# Question Bank of Programming In Python for CAT II

# [notes](https://github.com/neeraj46665/BTech-CS-Notes/tree/main/BTech-2nd-year/sem-4)

1. Define Python Pandas?

Pandas is a Python library used for working with data sets.

It has functions for analyzing, cleaning, exploring, and manipulating data.

The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis"

Pandas allows us to analyze big data and make conclusions based on statistical theories.

Pandas can clean messy data sets, and make them readable and relevant.

Relevant data is very important in data science.

Pandas gives you answers about the data. Like:

* Is there a correlation between two or more columns?
* What is average value?
* Max value?
* Min value?

Pandas are also able to delete rows that are not relevant, or contains wrong values, like empty or NULL values. This is called *cleaning* the data.

1. Mention different types of Data Structures in Panda?

Pandas, a data analysis library, supports two data structures:

* Series: one-dimensional labeled arrays *pd.Series(data)*
* DataFrames: two-dimensional data structure with columns, much like a table.

## **Series**

A series can be seen as a one-dimensional array. The data structure can hold any data type, that is includings strings, integers, floats and Python objects.

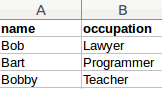
A very basic example is shown below, where it holds characters:

|  |
| --- |
| import pandas as pd s = pd.Series(['a','b','c']) |

## **Data Frames**

The data frame datastructure is similar to a table. Data Frames are the most commonly used Pandas data structures. So how is it made?

Lets say you have the following table:

[](https://pythonspot.com/wp-content/uploads/2016/08/panda-table.png)Example Dataframe

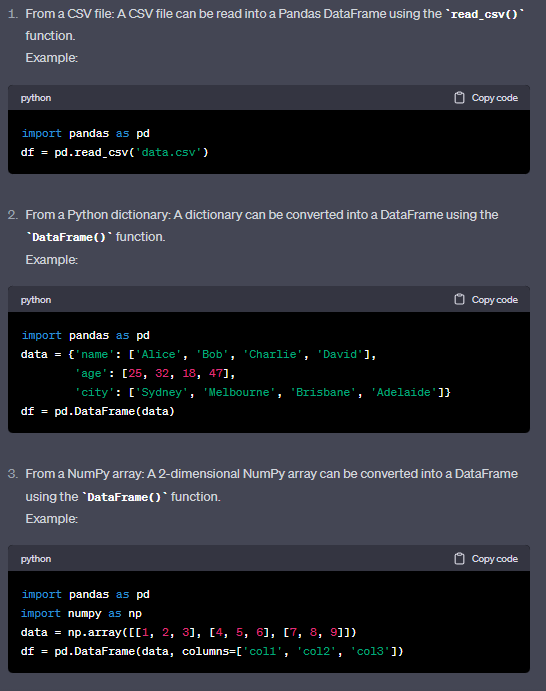
Then you can represent that as a Python dictionary like this:

|  |
| --- |
| d = { 'name': ['Bob','Bart','Bobby'],  'occupation': ['Lawyer','Programmer','Teacher']} |

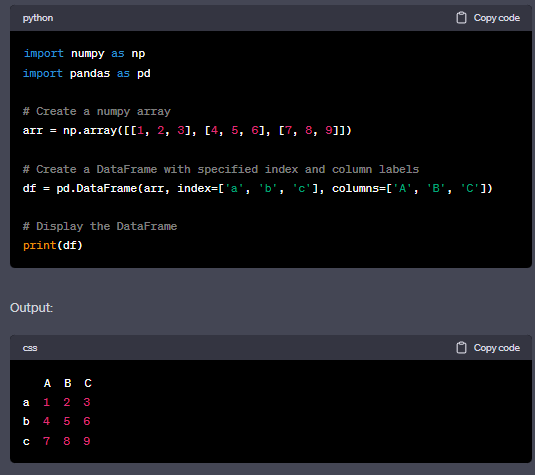
After that, you can create a new DataFrame object.

|  |
| --- |
| frame = pd.DataFrame(d, columns=['name','occupation']) |

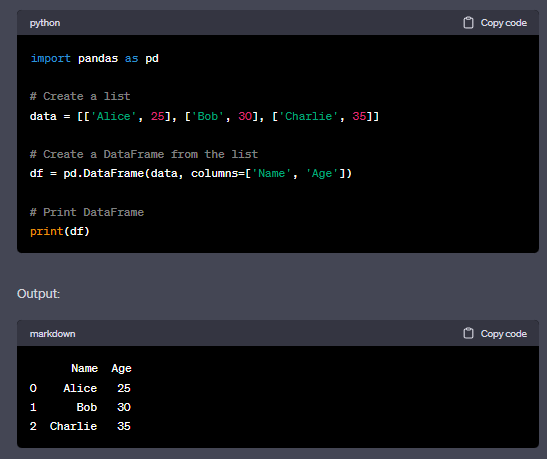
1. Explain different ways of creating Data Frames in Panda?

