

DEPARTMENT OF MATHEMATICS
(Under DBT Star College Scheme)

LABORATORY MANUAL



PYTHON PROGRAMMING

23CMAT2P



G.T.N. ARTS COLLEGE (AUTONOMOUS)

A Telugu Linguistic Minority Institution | Affiliated to Madurai Kamaraj University
Re-accredited (Cycle - II) by NAAC with 'A++' Grade
Recognized under DBT STAR COLLEGE SCHEME | Supported by DST - FIST Program
Listed in NIRF- 2024 Ranking
Old Karur Road, Dindigul – 624 005.

Sl. No.	Program	Page No.
	Introduction	1
	Syntax	9
1.	Print pyramid pattern of given numbers.	10
2.	Find the roots a quadratic equation.	11
3.	Find the factorial of the given number using function	12
4.	Compute Matrix Addition and Multiplication.	13
5.	Find the largest number in a list using function.	14
6.	Find the area of shapes using function	15
7.	Find reverse string, string palindrome, character count and replacing string..	17
8.	Create EB bill..	18
9.	Swap two variables.	19
10.	Draw a circle of square using turtle	20

Introduction to Python

Python is a high-level, interpreted, and general-purpose programming language developed by Guido van Rossum in 1991. It is known for its simplicity, readability, and flexibility, making it one of the most popular languages for beginners as well as professionals. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming. One of Python's greatest strengths is its vast standard library and a large collection of third-party packages, which make it suitable for a wide range of applications — from web development and data analysis to artificial intelligence (AI), machine learning (ML), scientific computing, and automation.

Python can run on different operating systems like Windows, macOS, and Linux without modification. The syntax of Python is designed to be clean and easy to understand, emphasizing code readability and reducing the cost of program maintenance. Because of its efficiency and versatility, Python has become the foundation for many modern technologies and research areas, making it a vital tool in both academic and industrial fields.

What does Python do?

Python includes hundreds of mathematical functions. It has a high-level, interpreted, and general-purpose programming language. A large number of functionalities is included in Python such as

- **Web Development:** Used to build websites and web applications.
- **Data Analysis and Visualization:** Helps analyze, process, and visualize data easily. Uses Libraries such as *Pandas*, *NumPy*, *Matplotlib*, *Seaborn*.
- **Automation (Scripting):** Automates repetitive tasks like file handling, email sending, or data entry.
- **Scientific and Numerical Computing :** Used for simulations, complex calculations, and research applications. Uses Libraries as *SciPy*, *SymPy*.
- **Cybersecurity and Ethical Hacking :** Helps in writing scripts for penetration testing and network monitoring.
- **Internet of Things (IoT):** Used in controlling and connecting smart devices.

Available methods:

- Linear algebra, Matrices
- Polynomials and rational functions
- Interpolation and regression
- Differentiation and integration
- Linear systems of equations
- Ordinary Differential Equation solver and Differential Algebraic Equations solver

Version: PYTHON 3.11

Released on October 24, 2022

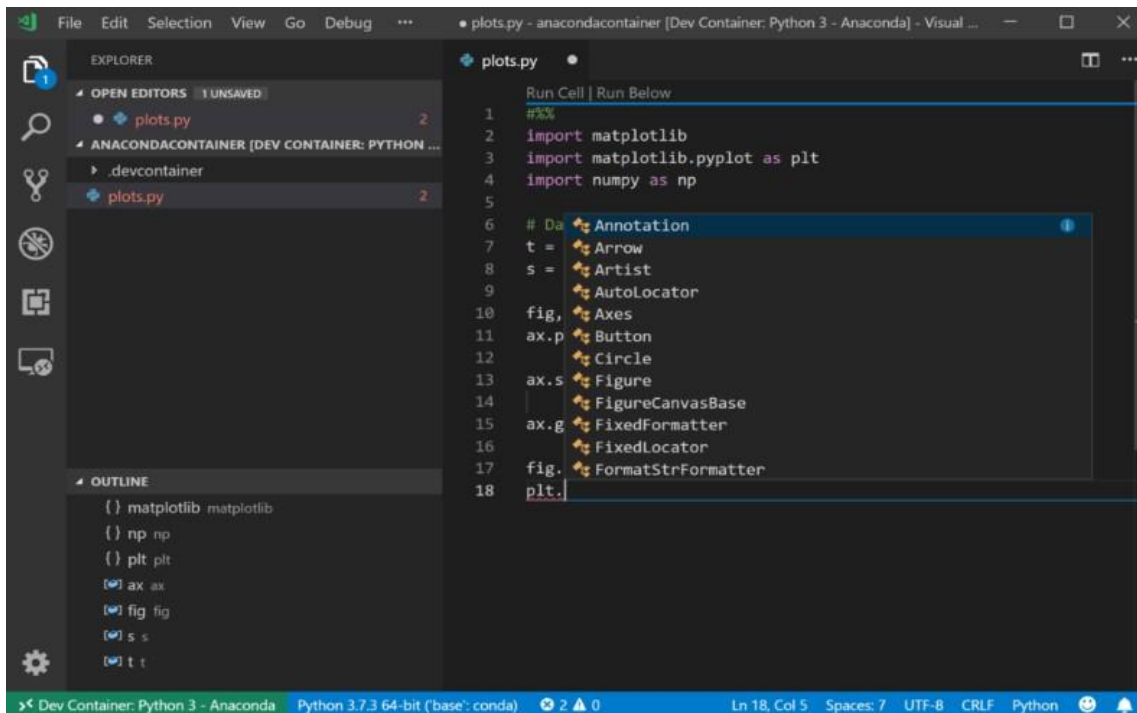
Windows 10, 11 (64-bit)

