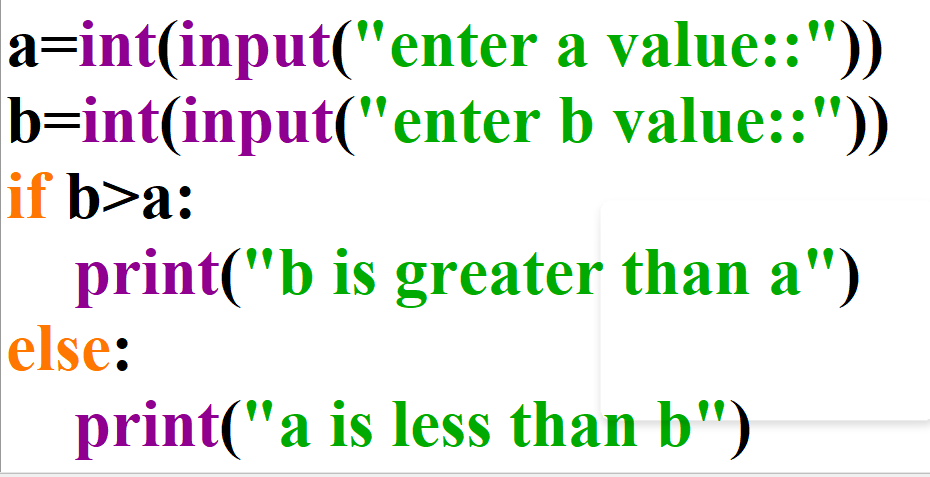
The If statement is the most fundamental decision-making statement, in which the code is executed based on whether it meets the specified condition. It has a code body that only executes if the condition in the if statement is true. The statement can be a single line or a block of code.

Syntax:

If expression

Statement





**If else statement**

This statement is used when both the true and false parts of a given condition are specified to be executed. When the condition is true, the statement inside the if block is executed; if the condition is false, the statement outside the if block is executed.

Syntax for if else:

if condition :

#Will executes this block if the condition is true

else :

#Will executes this block if the condition is false

Naresh=int (input ("enter any number you want ::"))

If Naresh>=0:

Print ("Positive or Zero")

else:

print ("Negative number")

**if elif else statement:**

In this case, the If condition is evaluated first. If it is false, the Elif statement will be executed; if it also comes false, the Else statement will be executed.

Syntax for if elif else:

if condition :

Body of if

elif condition :

Body of elif

else:

Body of else

NareshTech=int (input ("enter any number you want choose ::"))

If nareshTech>0:

print ("Positive number")

Elif nareshTech==0:

Print ("Zero")

Else:

print("Negative Number")

**Nested if statement:**

A Nested IF statement is one in which an If statement is nestled inside another If statement. This is used when a variable must be processed more than once. If, If-else, and If…elif…else statements can be used in the program. In Nested If statements, the indentation (whitespace at the beginning) to determine the scope of each statement should take precedence.

Syntax for nested if:

if (condition1):

#Executes if condition 1 is true

if (condition 2):

#Executes if condition 2 is true

#Condition 2 ends here

#Condition 1 ends here

p=int (input ("enter p value ::"))

If p>=0:

If p==0:

print("Zero")

else:

print("Positive number")

else:

print("Negative number::")

Price=int(input("enter price::"))

quantity=int(input("enter quantity::"))

amount=price\*quantity

if amount>200:

Print ("The amount is greater than 1000::")

Else:

If amount < 800:

Print ("The amount is between 800 and 1000")

Elif amount< 600:

Print ("The amount is between 600 and 1000")

Else:

Print ("The amount is between 400 and 1000")

If amount==200:

Print ("Amount is 200")

Else:

Print ("Amount is less than 200")

**Control Statements (Continue, Break, and pass)**

Continue Statement:

When the program encounters a continue statement, it will skip the statements which are present after the continue statement inside the loop and proceed with the next iterations.

for ch in 'python':

if ch=='y':

continue;

print("Current character: ",ch)

for ch in 'java':

if ch=='j':

continue;

print("Current character:", ch)

**Break Statement:**

The break statement is used to terminate the loop containing it,

for ch in 'Data Science':

if ch=='D':

break;

print("Current character: ", ch)

**Pass statement**

Pass statement is python is a null operation, which is used when the statement is required syntactically.

for ch in 'fullstack':

if ch=='k':

pass

print("Current character: ", ch)