### 2- Taking index and columns from the array

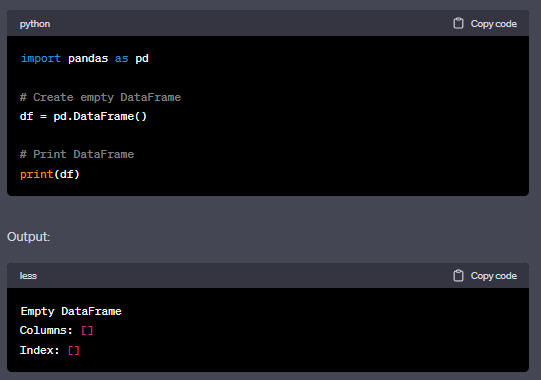


## 3- Creating DataFrame from a Python List

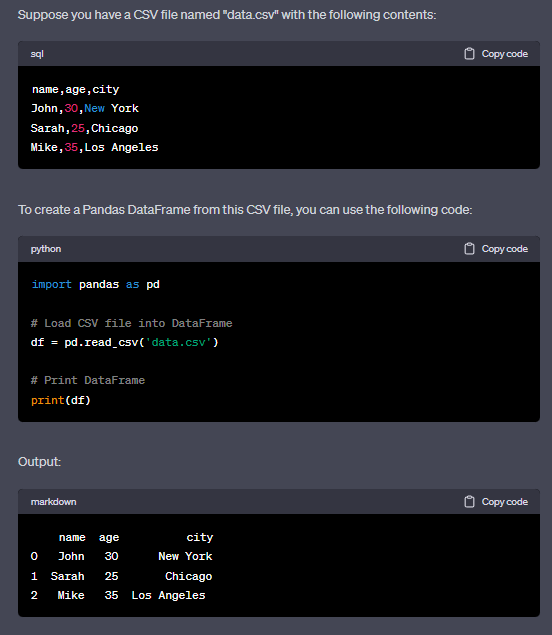


## 4- Creating DataFrame from Python a dictionary object

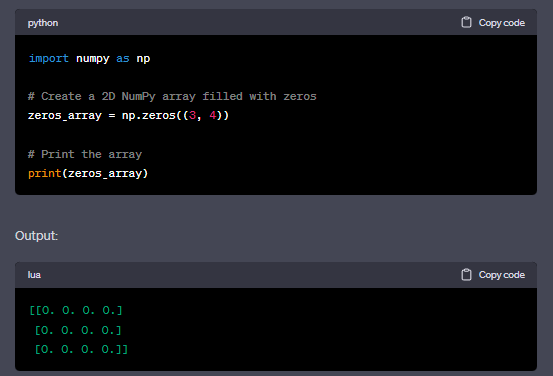
When you pass a Python dictionary object to the DataFrame constructor function, the resulting dataframe will have the keys in the dictionary as column names.



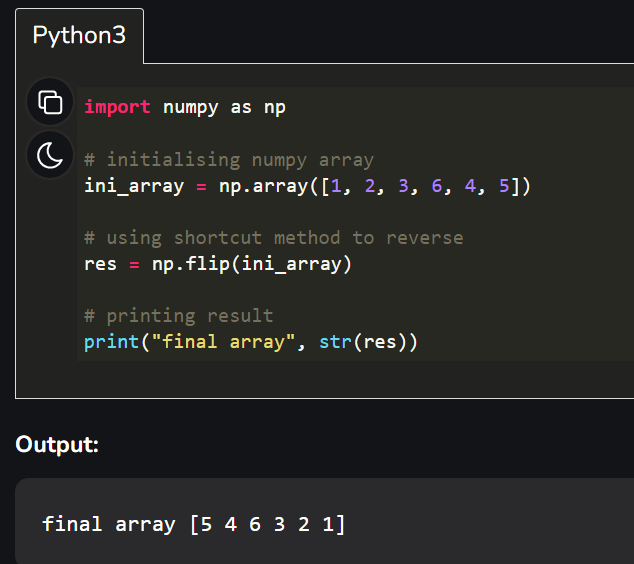
## 6- Creating DataFrame from a CSV file



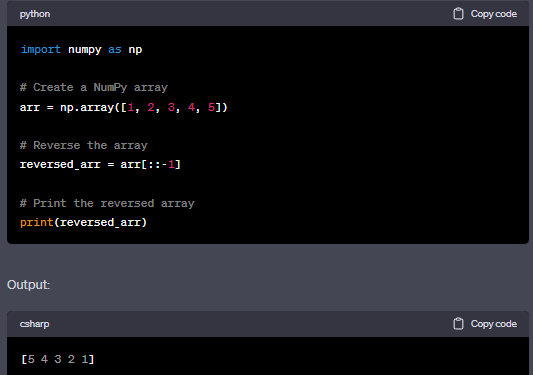
1. [Build a Numpy array filled with all zeros](https://www.geeksforgeeks.org/create-a-numpy-array-filled-with-all-zeros-python/).



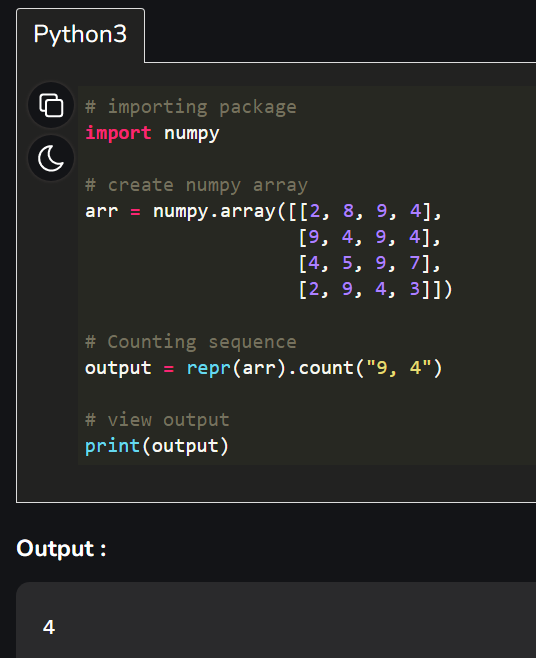
1. [Reverse a Numpy array.](https://www.geeksforgeeks.org/python-reverse-a-numpy-array/)



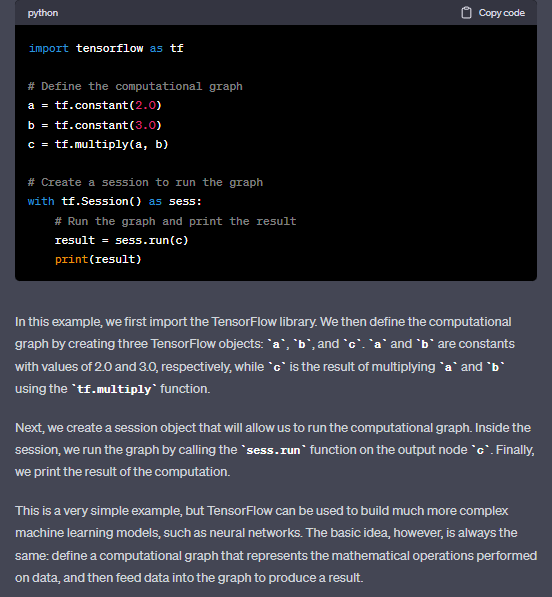
or



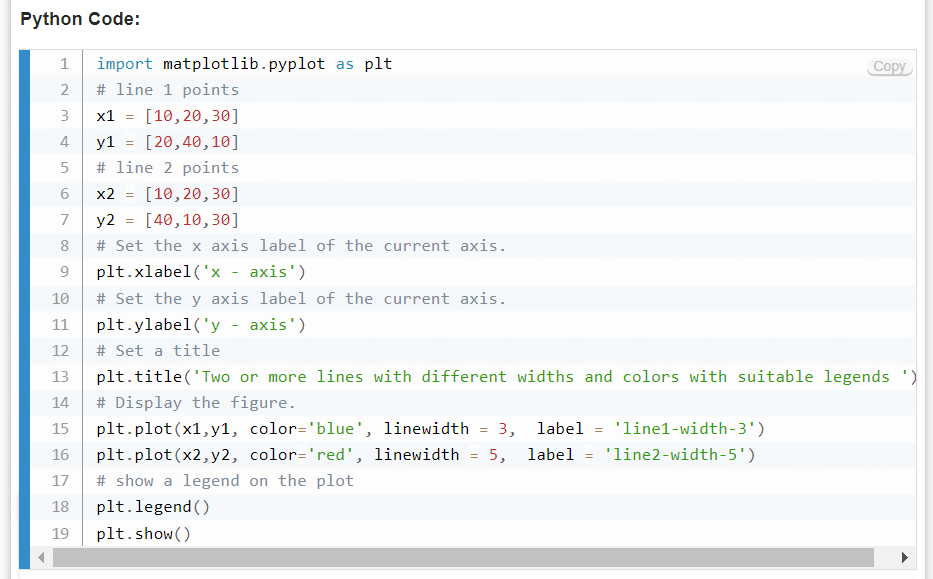
1. [Find the number of occurrences of a sequence in a NumPy array.](https://www.geeksforgeeks.org/find-the-number-of-occurrences-of-a-sequence-in-a-numpy-array/)

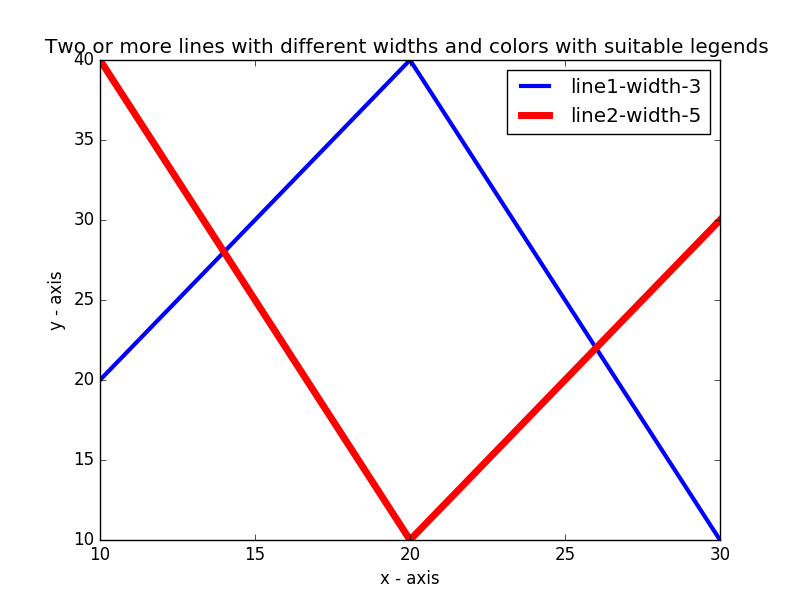


1. [Analyse the simple working of an algorithm in Tensor Flow?](https://intellipaat.com/blog/interview-question/tensorflow-interview-questions/#8)

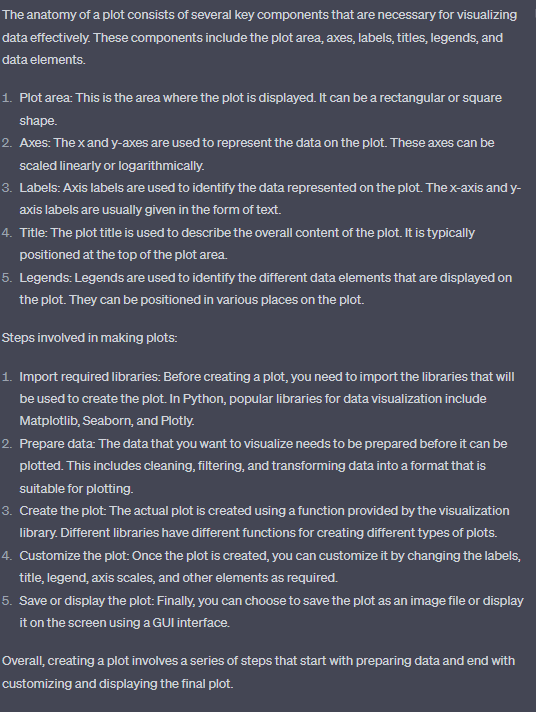


1. Describe steps involved in making plots. Explain plotting two or more lines on the same plot with an example.ans9-10
2. Develop a Python program to plot two or more lines with legends, different widths and colours.

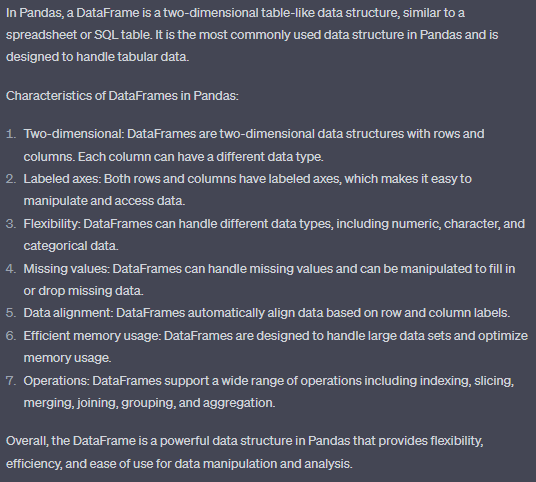




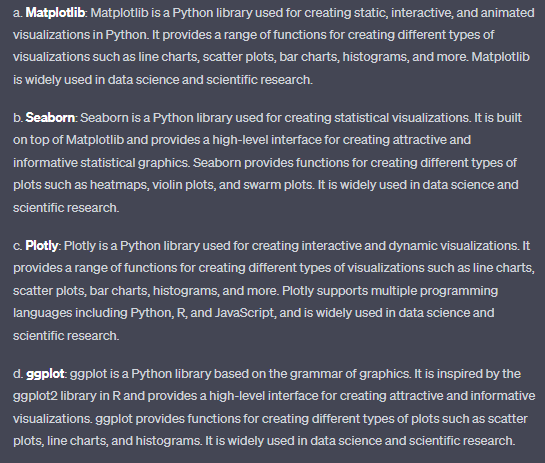
1. Describe anatomy of a plot. Explain steps involved in making plots.