Steps to Install Python 3.11 on Windows

1. Go to python.org/downloads.
2. Click on “**Download Python 3.11.x**” (the latest sub-version).
3. Run the installer file.
4. **Tick “Add Python to PATH”** before clicking “Install Now.”
5. Once installed, open **Command Prompt** and type: `python --version`

Getting started with Python:

Installation of the software is easy like any other software. The software gets ready after its installation into a computer system. Just like any other software, Python can be opened by Double clicking on the Python icon in desktop or by clicking on the icon appearing after the entering the name Python in the Windows search bar.



PYTHON WORKSPACE

There are two workspaces for Python programming: The Python Editor and the Python Shell.

Python Editor	Python Shell
• Create, edit and save Python programs • Syntax highlighting and auto-indentation • Inline prompts to guide with function arguments • Tooltips to show range of valid values • key lists global user variables and functions defined in the current program • Keypad shortcuts	• Run Python programs • Convenient for testing small code fragments • Interaction with Shell history to select previous inputs and outputs for re-use • key lists global user variables defined in the last program ran in the given problem

SOURCE CODE:

help – detailed help menu for Python commands

clc – clears command window

Run – to execute the code

New script-to open new file

Cmath- to use mathematical functions

Numpy –to use numbers

PYTHON OPERATORS

In python programing, Operators in general are used to perform operations on values and variables. These are standard symbols used for the purpose of logical and arithmetic operations. In this article, we will look into different types of Python operators.

- OPERATORS: These are the special symbols. Eg- + , * , /, etc.
- OPERAND: It is the value on which the operator is applied.

DATA TYPES IN PYTHON

Data types are the classification or categorization of data items. It represents the kind of value that tells what operations can be performed on a particular data. Since everything is an object in python programing, data types are actually classes and variables are instances (object) of these classes.

The following are the standard or built-in data types in Python:

- Numeric
- Sequence Type
- Boolean
- Set
- Dictionary
- Binary Types(memory view, byte array, bytes)

Assignment Operator

The assignment operator “=” is used for assigning a value (or a matrix) to variable. On the left-hand side of “=” is placed a variable name to which the value on right hand side is to be assigned.

Variable Names

Variable names may be as long as the user wants, but only the first 24 characters are taken into account in Python. For consistency, we should consider only variable names which are not made of more than 24 characters. All ASCII letters from "a" to "z", from "A" to "Z" and from "0" to "9" are allowed with the additional letters "%", "_", "#", "!", "\$", "?".

Basic Python Commands for Mathematical Computations

Python is a powerful open-source software for numerical computation. Below are some fundamental commands used in Python for mathematical operations:

Displaying Output in the Console:

print () function: `print()` displays variables or strings. It's useful for formatted output or messages. For example, `x = 10; print(x)`; displays 10. `disp("The value of x is: " + string(x))`; displays "The value of x is: 10". `disp("Hello, World!")`; displays "Hello, World!".

Method to take input in the Console:

Assigning values: `x = 10; y = "hello";`

Using input(): `a=int(input('Enter the value : '))`

1. Basic Arithmetic Operations

- Addition: `a + b`
- Subtraction: `a - b`
- Multiplication: `a * b`
- Division: `a / b`
- Pow(`a, b`)

2. Mathematical Functions

- Square root: `sqrt(x)`
- Exponential: `exp(x)`
- Logarithm (natural): `log(x)`
- Logarithm (base 10): `log10(x)`

- Sine: `sin(x)`
- Cosine: `cos(x)`
- Tangent: `tan(x)`
- Absolute value: `abs(x)`

3. Matrix Operations

- Creating a matrix:

```
a=np.array([[1,2,3],
            [1,2,3],
            [1,2,3]])
```

4. Defining Variables & Functions

Assigning a value: `x = 5`

Defining a function: `def fact(x):`

5. Loops and Conditional Statements

For Loop:

