\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

03/06/2024 - Monday

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**What is Python?**

->Python is a popular programming language.

->Created by Guido van Rossum, and released in 1991.

->simple syntax similar to the English language.

->runs on an interpreter system.

->Python can be treated in a procedural way, an object-oriented way or a functional way.

--------------------------------------------------------------------------------------

=> Can be used :

- on a server to create web applications.

* Python is Case sensitive.

--------------------------------------------------------------------------------------

XAMPP : means->

X->Cross-Platform

A->Apache

M->MariaDB

P->PHP

P->PERL

--------------------------------------------------------------------------------------

1. **Python Indentation**

Indentation refers to the spaces at the beginning of a code line.

Example:

if 5 > 2:

print("Five is greater than two!")

Syntax Error:

if 5 > 2:

print("Five is greater than two!")

--------------------------------------------------------------------------------------

1. **Python Variables**

Example

x = 5

y = "Hello, World!"

--------------------------------------------------------------------------------------

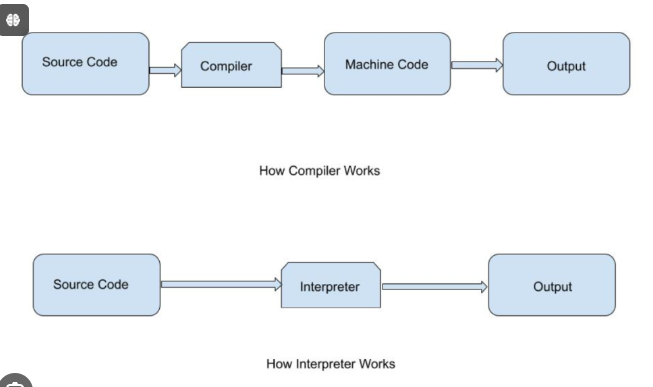
1. **Python Comments**

* comments start with '#' in python.
* There is NO syntax for Multiline Comments in python
* To add a multiline comment you could insert a # for each line.
* But , you can use a multiline string.

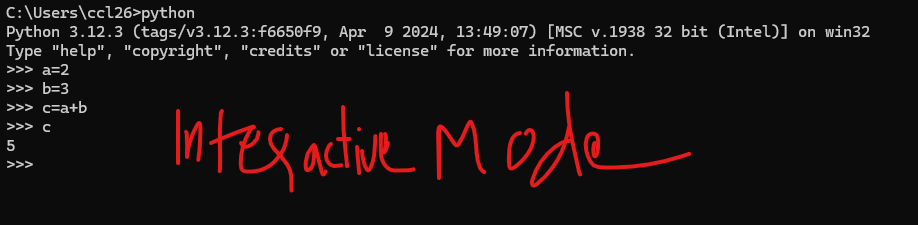
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

05/06/2024 - Wednesday

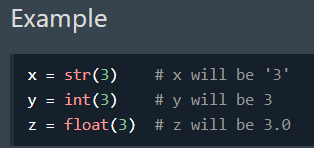
1. **Compiler vs Interpreter :**



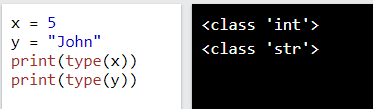
1. **Python interactive Mode :**



1. **Python Variables :**
   * **Casting :** If you want to specify the data type of a variable, this can be done with casting.



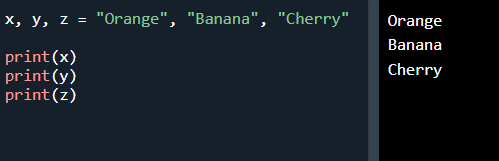
* + **Get the Type :** You can get the data type of a variable with the type() function.