1. Characterize the Data Frames in Pandas?



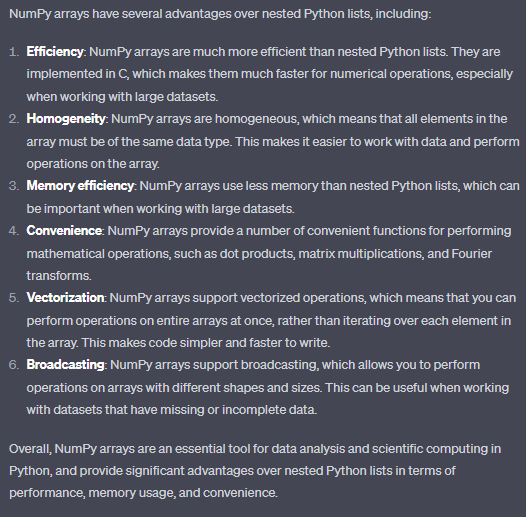
1. Explain the following:
   1. Matplotlib
   2. Seaborn
   3. Plotly
   4. ggplot



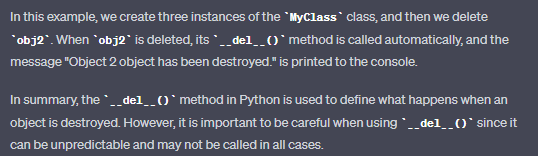
1. Explain the Applications of SciPy, Scrapy, Scikit-learn, PyGame, PyTorch, PyBrain and Keras.



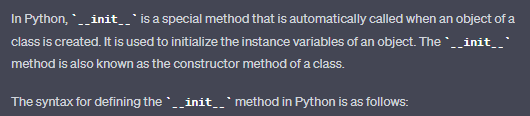
1. List the advantages NumPy Arrays have over (nested) Python lists?

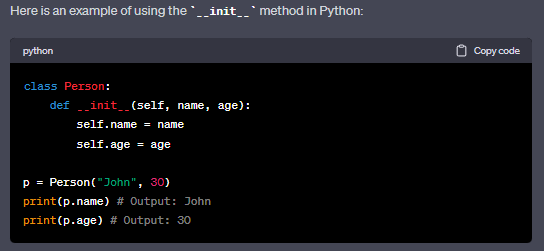


1. Briefly explain the use of finalise method in python.

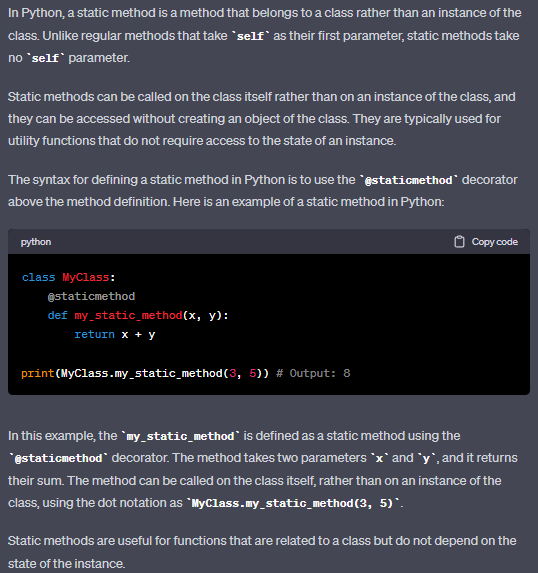
1. Explain the use of init function in python.



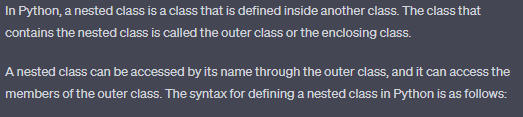


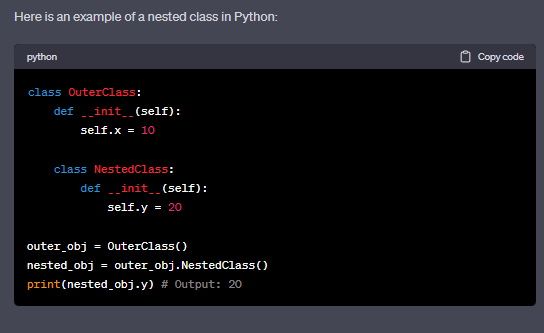


1. Explain Python's static methods.

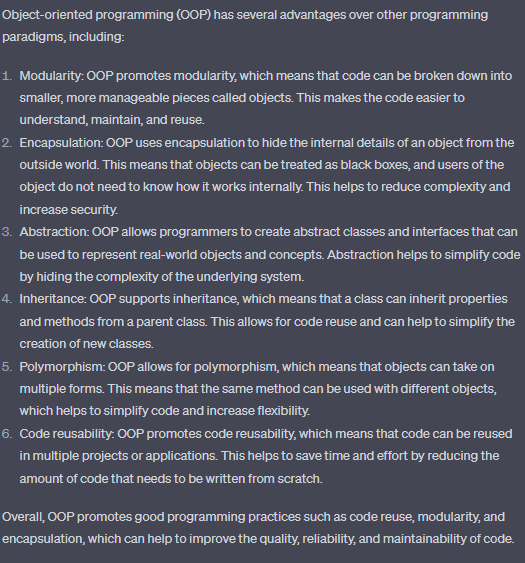


1. Explain Python's Nested Class.





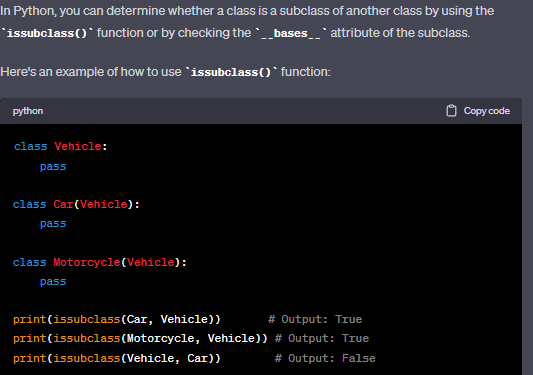
1. List the advantages of using OOPs.

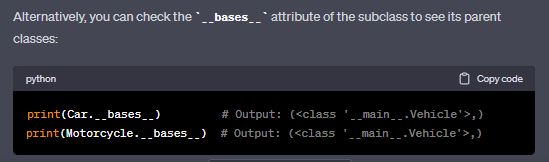


1. Explain access specifiers in python.

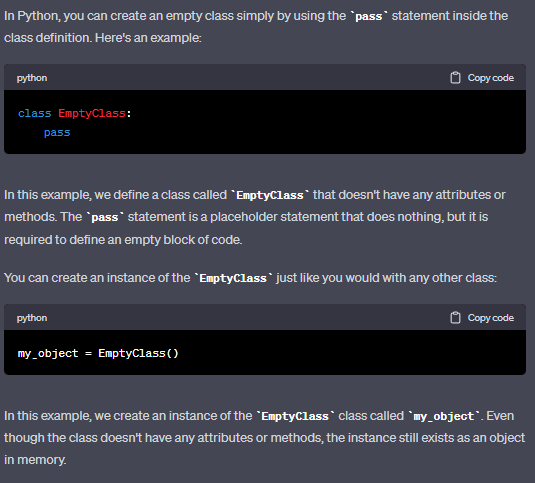


1. Can a parent class be called without first creating an instance of it? Explain. ans-25
2. How can you determine whether a class is a subclass of another class?

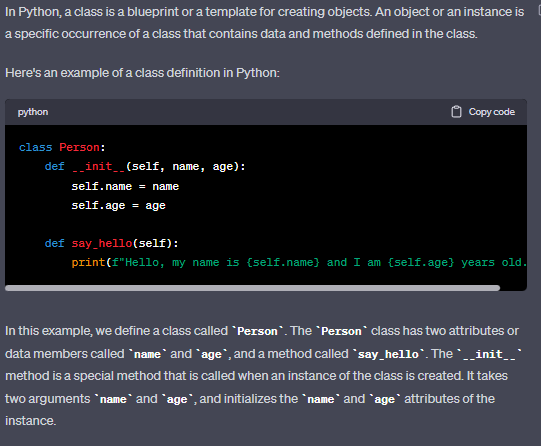


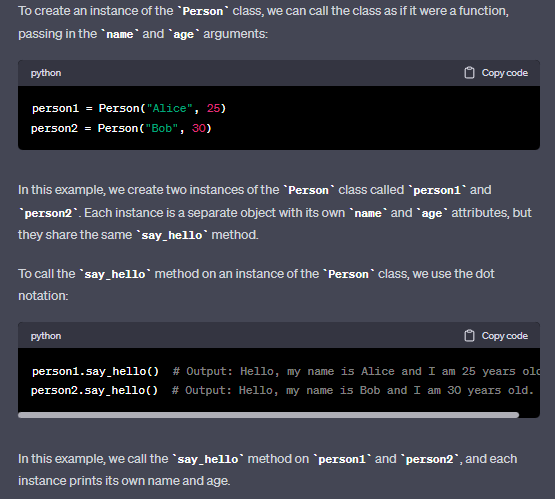


1. How do you make a Python class that is empty?

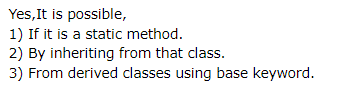


1. Explain a class object or instance in python.





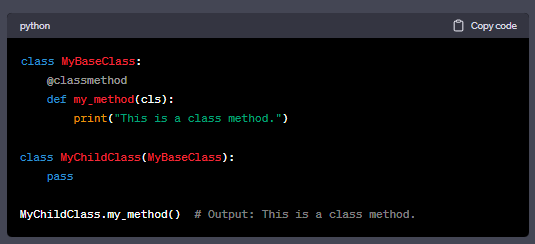
1. Can you call the base class method without creating an instance? Explain.



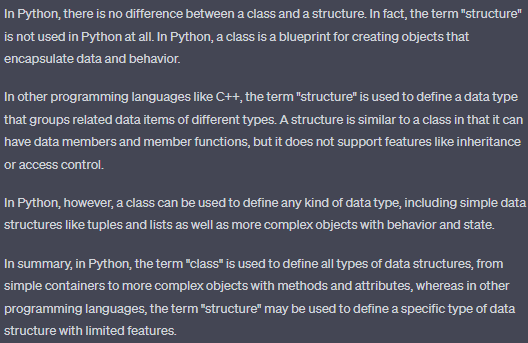
1)from static

class Music:  
 @staticmethod  
 def play():  
 print("\*playing music\*")  
  
Music.play()

2)

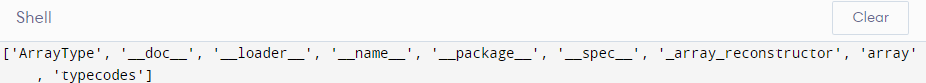


1. What is the difference between a class and a structure?

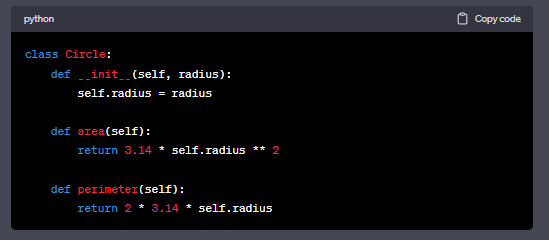


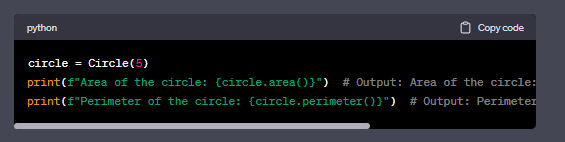
1. Write a Python program to import a built-in array module and display the namespace of the said module.



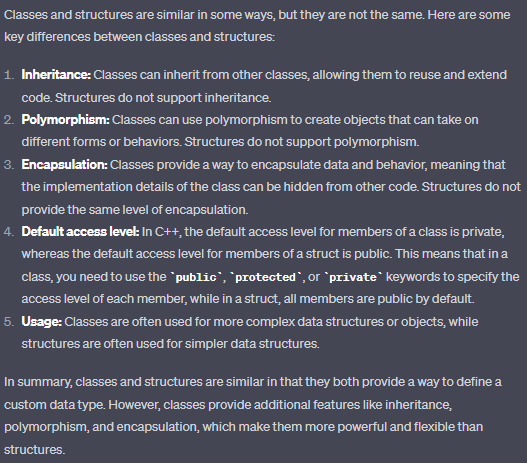


1. Write a Python class named Circle constructed from a radius and two methods that will compute the area and the perimeter of a circle.

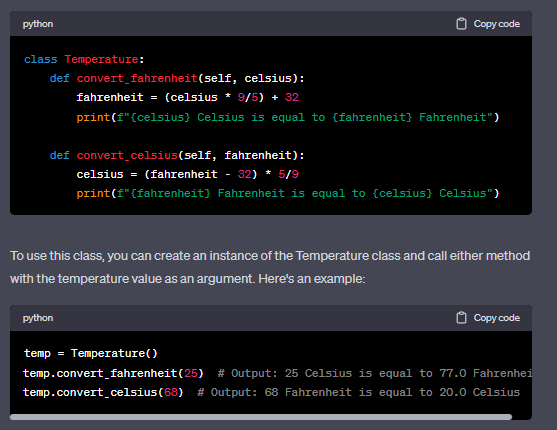




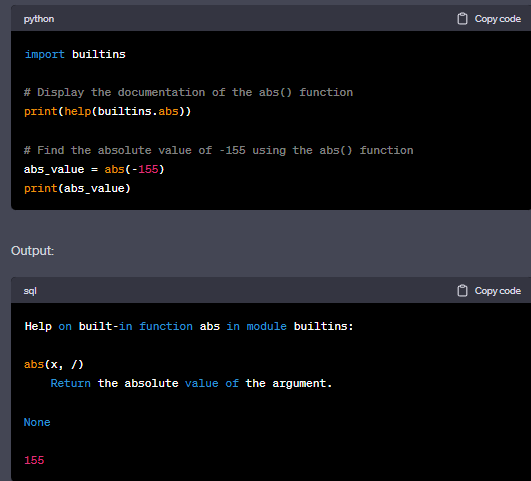
1. Are class and structure the same? If not, what's the difference between a class and a structure?



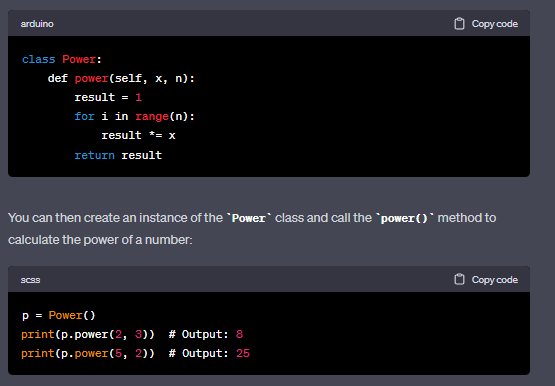
1. Create a Temperature class. Make two methods:
2. convert Fahrenheit - It will take Celsius and will print it into Fahrenheit.
3. convert Celsius - It will take Fahrenheit and will convert it into Celsius**.**

****

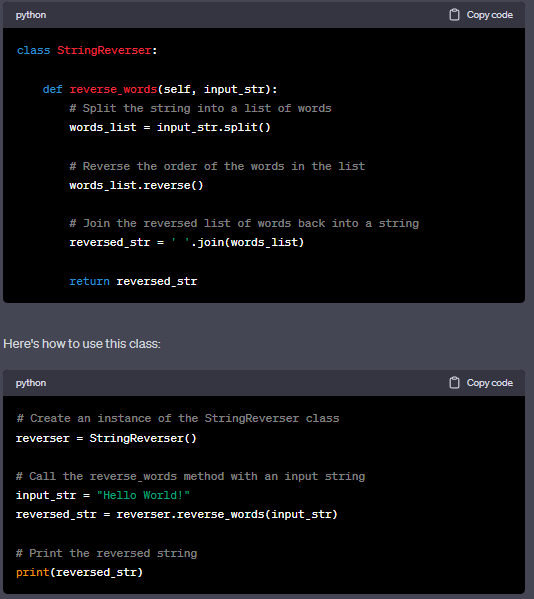
1. Write a Python program that imports the abs() function using the built-ins module, displays the documentation of the abs() function and finds the absolute value of -155.



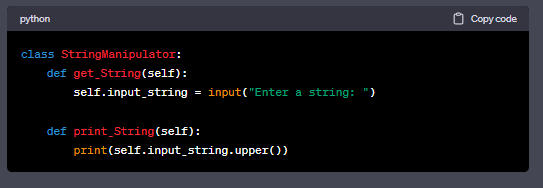
1. Write a Python class to implement pow(x, n).

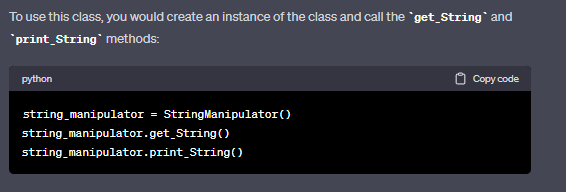


1. Write a Python class to reverse a string word by word.



1. Write a Python class that has two methods: get\_String and print\_String , get\_String accept a string from the user and print\_String prints the string in upper case.





1. Illustrate the difference between:
2. read( ) and readlines ( )
3. write( ) and writelines( )
4. r+ file mode and rb+ mode.
5. w‟ and „a‟ modes

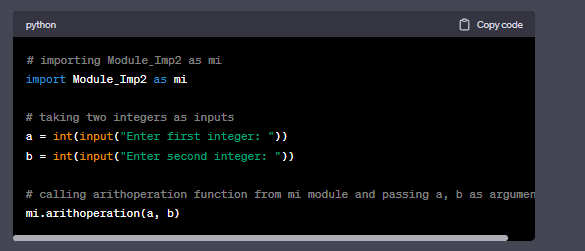
a) The **read()** method reads the whole content of a file as a string, while the **readlines()** method reads the file line by line and returns a list of strings, each string representing a line.

b) The **write()** method writes a string to a file, while the **writelines()** method writes a list of strings to a file.

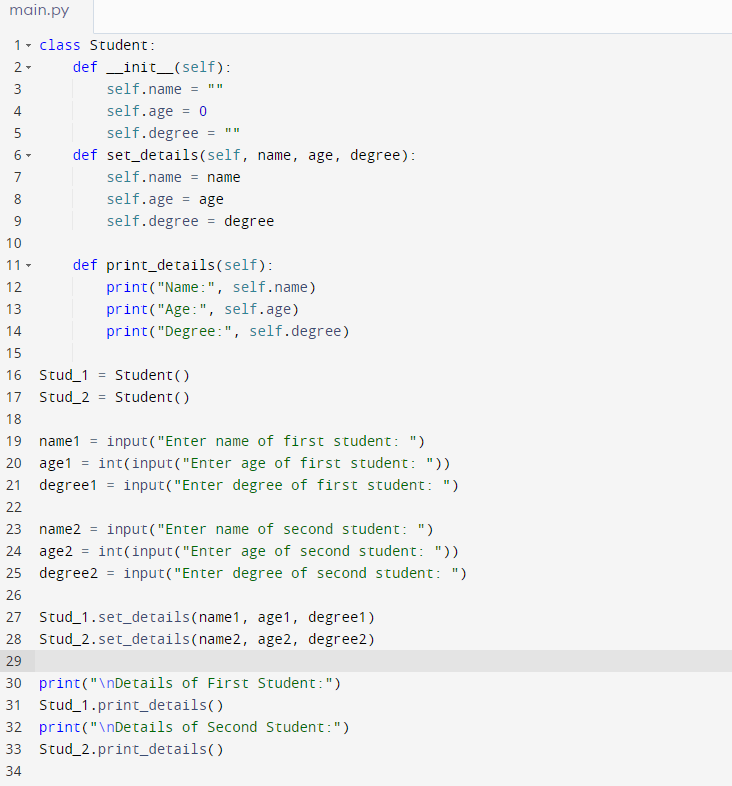
c) **r+** mode opens a file for both reading and writing. If the file does not exist, an error is raised. **rb+** mode opens a binary file for both reading and writing. If the file does not exist, an error is raised.

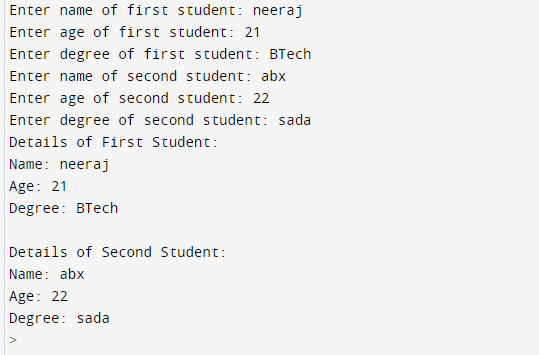
d) **w** mode opens a file for writing and truncates the file to zero length if it already exists. If the file does not exist, it creates a new file. **a** mode opens a file for appending. The file pointer is positioned at the end of the file. If the file does not exist, it creates a new file.

1. Apply the below instructions when writing the program.
2. Import Module\_Imp2 as mi
3. Take two integers a and b as inputs from the user
4. Call the function mi.arithoperation by passing a, b.



1. Demonstrate the use of Class while writing a code following the given instructions to add the details of the two students by taking the inputs from the user.
2. Create a class Student.
3. Create an instance Stud\_1 of class Student.
4. Create another instance Stud\_2 of class Student.
5. Take name, age, and degree of the student as inputs from the user.
6. Print the details of the student.





1. f= open(“data.txt”)

Make use of the code given above and write the answers of the following:

1. Identify name of the file.
2. What is „f‟ in above code?
3. What is the mode of operation in the above file?
4. Discuss the different types of close ().

a) The name of the file is "data.txt".

b) "f" is a file object created by opening the "data.txt" file in read mode.

c) The mode of operation in the above file is read mode, which is the default mode if no mode is specified explicitly.

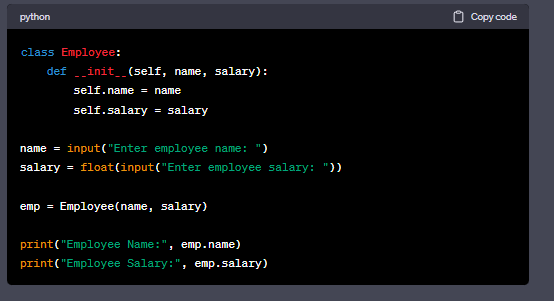
d) The different types of close() method are:

close(): This method closes the file and releases any system resources associated with it. It is the most commonly used form of close() method.

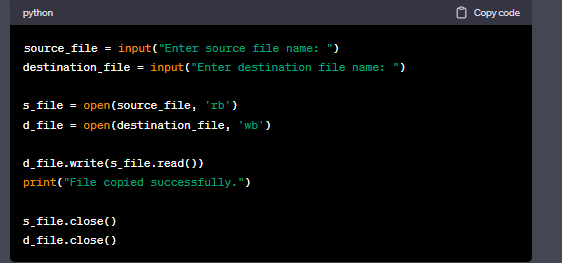
flush(): This method flushes the write buffers of the file. It is used when we want to make sure that all the data written to the file is actually stored on the disk, rather than being buffered in memory.

closed: This attribute returns a boolean value that indicates whether the file is closed or not. If the file is closed, it returns True, otherwise, it returns False.

1. Build a class Employee, which contains the details of an employee like name and salary. Take name and salary as inputs from the console, print the result.



1. Assume a filename and write a Python program to **copy** one file to another file in file handling. Explain the tell() and seek() in python file handling.



Now, regarding the tell() and seek() methods in file handling:

* tell() method returns the current position of the file pointer in bytes, i.e., the byte offset from the beginning of the file.
* seek(offset, whence) method changes the position of the file pointer to a specified byte offset, relative to a given whence parameter. The whence parameter can take the values 0 (seek from the beginning of the file), 1 (seek from the current position), or 2 (seek from the end of the file). For example, f.seek(0, 0) moves the file pointer to the beginning of the file, while f.seek(10, 1) moves the file pointer 10 bytes ahead of the current position.

1. Define a function checkNegativeNumber which has an argument num. Write a program to check the given num is a **positive** or **negative**. Take the input number from the user, and print the result as shown in the examples.

# Sample Input and Output 1:

a: 20

positive

**Sample Input and Output 2:**

a: -90

negative



1. The below program **Module\_Imp2** is already written.

* This file is same as **Module1** written earlier, which takes two parameters and does arithmetic operations on these two, and print the result.
* Import it in your **Module4.py**.
* Take two integers a and b as input from the user.
* Call the function **arithoperations()** of **Module\_Imp2** module by passing the arguments a and b.

# Sample Input and Output:

a: 12

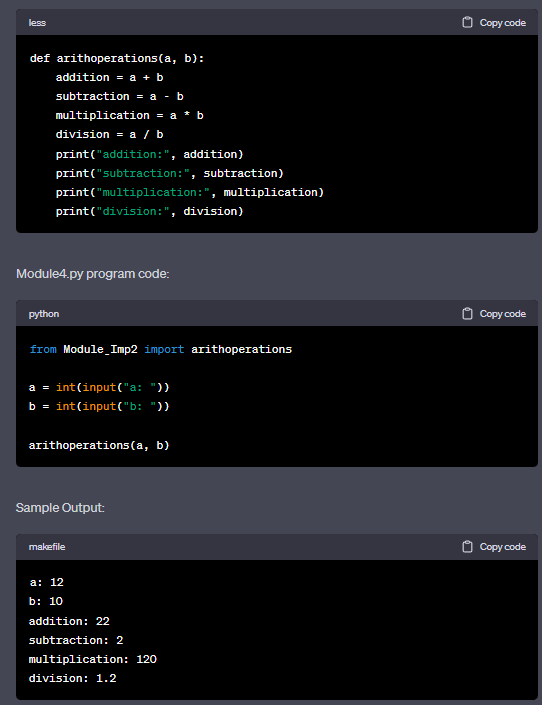
b: 10

addition: 22

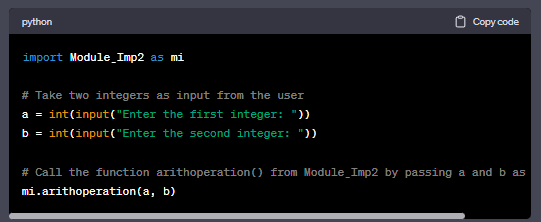
subtraction: 2

multiplication: 120

division: 1.2

****

1. Apply the below instructions when writing the program.
2. Import Module\_Imp2 as mi
3. Take two integers a and b as inputs from the user.
4. Call the function mi.arithoperation by passing a, b.



1. Inspect whether class and structure are the same? If not, what's the difference between a class and a structure?

In Python, there is no difference between a class and a structure as both are created using the class keyword.

However, in some other programming languages such as C++, C#, and Java, there is a difference between a class and a structure. The main difference is that a structure is a value type while a class is a reference type.

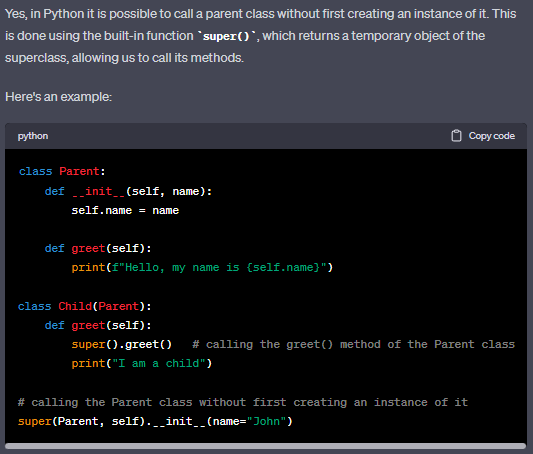
In C++, a structure is a collection of related data items that may have different data types. By default, the members of a structure are public. A class is also a collection of related data items, but the members of a class are private by default.

In C#, a structure is similar to a class, but it is a value type rather than a reference type. A structure can contain fields, methods, properties, and events, just like a class. However, a structure cannot inherit from other structures or classes.

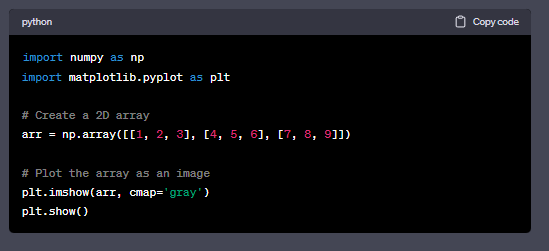
In Java, there is no concept of a structure. However, Java has a similar concept called a struct that is used in native code. A struct is a data structure that groups together variables of different data types under a single name.

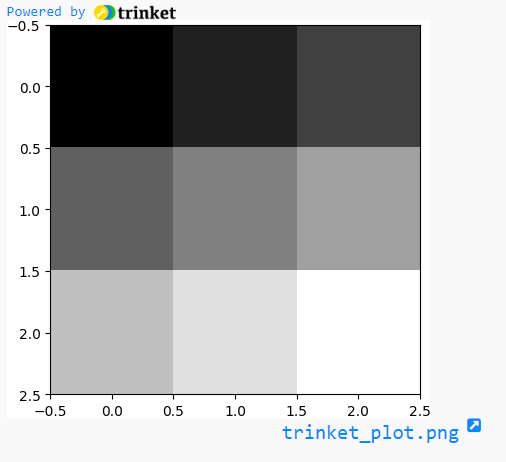
In summary, while the concept of a class and a structure are similar in Python, in other programming languages there are differences in how they are implemented and used.

1. Inspect whether a parent class be called without first creating an instance of it? Explain.



1. Build a two 2-D array. Plot it using matplotlib.



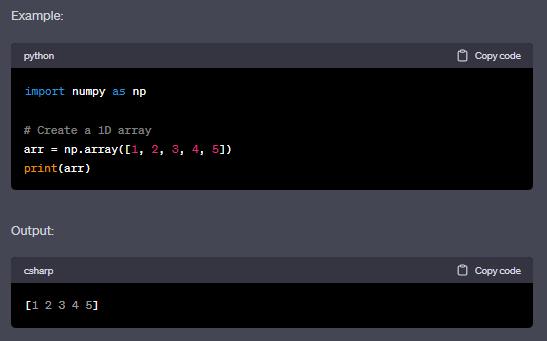


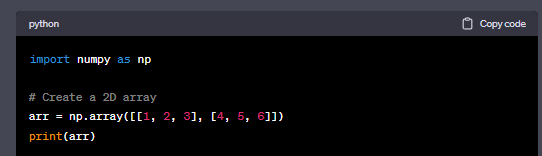
47.Analyse the steps to create a 1D array and 2D array.

To create a 1D array in Python, we can use the NumPy library. Here are the steps to create a 1D array:

Import the NumPy library using the import numpy as np statement.

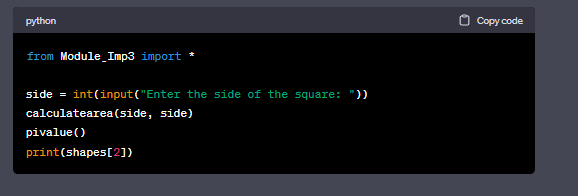
Create an array using the np.array() function and passing a list or tuple of values as its argument.



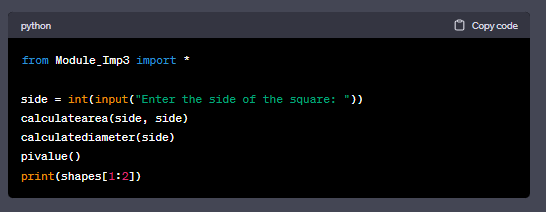


48.Write a program that uses \* in import.

* from Module\_Imp3 import \*
* Take integer as input from user and store it in the variable **side.**
* Call the function **calculatearea**(side, side)
* Call the function pivalue()
* print the third element in shapes.
* Note: The Module\_Imp3.py already is written.



1. Follow the given instructions while writing the program
   * Use the **Module\_Imp3** which contains functions that can be imported.
   * Use from Module\_Imp3 import \*
   * Take an integer as **input** from user and store it in the variable **side.**
   * Call the function calculatearea(side,side)
   * Call the function calculatediameter(side)
   * Call the function pivalue()
   * print shapes[1:2]



1. Write a simple program followed by the instructions given below:
   * A base class **Person** and a derived class **Student** with **Person** as its base class.
   * Add two methods **setname()** (which takes the parameter self and name)and **getname()** which prints the name in the base class.
   * Add two methods in the derived class: **setage()** (which takes the parameters self and age) which sets the age and **getage()** which prints the age.
   * Create an instance of **Student** and name it as **s1**.
   * Take **name** and **age** as inputs from the console.
   * Call the **setname()** and **setage()** on this instance by passing the **name** and **age** parameters.
   * Call the **getname()** and **getage()** on this class, which prints the passed parameters