`for i in range (1,var+1):`

`print(i)`

If Statement:

`if x > 0 then`

`print("Positive number")`

`elif x==0`

`print("x is zero ")`

`elif`

`print("Non-positive number")`

6. Drawing a circle

```
import turtle as tur
t=tur.Turtle()
t.speed(0)
t.pensize(1)
```

Syntax

1. Matrix Addition and Multiplication

```
a=np.array([[1,2,3],[1,2,3],[1,2,3]])
b=np.array([[1,2,3],[1,2,3],[1,2,3]])
r=np.add(a,b)
s=np.dot(a,b)
```

2. Inbuilt Functions

```
def functionname(variable):
    return()
```

3. Loops and Conditional Statements

For Loop:

```
for i in range (1,var+1):
```

```
    print(i)
```

If Statement:

```
if x > 0 then
    print("Positive number")
elif x==0
    print("x is zero ")
elif
    print("Non-positive number")
```

4. Circle of squares

```
import turtle as tur
t=tur.Turtle()
t.speed(0)
t.pensize(1)
for i in range (50):
    for j in ['red','green']:
        t.color(j)
        t.fd(200)
        t.lt(91)
```

Problem 1:

Write a python program to print pyramid pattern of given number.

Aim: To print pyramid pattern of given numbers using python.

Program:

```
# program to print pyramid
a=int(input('Enter the number : '))
for i in range (1,a+1):
    # print spaces
    print(" " * (a - i), end="")
    # print numbers
    for j in range (1,i+1):
        print(j,end=" ")
    print()
```

Output:

```
Enter the number : 10
      1
     1 2
    1 2 3
   1 2 3 4
  1 2 3 4 5
 1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 10
```

Problem 2:

Write a python program to find the roots a quadratic equation

Aim: To write a Python program to find the roots of the quadratic equation.

Program:

```
import cmath

print("General Form: ax^2+bx+c=0")
a=int(input("Enter a(a!=0):"))
b=int(input("Enter b :"))
c=int(input("Enter c :"))
d=pow(b,2)-(4*a*c)
sol1=(-b-cmath.sqrt(d))/(2*a)
sol2=(-b+cmath.sqrt(d))/(2*a)
print("\n")
print(f"Results for equation, {a}x^2+{b}x+{c}=0, \n are:")
if d>0:
    print("Type of Roots:Two Distinct Real Roots")
elif d==0:
    print("Type of Roots:Two Equal Real Roots")
elif d<0:
    print("Type of roots:Two Complex Roots")
print("The solutions are", sol1, "and", sol2)
```

Output:

```
General Form: ax^2+bx+c=0
Enter a(a!=0):1
Enter b :5
Enter c :6

Results for equation, {a}x^2+{b}x+{c}=0,
are:
Type of Roots:Two Distinct Real Roots
The solutions are (-3+0j) and (-2+0j)
```

Problem 3:

Write a python program to find the factorial of the given number using function.

Aim: To write a python program to find the factorial of the given number using function.

Program:

```
x=int(input("Enter a number : "))

def fact(x):

    if x==0:

        return 1

    else:

        return x*fact(x-1)

print("The factorial of %d is %d"%(x,fact(x)))
```

Output:

```
Enter a number : 5
The factorial of 5 is 120
```

Problem 4:

Write a Python program to implement Matrix Addition and Multiplication.

Aim: To write a Python program to compute Matrix Addition and Multiplication .

Program:

```
import numpy as np

a=np.array([[1,2,3],[1,2,3],[1,2,3]])

b=np.array([[1,2,3],[1,2,3],[1,2,3]])

r=np.add(a,b)

s=np.dot(a,b)

print('Matrix a : \n',a)

print('Matrix b : \n',b)

print('Addtion of matrices : \n',r)

print('Multiplication of matrices : \n',s)
```

Output:

```
Matrix a :
[[1 2 3]
 [1 2 3]
 [1 2 3]]
Matrix b :
[[1 2 3]
 [1 2 3]
 [1 2 3]]
Addtion of matrices :
[[2 4 6]
 [2 4 6]
 [2 4 6]]
Multiplication of matrices :
[[ 6 12 18]
 [ 6 12 18]
 [ 6 12 18]]
```

Problem 5:

Write a python program to find the largest number in a list using function.

Aim: To write a python program to find the largest number in a list using function.

Program:

Function to find the largest number in a list

```
def find_largest(numbers):
```

```
    largest = numbers[0]    # Assume first element is the largest
```

```
    for num in numbers:
```

```
        if num > largest:
```

```
            largest = num
```

```
    return largest
```

Taking input from user

```
n = int(input("Enter how many numbers: "))
```

```
numbers = [ ]
```

```
for i in range(n):
```

```
    value = int(input(f"Enter number {i+1}: "))
```

```
    numbers.append(value)
```

Calling the function

```
result = find_largest(numbers)
```

```
print("The largest number in the list is:", result)
```

Output:

```
Enter how many numbers: 5
Enter number 1: 5
Enter number 2: 7
Enter number 3: 3
Enter number 4: 8
Enter number 5: 12
The largest number in the list is: 12
```

Problem 6:

Write a python program to find the area of shapes using function.