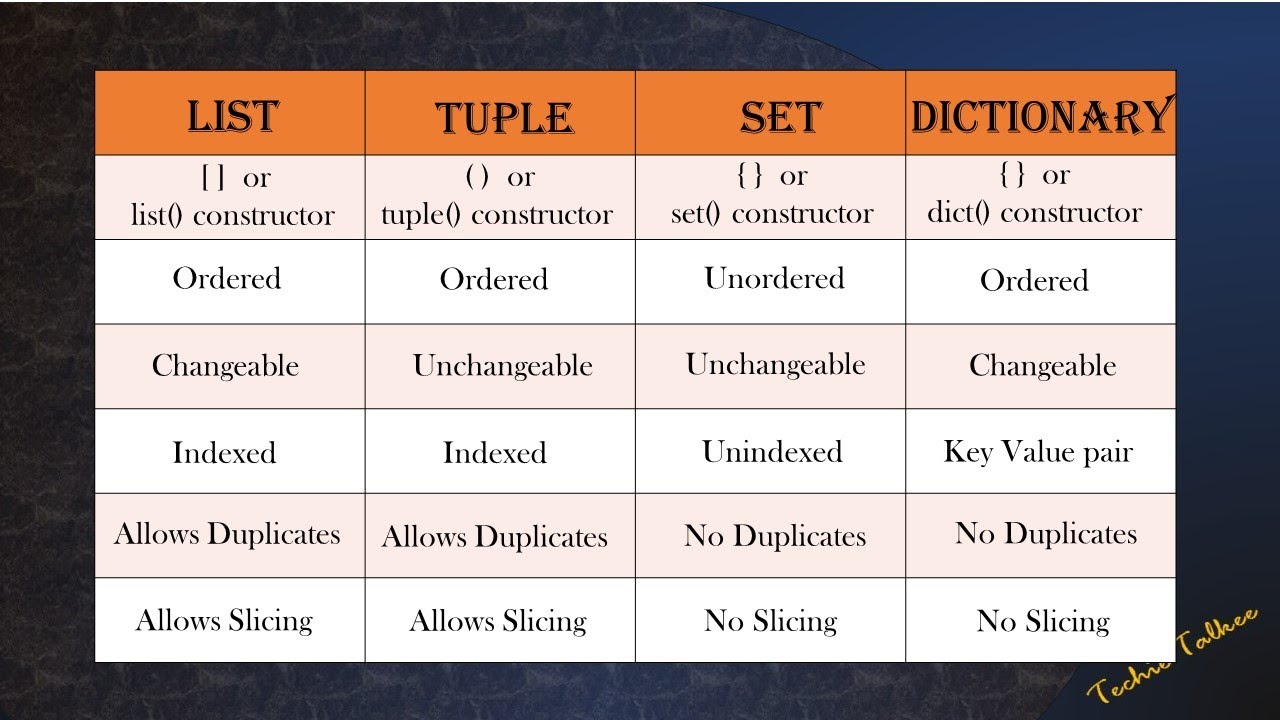
****

* + **Variable name :**

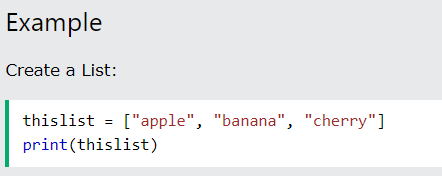
****

* + **Assign values to Mltiple variables :**

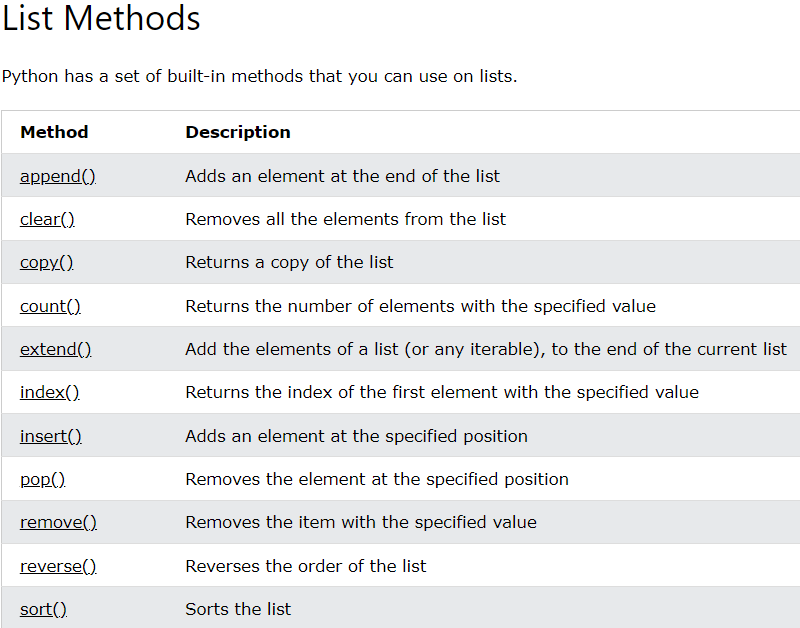
****

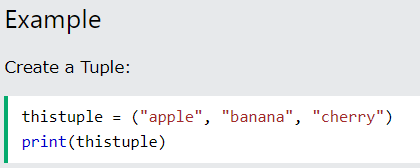
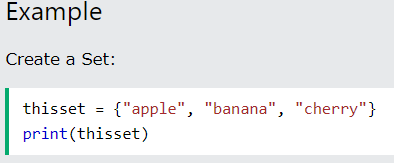
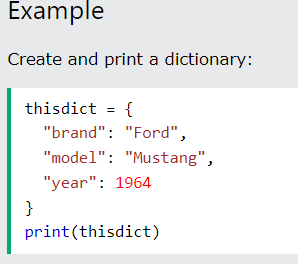
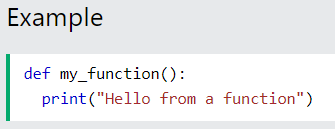
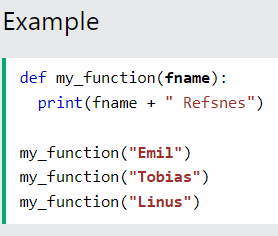


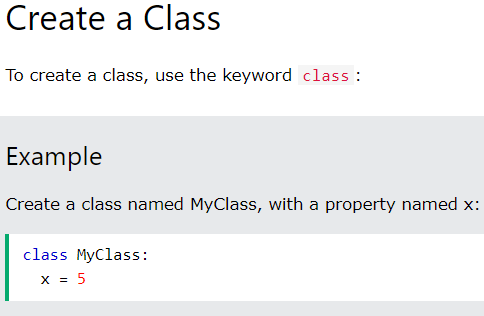
1. **Python List:**

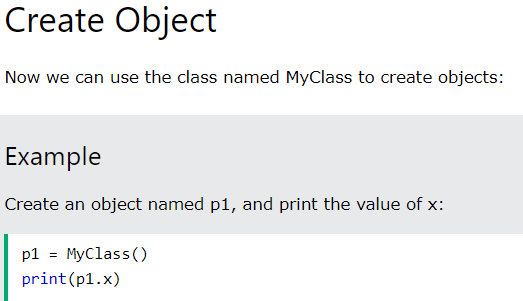
****

* + Lists are used to **store multiple items in a single variable**.
  + Lists are created using square brackets.
  + **List is :**
    - **Ordered,**
    - **Changeable,**
    - **Allow Duplicates**

****

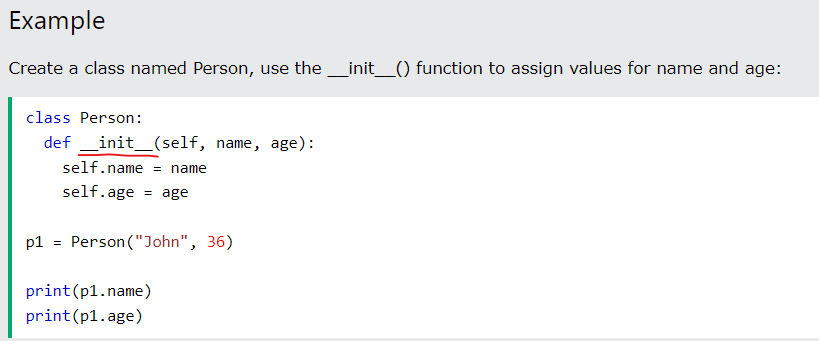
1. **Python Tuples:**
   * ****
   * Tuples are used **to store multiple items in a single variable**.
   * A tuple is a collection which is:
     + **ordered** .
     + **Unchangeable.**
     + **Allow Duplicates.**
2. **Python Sets:**
   * ****
   * Sets are used **to store multiple items in a single variable**.
   * A Sets is a collection which is:
     + **Unordered** .
     + **Unchangeable.**
     + **Duplicates Not Allowed.**
3. **Python Dictionaries :**
   * ****
   * Dictionaries are used **to store data values in key:value pairs**.
4. **Python Function :**
   * In Python a function is defined using the **def**keyword:
   * ****
   * ****
5. **Python Classes and Objects :**
   * Python is an object oriented programming language.
   * A Class is like an object constructor, or a "blueprint" for creating objects.

****

* + ****

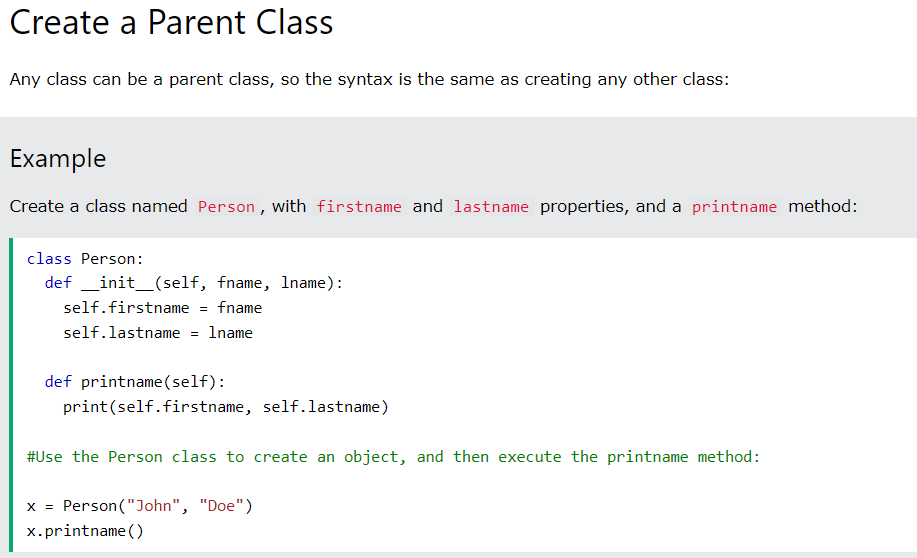
The \_\_init\_\_() Function

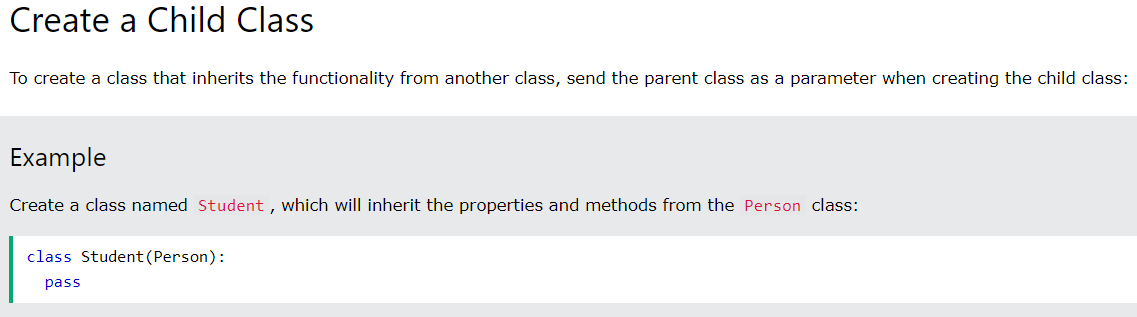
* + All classes have a function called \_\_init\_\_(), **which is always executed when the class is being initiated**.
  + It is used to assign values to object properties.

****

* + **Note:** The **\_\_init\_\_()**function is called automatically every time the class is being used to create a new object.And **self** is the first parameter.
  + **asd**

1. **Inheritance in Python:**
   * Inheritance allows us to define a class that **inherits all the methods and properties from another class**.
   * **Parent class** is the class being inherited from, also called base class.
   * **Child class** is the class that inherits from another class, also called derived class.





**Note:** Use the pass keyword when you do not want to add any other properties or methods to the class.

EXAMPLE:

#Parent Class :

class Person:

    def \_\_init\_\_(self, fname , lname) :

        self.firstname = fname

        self.lastname = lname

    def printName(self):

        print(self.firstname,self.lastname)

x = Person('Anshad' , 'Muhammad')

x.printName()

#Child Class :

class Student(Person):

    def \_\_init\_\_(self , fname,lname,age) :

        super().\_\_init\_\_(fname,lname) #super function called to pass values to parent class

        self.Age = age

    def printAll(self) :

        super().printName()

        print(self.Age)

y = Student('Nihal','Muhd',23)

y.printAll()

1. **Polymorphism in Python :**
   * The word "polymorphism" means "many forms", and in programming it refers to methods/functions/operators with the same name that can be executed on many objects or classes.
   * Example[Function Polymorphism] : ‘ len() ‘ function:=>
     + For strings, len() returns the **number of characters**.
     + For tuples len() returns the **number of items in the tuple**.
     + For dictionaries len() returns the **number of key/value pairs** in the dictionary.
   * Class Polymorphism :
     + In Class, we can have **multiple classes with the same method name**.

#Polymorphism :

class Car :

    def \_\_init\_\_(self, brand ,model) :

        self.brand = brand

        self.model = model

    def move(self) :

        print("Drive.")

class Boat :

    def \_\_init\_\_(self,brand , model):

        self.brand =  brand

        self.model = model

    def move(self):

        print("Sail")

class Plane :

    def \_\_init\_\_(self, brand , model):

        self.brand = brand

        self.model = model

    def move(self) :

        print("Fly.")

car1 = Car("Toyota","Fortuner")

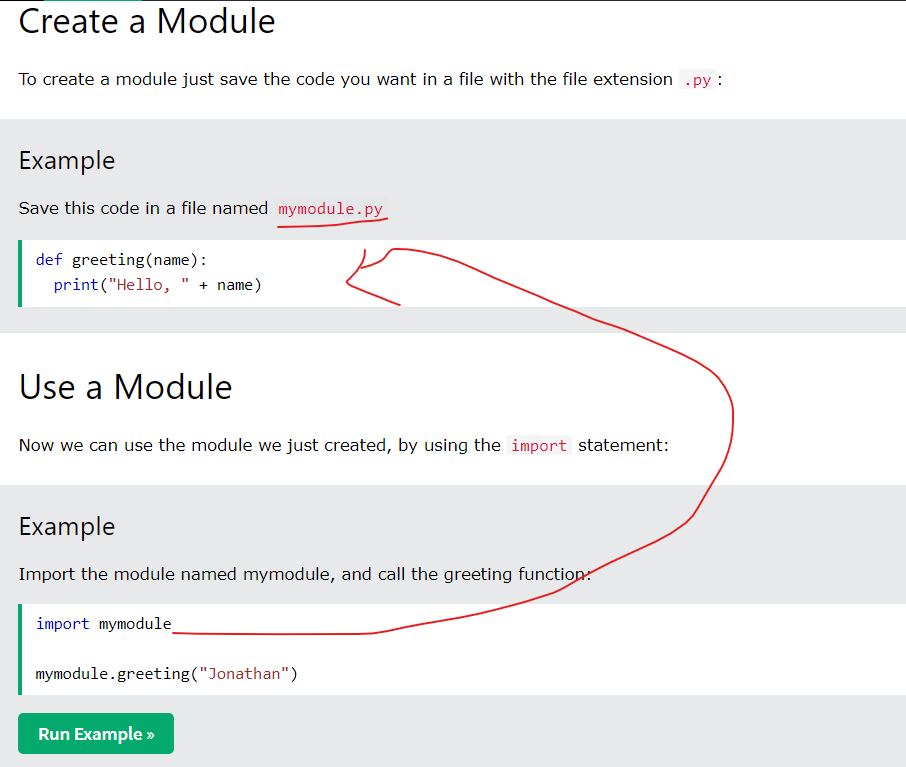
boat1 = Boat("Najad","Yachts")

plane1 = Plane("Lockhead\_Martin","747")

for x in (car1,boat1,plane1):

    x.move()

1. **Python Scope**
2. **Python Module:**

****

1. **Python Dates:**

import datetime

x = datetime.datetime.now()

print(x.year)

print(x.strftime("%A"))

1. **Python JSON :**

#-----Convet from JSON to Python------

import json

#JSON case:

x =  '{ "name":"John", "age":30, "city":"New York"}'

# parsing JSON which is in  x:

y = json.loads(x)

# As a result we get dictionary:

print(y["age"])

print(type(y))

#-----Convet from Python to Json-----------------------------

# a Python object (dict):

x = {

  "name": "John",

  "age": 30,

  "city": "New York"

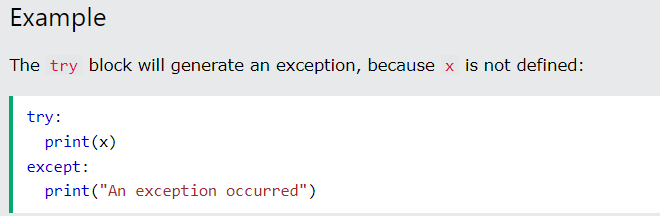
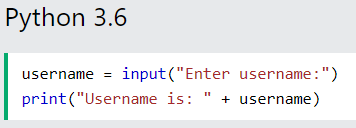
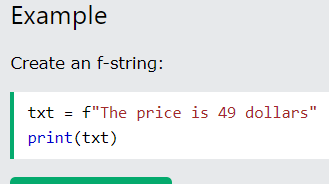
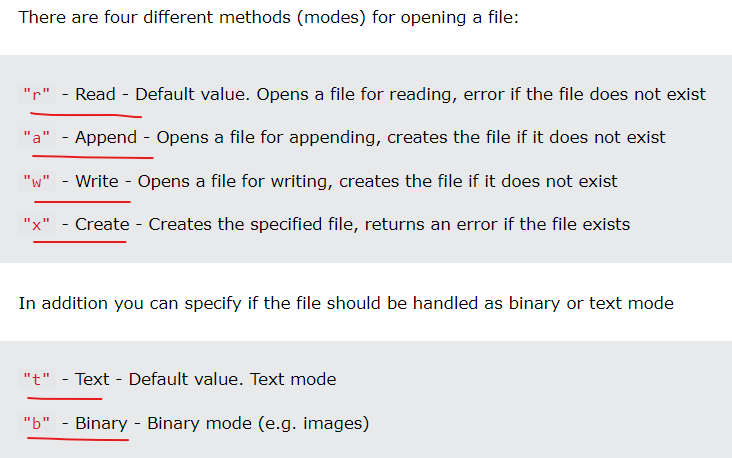
}

# convert into JSON:

y = json.dumps(x)

# the result is a JSON string:

print(y)

1. **Python PIP :**
   * PIP is a package manager for Python packages.
2. **Python Try Except [Exception Handling]:**
   * The try block lets you test a block of code for errors.
   * The except block lets you handle the error.
   * The else block lets you execute code when there is no error.
   * The finally block lets you execute code, regardless of the result of the try- and except blocks.
   * ****
3. **Python Error :**
   * Python 3.6 uses the input() method.
   * Python 2.7 uses the raw\_input() method.
   * ****
4. **Python String Formatting[f-string]:**
   * F-string allows you to format selected parts of a string.
   * To specify a string as an f-string, simply put an f in front of the string literal, like this:
   * ****
5. **File Handling :**
   * he key function for working with files in Python is the **open()** function.
   * ****

#File handling:

#To create a file

f = open("file1.txt","x")

#To Write into a file

f2 = open("file2.txt","w")

f2.write("Hello from code")

#To read a file

f3 = open("file1.txt","r")

print(f3.read())

#To append to a File

f4 = open("file1.txt","a")

f4.write("Appending to FIle1")

#To delete a file :

import os

os.remove("filedel.txt")