Aim: To write a Python program to find the area of shapes using function.

Program:

```
import math
#Square
s=input("Enter the Name of Shape : ")
if (s=='square'):
    a=int(input("Enter the side : "))
    area=pow(a,2)
    print("Area of the Square = ",area)
#Rectangle
elif (s=='rectangle'):
    b=int(input("Enter the breadth : "))
    h=int(input("Enter the height : "))
    area=b*h
    print("Area of the Rectangle = ",area)
#Triangle
elif (s=='triangle'):
    b=int(input("Enter the base : "))
    h=int(input("Enter the height : "))
    area=(1/2)*b*h
    print("Area of the Triangle = ",area)
#Circle
elif (s=='circle'):
    r=int(input("Enter the radius : "))
    area=math.pi * pow(r,2)
    print("Area of the Circle = ",area)
else:
    print("Please enter the Shape's Name")
```

Output:

Enter the Name of Shape : square
Enter the side : 3
Area of the Square = 9

Enter the Name of Shape : rectangle
Enter the breadth : 8
Enter the height : 9
Area of the Rectangle = 72

Enter the Name of Shape : triangle
Enter the base : 5
Enter the height : 7
Area of the Triangle = 17.5

Enter the Name of Shape : circle
Enter the radius : 6
Area of the Circle = 113.09733552923255

Problem 7:

Write a python program to find reverse string, string palindrome, character count and replacing string.

Aim: To write a Python program to find reverse string, string palindrome, character count and replacing string.

Program:

```
import re
str=input("Enter the string:")
count = 0
for i in range(0,len(str)):
    if(str[i]!=""):
        count = count + 1
print("Total number of characters in a string: ",count)
m=""
for i in str:
    m=i+m
print("The Original String is: ",str)
print("The Reversed String is: ",m)
if(str==str[::-1]):
    print("The string is a palindrome")
else:
    print("The string isn't a palindrome")
print("Replaced string:",re.sub("[a-z,A-Z]", "#",str))
```

Output:

```
Enter the string:MadaM
Total number of characters in a string:  5
The Original String is:  MadaM
The Reversed String is:  MadaM
The string is a palindrome
Replaced string: #####
```

```
Enter the string:Rani
Total number of characters in a string:  4
The Original String is:  Rani
The Reversed String is:  inaR
The string isn't a palindrome
Replaced string: #####
```

Problem 8:

Write a python program to create EB bill.

Aim: To write a python program to create EB bill.

Program:

```
# Python program to calculate Electricity (EB) Bill
def calculate_bill(units):
    if units <= 100:
        amount = units * 1.50
    elif units <= 200:
        amount = (100 * 1.50) + (units - 100) * 2.50
    elif units <= 300:
        amount = (100 * 1.50) + (100 * 2.50) + (units - 200) * 4.00
    else:
        amount = (100 * 1.50) + (100 * 2.50) + (100 * 4.00) + (units - 300) * 5.00
    return amount

# Taking user input
name = input("Enter Customer Name: ")
units = int(input("Enter Total Units Consumed: "))

# Calculate total bill
bill_amount = calculate_bill(units)

# Display the bill
print("\n----- Electricity Bill -----")
print("Customer Name :", name)
print("Units Consumed :", units)
print("Total Amount : ₹", round(bill_amount, 2))
print("-----")
```

Output:

```
Enter Customer Name: R UMA
Enter Total Units Consumed: 170

----- Electricity Bill -----
Customer Name : R UMA
Units Consumed : 170
Total Amount : ₹ 325.0
-----
```

Problem 9:

Write a python program to swap two variables.

Aim: To write a python program to swap two variables.

Program:

```
a=int(input("Enter the number1 = "))
b=int(input("Enter the number2 = "))
print("Before Swapping")
print(a)
print(b)
temp=a
a=b
b=temp
print(" After Swapping")
print(a)
print(b)
```

Output:

```
Enter the number1 = 89
Enter the number2 = 76
Before Swapping
89
76
 After Swapping
76
89
```

Problem 10:

Write a python program to draw a circle of square using turtle.

Aim: To write a Scilab program to compute matrix addition and multiplication.

Program:

```
import turtle as tur
```

```
t=tur.Turtle()
```

```
t.speed(0)
```

```
t.pensize(1)
```

```
for i in range (50):
```

```
    for j in ['red','green']:
```

```
        t.color(j)
```

```
        t.fd(200)
```

```
        t.lt(91)
```

Output: