

RISHI MS INSTITUTE OF ENGINEERING & TECHNOLOGY for WOMEN

**(Approved by AICTE, New Delhi and Affiliated to JNTUH)
Nizampet Cross Roads, JNTUH Kukatpally Hyderabad – 500085**



DEPARTMENT OF INFORMATION TECHNOLOGY

LAB MANUAL

PYTHON PROGRAMMING

B.Tech I YEAR II SEM (R22 REGULATIONS)

ACADEMIC YEAR 2023-24

INDEX

S.NO	TOPIC	PAGE NO
I	List of Experiments	ii
II	V/M /POs/PSOs/PEOs	iv
III	Syllabus	ix
IV	Course Objectives & Course Outcomes	xi

List of Experiments

Exp. No.	Experiment Name	Page No.
1	<ol style="list-style-type: none"> 1. i) Use a web browser to go to the Python website http://python.org. This page contains information about Python and links to Python-related pages, and it gives you the ability to search the Python documentation. ii) Start the Python interpreter and type <code>help()</code> to start the online help utility. 2. Start a Python interpreter and use it as a Calculator. 3. <ol style="list-style-type: none"> i) Write a program to calculate compound interest when principal, rate and number of periods are given. ii) Given coordinates (x1, y1), (x2, y2) find the distance between two points 4. Read name, address, email and phone number of a person through keyboard and print the details. 	
2	<ol style="list-style-type: none"> 1. Print the below triangle using for loop. <pre> 5 4 4 3 3 3 2 2 2 2 1 1 1 1 1 </pre> 2. Write a program to check whether the given input is digit or lowercase character or uppercase character or a special character (use 'if-else-if' ladder) 3. Python Program to Print the Fibonacci sequence using while loop 4. Python program to print all prime numbers in a given interval (use break 	
3	<ol style="list-style-type: none"> 1. i) Write a program to convert a list and tuple into arrays. ii) Write a program to find common values between two arrays. 2. Write a function called gcd that takes parameters a and b and returns their greatest common divisor. 3. Write a function called palindrome that takes a string argument and returns True if it is a palindrome and False otherwise. Remember that you can use the built-in function <code>len</code> to check the length of a string. 	

4	<ol style="list-style-type: none"> 1. Write a function called <code>is_sorted</code> that takes a list as a parameter and returns True if the list is sorted in ascending order and False otherwise. 2. Write a function called <code>has_duplicates</code> that takes a list and returns True if there is any element that appears more than once. It should not modify the original list. <ol style="list-style-type: none"> i). Write a function called <code>remove_duplicates</code> that takes a list and returns a new list with only the unique elements from the original. Hint: they don't have to be in the same order. ii). The wordlist I provided, <code>words.txt</code>, doesn't contain single letter words. So you might want to add "I", "a", and the empty string. iii). Write a python code to read dictionary values from the user. Construct a function to invert its content. i.e., keys should be values and values should be keys. 3. <ol style="list-style-type: none"> i) Add a comma between the characters. If the given word is 'Apple', it should become 'A,p,p,l,e' ii) Remove the given word in all the places in a string? iii) Write a function that takes a sentence as an input parameter and replaces the first letter of every word with the corresponding upper case letter and the rest of the letters in the word by corresponding letters in lower case without using a built-in function? 4. Writes a recursive function that generates all binary strings of n-bit length 	
5	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> i) Write a python program that defines a matrix and prints ii) Write a python program to perform addition of two square matrices iii) Write a python program to perform multiplication of two square matrices 2. How do you make a module? Give an example of construction of a module using different geometrical shapes and operations on them as its functions. 3. Use the structure of exception handling all general purpose exceptions 	
6	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> a. Write a function called <code>draw_rectangle</code> that takes a Canvas and a Rectangle as arguments and draws a representation of the Rectangle on the Canvas. b. Add an attribute named <code>color</code> to your Rectangle objects and modify <code>draw_rectangle</code> so that it uses the <code>color</code> attribute as the fill color. c. Write a function called <code>draw_point</code> that takes a Canvas and a Point as arguments and draws a representation of the Point on the Canvas. d. Define a new class called Circle with appropriate attributes and instantiate a few Circle objects. Write a function called <code>draw_circle</code> that draws circles on the canvas. 2. Write a Python program to demonstrate the usage of Method Resolution Order (MRO) in multiple levels of Inheritances. 3. Write a python code to read a phone number and email-id from the user 	

	and validate it for correctness.	
7	<ol style="list-style-type: none"> 1. Write a Python code to merge two given file contents into a third file. 2. Write a Python code to open a given file and construct a function to check for given words present in it and display on found. 3. Write a Python code to Read text from a text file, find the word with most number of occurrences. 4. Write a function that reads a file file1 and displays the number of words, number of vowels, blank spaces, lower case letters and uppercase letters. 	
8	<ol style="list-style-type: none"> 1. Import numpy, Plotpy and Scipy and explore their functionalities. 2. a) Install NumPy package with pip and explore it. 3. Write a program to implement Digital Logic Gates – AND, OR, NOT, EX-OR 4. Write a program to implement Half Adder, Full Adder, and Parallel Adder 5. Write a GUI program to create a window wizard having two text labels, two text fields and two buttons as Submit and Reset 	



RISHI M.S INSTITUTE OF ENGINEERING & TECHNOLOGY FOR WOMEN

(Affiliated to JNTUH University, Approved by AICTE)

Department of Information Technology

Vision of the institution:

To be a center of excellence in producing women engineers and scientists who are professionally competent social leaders to face multi-disciplinary global environment by imparting quality technical education, values and ethics through innovation methods of teaching and learning.

Mission of the institution:

- To promote women technocrats capable enough to resolve the problems faced by the society using the knowledge imparted.
- To prepare self-reliant women engineering for technological growth of the nation and society by laying strong theoretical foundation accompanied by wide practical training.
- To equip the young women with creative thinking capabilities and empowering them towards innovation.



RISHI M.S INSTITUTE OF ENGINEERING & TECHNOLOGY FOR WOMEN

(Affiliated to JNTUH University, Approved by AICTE)

Department of Information Technology

Vision & Mission of Department

Vision of the department

To empower women by providing cutting-edge technology to female technocrats in the fields of Information Technology, allowing them to develop into competent engineers and entrepreneurs.

Mission of the department

- Adopting creative techniques to nurture and strengthen the core skill of Computer Science.
- Introduce students to the most recent technological advancements.
- Impart quality education, improve the research, entrepreneurial, and employability skills of women technocrats.
- Instil professional ethics and a sense of social responsibility in students.
- Strengthen the Industry-Academia interface, which will enable graduates to emerge as academic leaders or inspiring entrepreneurs



RISHI M.S INSTITUTE OF ENGINEERING & TECHNOLOGY FOR WOMEN

(Affiliated to JNTUH University, Approved by AICTE)

Program outcomes (POs)

1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem Analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/Development of Solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct Investigations of Complex Problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The Engineer and Society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and Sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and Team Work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project Management and Finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.



RISHI M.S INSTITUTE OF ENGINEERING & TECHNOLOGY FOR WOMEN
(Affiliated to JNTUH University, Approved by AICTE)

Department of Information Technology

Program specific outcomes (PSOs)

- PSO 1:** Improve the student's ability to decipher the basic principles and methodology of computer systems. Improve the students' ability to absorb facts and technical ideas in order to build and develop software.
- PSO 2:** The capacity to create novel job routes as an entrepreneur using modern computer languages and evolving technologies like SDLC, Python, Machine Learning, Social Networks, Cyber Security, Mobile Apps etc.



RISHI M.S INSTITUTE OF ENGINEERING & TECHNOLOGY FOR WOMEN

(Affiliated to JNTUH University, Approved by AICTE)

Department of Information Technology

Program educational objectives (PEOs)

PEO-1: Engineering graduates with excellent fundamental and technical skills will have successful careers in industry, meeting the needs of Indian and worldwide firms.

PEO-2: With determination, development, self-reliance, leadership, morality, and moral principles, engineering graduates will become successful entrepreneurs who will leverage employability.

PEO-3: To support personal and organisational progress, engineering graduates will pursue higher education and engage in lifelong learning.

SYLLABUS
Python Programming Lab
B.TECH I Year II Sem

Week -1:

1. i) Use a web browser to go to the Python website <http://python.org>. This page contains information about Python and links to Python-related pages, and it gives you the ability to search the Python documentation.
- ii) Start the Python interpreter and type `help()` to start the online help utility.
2. Start a Python interpreter and use it as a Calculator.
- 3.
- i) Write a program to calculate compound interest when principal, rate and number of periods are given.
- ii) Given coordinates (x1, y1), (x2, y2) find the distance between two points
4. Read name, address, email and phone number of a person through keyboard and print the details.

Week - 2:

1. Print the below triangle using for loop.
5
4 4
3 3 3
2 2 2 2
1 1 1 1 1
2. Write a program to check whether the given input is digit or lowercase character or uppercase character or a special character (use 'if-else-if' ladder)
3. Python Program to Print the Fibonacci sequence using while loop
4. Python program to print all prime numbers in a given interval (use break)

Week - 3:

1. i) Write a program to convert a list and tuple into arrays.
- ii) Write a program to find common values between two arrays.
2. Write a function called `gcd` that takes parameters `a` and `b` and returns their greatest common divisor.
3. Write a function called `palindrome` that takes a string argument and returns `True` if it is a palindrome and `False` otherwise. Remember that you can use the built-in function `len` to check the length of a string.

Week - 4:

1. Write a function called `is_sorted` that takes a list as a parameter and returns `True` if the list is sorted in ascending order and `False` otherwise.

2. Write a function called `has_duplicates` that takes a list and returns `True` if there is any element that appears more than once. It should not modify the original list.
 - i). Write a function called `remove_duplicates` that takes a list and returns a new list with only the unique elements from the original. Hint: they don't have to be in the same order.
 - ii). The wordlist I provided, `words.txt`, doesn't contain single letter words. So you might want to add "I", "a", and the empty string.
 - iii). Write a python code to read dictionary values from the user. Construct a function to invert its content. i.e., keys should be values and values should be keys.
3.
 - i) Add a comma between the characters. If the given word is 'Apple', it should become 'A,p,p,l,e'
 - ii) Remove the given word in all the places in a string?
 - iii) Write a function that takes a sentence as an input parameter and replaces the first letter of every word with the corresponding upper case letter and the rest of the letters in the word by corresponding letters in lower case without using a built-in function?
4. Writes a recursive function that generates all binary strings of n-bit length

Week - 5:

1.
 - i) Write a python program that defines a matrix and prints
 - ii) Write a python program to perform addition of two square matrices
 - iii) Write a python program to perform multiplication of two square matrices
2. How do you make a module? Give an example of construction of a module using different geometrical shapes and operations on them as its functions.
3. Use the structure of exception handling all general purpose exceptions.

Week-6:

1.
 - a. Write a function called `draw_rectangle` that takes a `Canvas` and a `Rectangle` as arguments and draws a representation of the `Rectangle` on the `Canvas`.
 - b. Add an attribute named `color` to your `Rectangle` objects and modify `draw_rectangle` so that it uses the `color` attribute as the fill color.
 - c. Write a function called `draw_point` that takes a `Canvas` and a `Point` as arguments and draws a representation of the `Point` on the `Canvas`.
 - d. Define a new class called `Circle` with appropriate attributes and instantiate a few `Circle` objects. Write a function called `draw_circle` that draws circles on the canvas.
2. Write a Python program to demonstrate the usage of Method Resolution Order (MRO) in multiple levels of Inheritances.
3. Write a python code to read a phone number and email-id from the user and validate it for correctness.

Week- 7

1. Write a Python code to merge two given file contents into a third file.
2. Write a Python code to open a given file and construct a function to check for given words present in it and display on found.
3. Write a Python code to Read text from a text file, find the word with most number of occurrences
4. Write a function that reads a file file1 and displays the number of words, number of vowels, blank spaces, lower case letters and uppercase letters.

Week - 8:

1. Import numpy, Plotpy and Scipy and explore their functionalities.
2. a) Install NumPy package with pip and explore it.
3. Write a program to implement Digital Logic Gates – AND, OR, NOT, EX-OR
4. Write a program to implement Half Adder, Full Adder, and Parallel Adder
5. Write a GUI program to create a window wizard having two text labels, two text fields and two buttons as Submit and Reset.

TEXT BOOKS:

1. Supercharged Python: Take your code to the next level, Overland
2. Learning Python, Mark Lutz, O'reilly

REFERENCE BOOKS:

1. Python for Data Science, Dr. Mohd. Abdul Hameed, Wiley Publications - 1st Ed. 2021.
2. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
3. Python Programming A Modular Approach with Graphics, Database, Mobile, and Web Applications, Sheetal Taneja, Naveen Kumar, Pearson
4. Programming with Python, A User's Book, Michael Dawson, Cengage Learning, India Edition
5. Think Python, Allen Downey, Green Tea Press
6. Core Python Programming, W. Chun, Pearson
7. Introduction to Python, Kenneth A. Lambert, Cengage

Course Objectives: To learn

1. Understand the usage of data types, loops and conditional statements and functions.
2. Understand Lists, Dictionaries and Regular expressions in Python.
3. Handle Strings and Files in Python.
4. Learn the implantation of Python modules like Numpy, Plotpy, Turtle

Course Outcomes: After learning the contents of this course the student is able to

CO1. Examine Python syntax and semantics and be fluent in the use of Python flow control and functions. Demonstrate proficiency in handling Strings and File Systems.

CO2. Create, run and manipulate Python Programs using core data structures like Lists, Tuples, and Dictionaries.

CO3. Implement exemplary applications related to Numpy, Pandas and matplotlib in Python.

CO - PO MAPPING:

	CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
Python Programming Lab	CO1	3	2	2	2	2							2
	CO2	3	2	3	2	3							3
	CO3	3	3	3	3	3							3

CO - PSO MAPPING:

	PSO-1	PSO-2
CO1	1	3
CO2	2	3
CO3	2	3

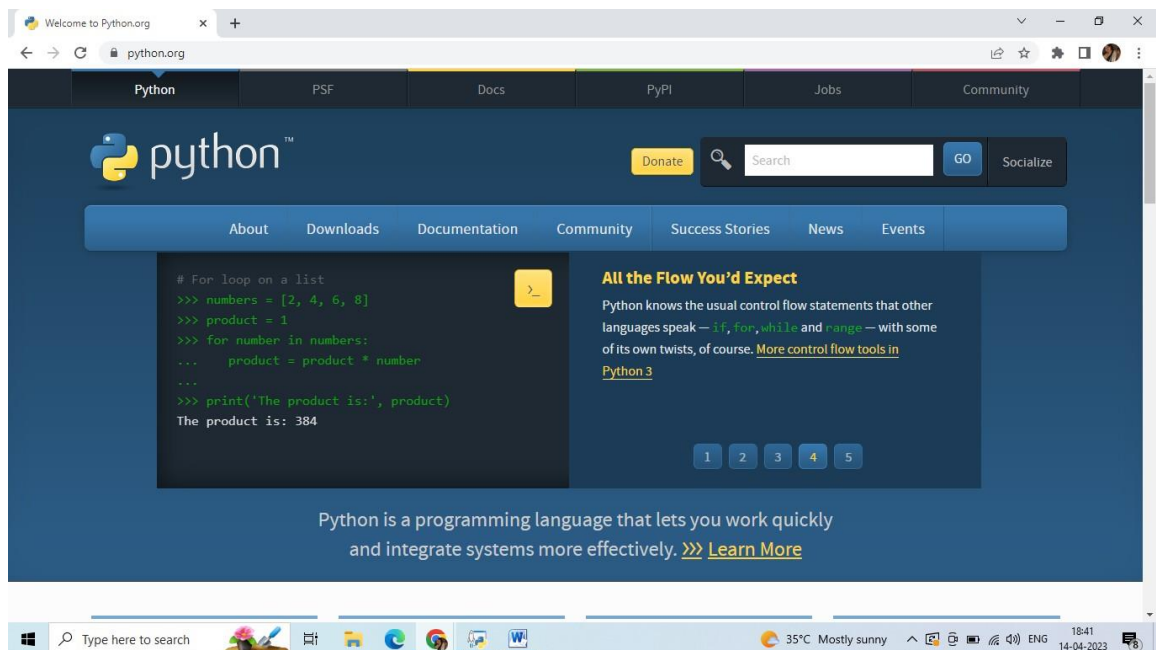
Week 1

- i) Use a web browser to go to the Python website <http://python.org>. This page contains information about Python and links to Python-related pages, and it gives you the ability to search the Python documentation.

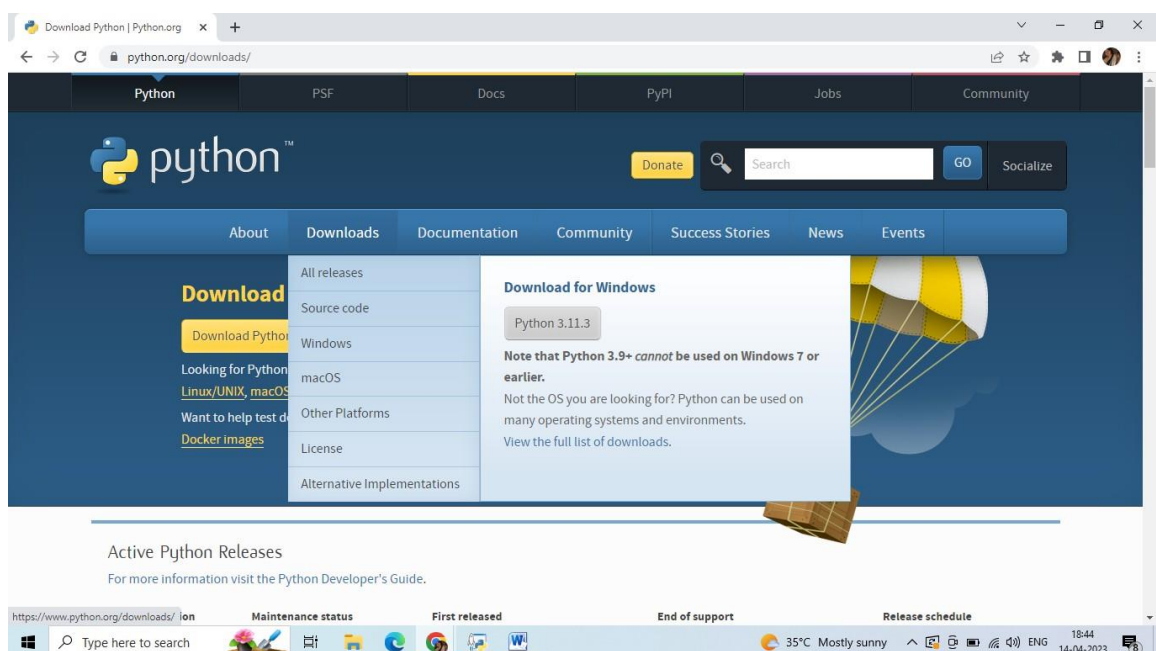
ii) Start the Python interpreter and type `help()` to start the online help utility.

Stepss:

Open any web browser and type <http://python.org>

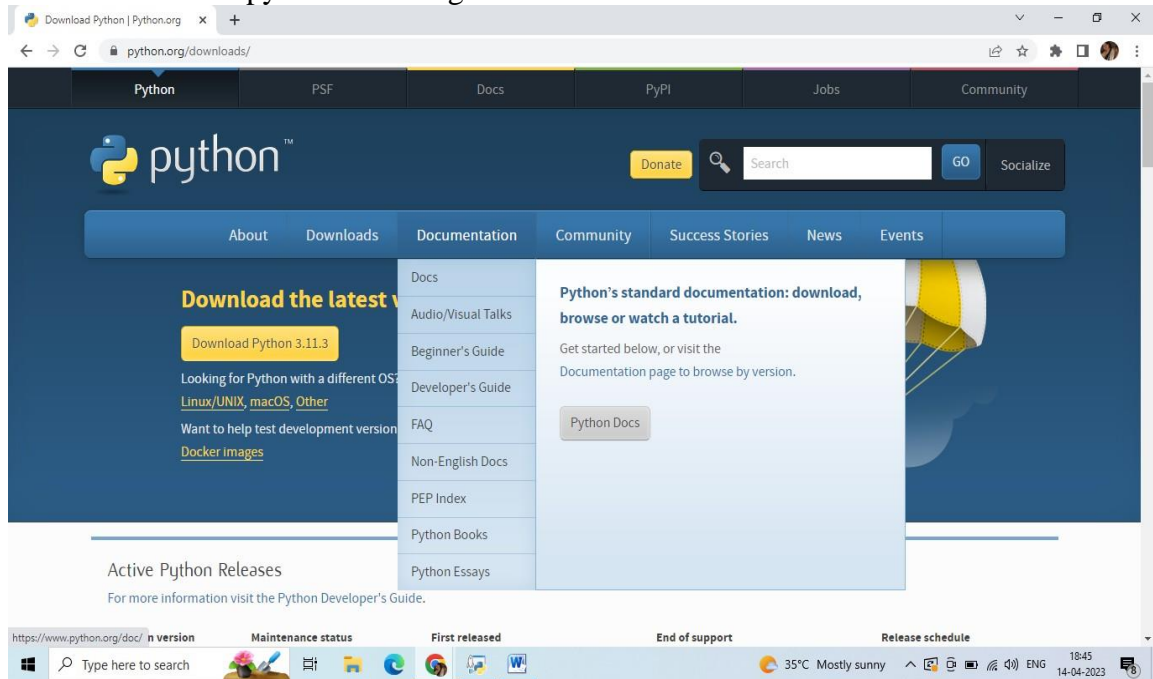


You can download latest version of python for any operating system and install them.

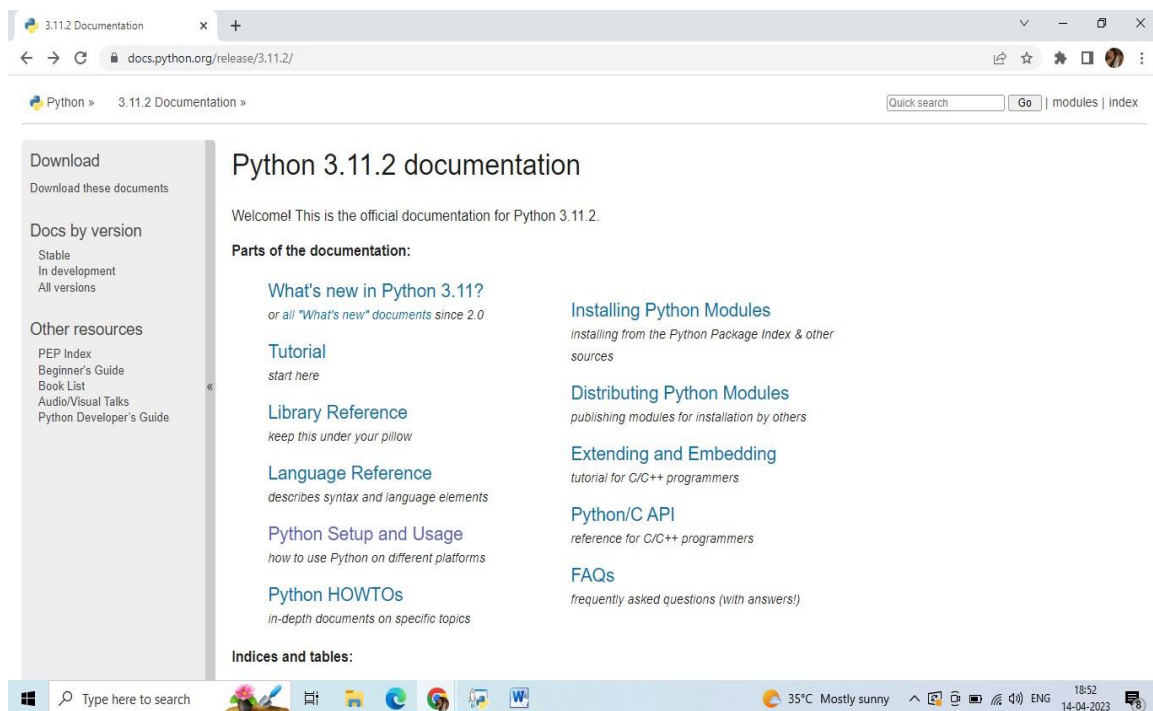


Python Documentation acts as a user manual and provides with all the required

information about python including tutorials.

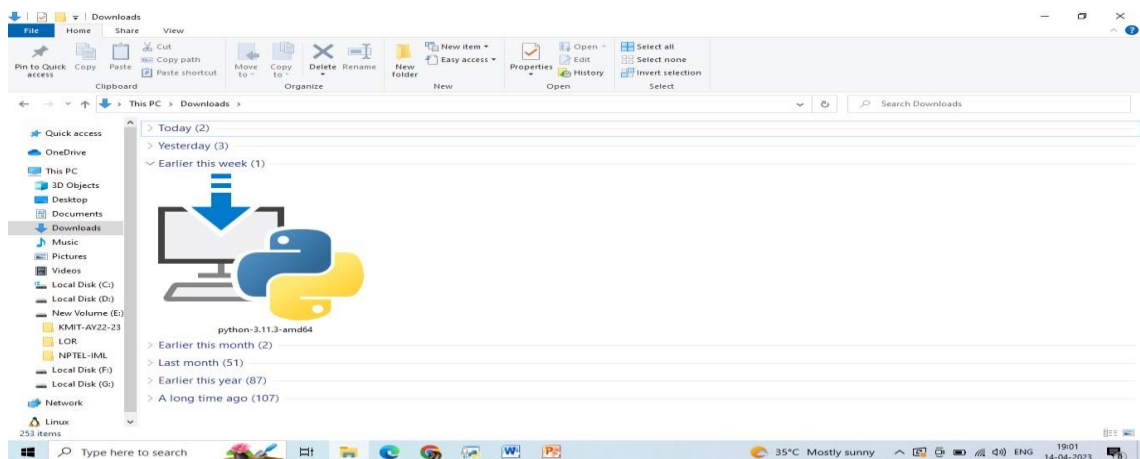
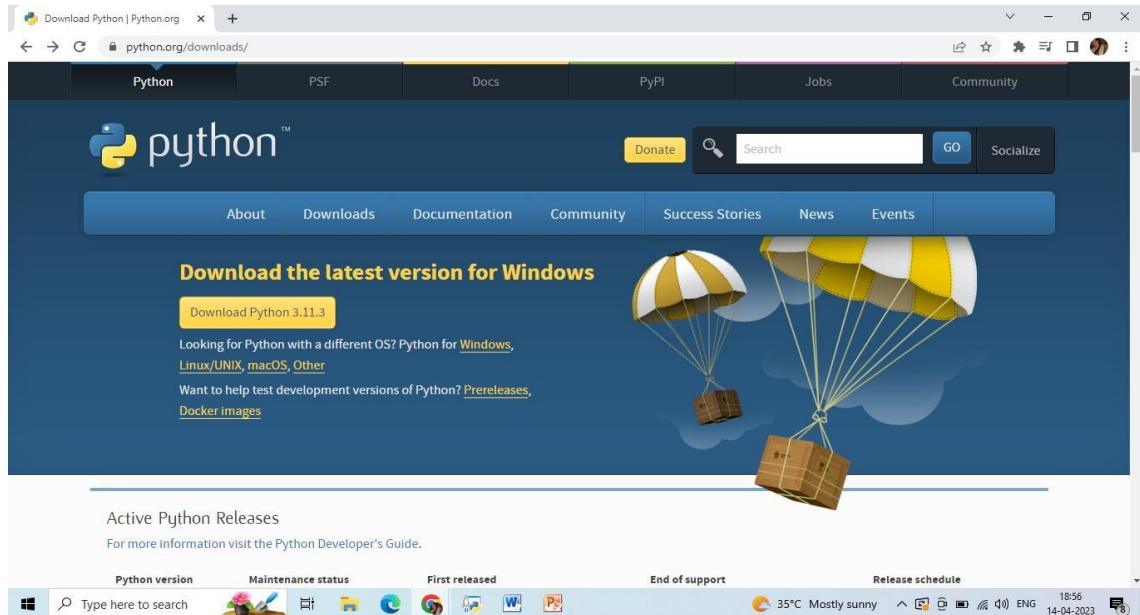


Sample of the documentation page



Installing Python on Windows

1. Download the latest version installer for Windows from the official website
“<http://python.org>”

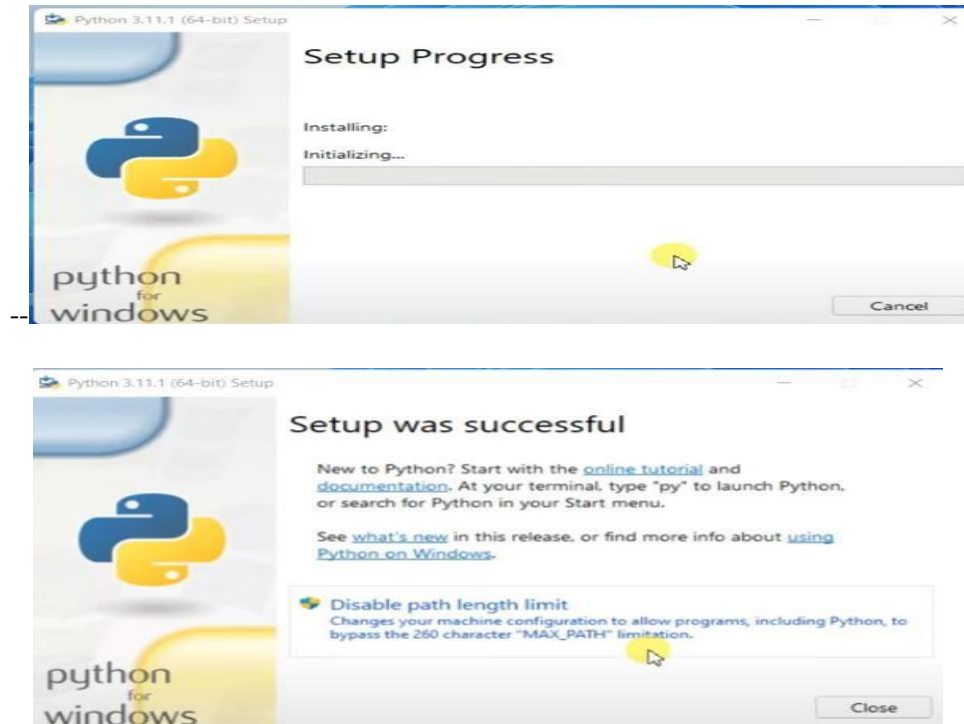


2. Run the Setup

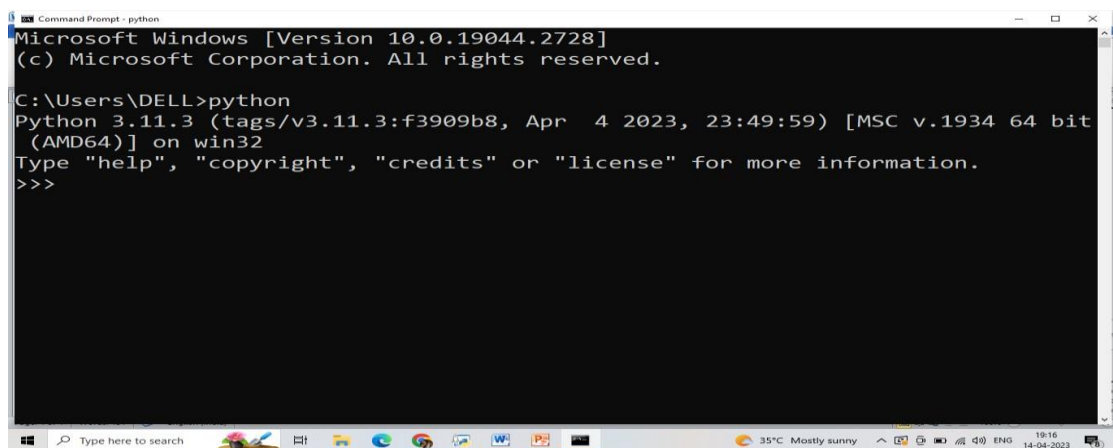


Check this

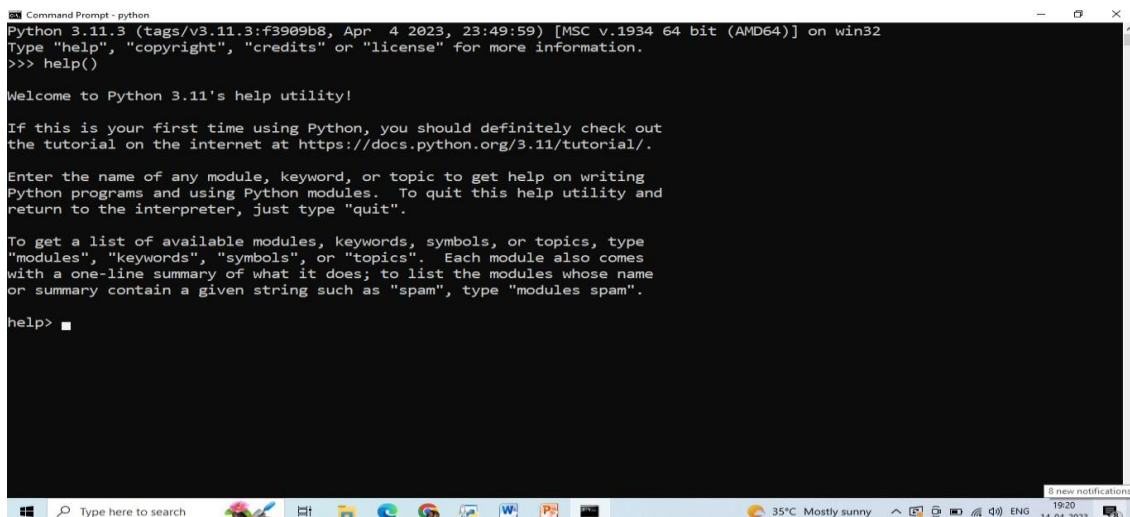




1. To start the python interpreter :
2. Goto Windows Command Prompt and type python to run the interpreter



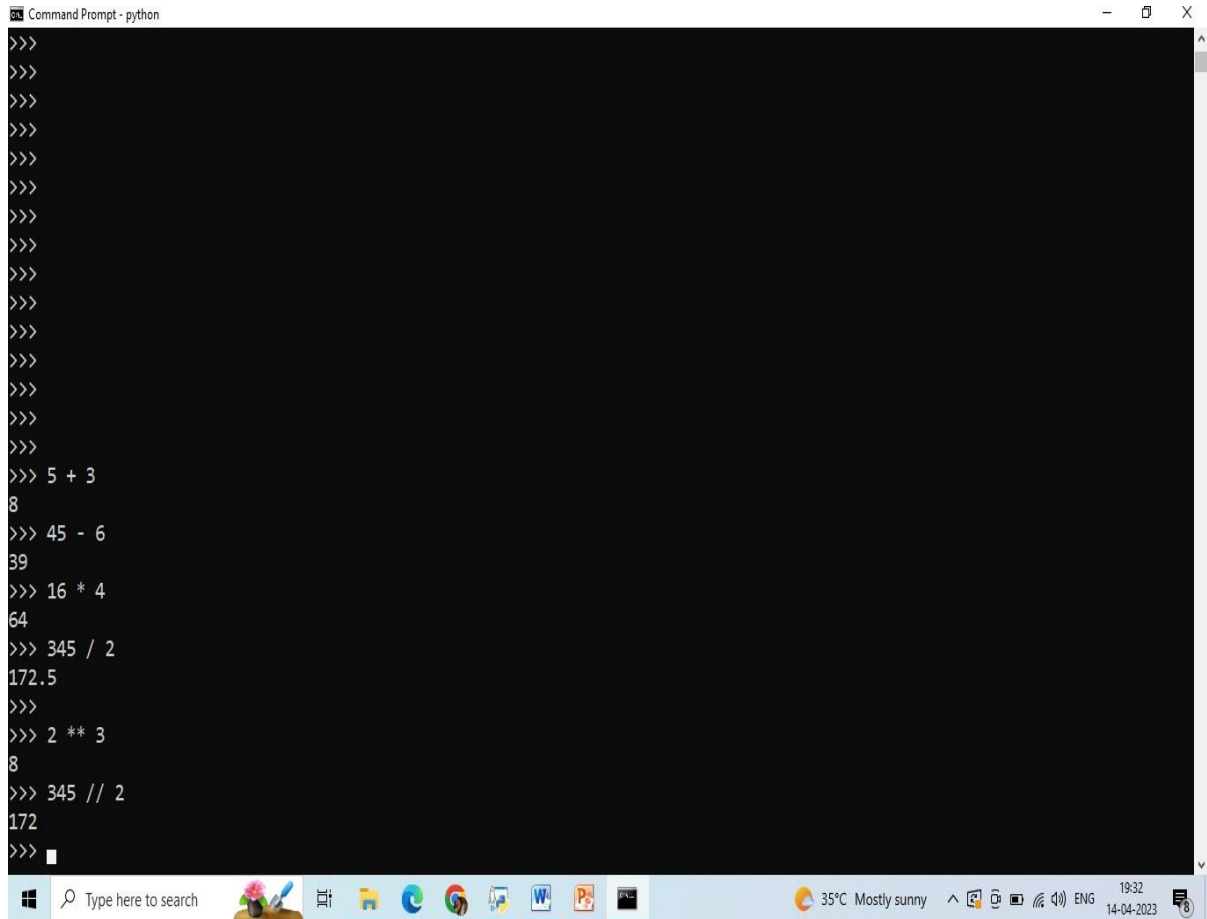
3. Type **help()** to start the online help utility
- 4.



2. Start a Python interpreter and use it as a Calculator.

Simple arithmetic calculations can be completed at the Python Prompt, also called the *Python REPL*. REPL stands for Read Evaluate Print Loop.

The Python REPL shows three arrow symbols `>>>` followed by a blinking cursor. Programmers type commands at the `>>>` prompt then hit [ENTER] to see the results.



```
Command Prompt - python
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> 5 + 3
8
>>> 45 - 6
39
>>> 16 * 4
64
>>> 345 / 2
172.5
>>>
>>> 2 ** 3
8
>>> 345 // 2
172
>>> 
```

3. i) Write a program to calculate compound interest when principal, rate and number of periods are given.

ii) Given coordinates (x1, y1), (x2, y2) find the distance between two points

Program :

Python program to find compound interest for given values.

```
principal = int(input("enter principal amount : "))
rate = float(input("enter rate of interest : "))
time = int(input("enter period : "))
```

```
# Calculates compound interest
Amount = principal * (pow((1 + rate / 100), time))
CI = Amount - principal
print("Compound interest is : ", CI)
```

Sample Input and Output 1:

enter principal amount : 10000
enter rate of interest : 8.5
enter period : 5
Compound interest is : 5036.566901781247

Sample Input and Output 2:

enter principal amount : 1500
enter rate of interest : 6.25
enter period : 2
Compound interest is : 193.359375

ii) Given coordinates (x1, y1), (x2, y2) find the distance between two points**Program :**

Python program to find distance between two points

import math

```
x1=int(input("enter x1 : "))
x2=int(input("enter x2 : "))
y1=int(input("enter y1 : "))
y2=int(input("enter y2 : "))
result = math.sqrt(((x2 - x1 )**2) + ((y2-y1)**2))
print("distance between",(x1,x2),"and",(y1,y2),"is : ",result)
```

Note : instead of using the math module - square root can be derived using exponentiation

result = (((x2 - x1)2) + ((y2-y1)**2))**0.5)**

Sample Input and Output 1:

enter x1 : 4
enter x2 : 6
enter y1 : 1
enter y2 : 5
distance between (4, 6) and (1, 5) is : 4.47213595499958

Sample Input and Output 2:

enter x1 : 0
enter x2 : 3
enter y1 : 8
enter y2 : 7
distance between (0, 3) and (8, 7) is : 3.1622776601683795

4. Read name, address, email and phone number of a person through keyboard and print the details.

Program:

```
# python program to display personal details
name = input("enter your name : ")
address = input("enter address : ")
email = input("enter email id : ")
mobile = input("enter mobile number : ")

print("\nPersonal Details \n")
print("Name: {} \nAddress: {} \nEmail: {} \nMobile: {}".format(name, address, email,
mobile))
```

Sample Input and Output 1:

```
enter your name : Trishika
enter address : Sainikpuri, Secunderabad
enter email id : shika@gmail.com
enter mobile number : 8777675747
```

Personal Details

```
Name: Trishika
Address: Sainikpuri, Secunderabad
Email: shika@gmail.com
Mobile: 8777675747
```

Week 2

1. Print the below triangle using for loop.

```
5
4 4
3 3 3
2 2 2 2
1 1 1 1 1
```

Program:

```
#python program to print numbers in triangular form using for loop

rows = int(input("Enter number of rows: "))

for i in range(rows,0,-1):
    print("\n")
    for j in range(rows-i+1):
        print(i, " ", end="")
```

Sample Input and Output 1:

Enter number of rows: 5

```
5
4 4
3 3 3
2 2 2 2
1 1 1 1 1
```

Sample Input and Output 2:

Enter number of rows: 8

```
8
7 7
6 6 6
5 5 5 5
4 4 4 4 4
3 3 3 3 3 3
2 2 2 2 2 2 2
1 1 1 1 1 1 1 1
```

2. Write a program to check whether the given input is digit or lowercase character or uppercase character or a special character (use 'if-else-if' ladder)

Program:

```
#To check given Input is Digit or Uppercase or Lowercase or Special Character

ch = input("Enter a character: ")
if ch.isdigit():
    print("Digit")
```

```
elif ch.isupper ():
    print("Uppercase character")
elif ch.islower ():
    print("Lowercase character")
else:
    print("Special character")
```

Sample Input and Output 1:

Enter a character: 5
Digit

Sample Input and Output 2:

Enter a character: x
Lowercase character

Sample Input and Output 3:

Enter a character: A
Uppercase character

Sample Input and Output 4:

Enter a character: &
Special character

4. Python Program to Print the Fibonacci sequence using while loop

Program:

```
# Program to display the Fibonacci sequence up to n-th term

nterms = int(input("How many terms? "))

# first two terms
n1, n2 = 0, 1
count = 0

# check if the number of terms is valid
if nterms <= 0:
    print("Please enter a positive integer")
# if there is only one term, return n1
elif nterms == 1:
    print("Fibonacci sequence upto",nterms,":")
    print(n1)
# generate fibonacci sequence
else:
    print("Fibonacci sequence:")
    while count < nterms:
        print(n1, end=" ")
        n3 = n1 + n2
        # update values
        n1 = n2
        n2 = n3
        count += 1
```

Sample Input and Output 1:

How many terms? 10
Fibonacci sequence:
0 1 1 2 3 5 8 13 21 34

Sample Input and Output 2:

How many terms? 25
Fibonacci sequence:
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584
4181 6765 10946 17711 28657 46368

4. Python program to print all prime numbers in a given interval (use break)**Program:**

```
# Python program to display all the prime numbers within an interval

lower = int(input("enter starting limit : "))
upper = int(input("enter ending limit : "))

print("\nPrime numbers between", lower, "and", upper, "are:")

for num in range(lower, upper + 1):
    # all prime numbers are greater than 1
    if num > 1:
        for i in range(2, num):
            if (num % i) == 0:
                break
        else:
            print(num, " ", end=" ")
```

Sample Input and Output 1:

enter starting limit : 100
enter ending limit : 200

Prime numbers between 100 and 200 are:
101 103 107 109 113 127 131 137 139 149 151 157 163 167 173 179
181 191 193 197 199

Sample Input and Output 2:

enter starting limit : 10
enter ending limit : 50

Prime numbers between 10 and 50 are:
11 13 17 19 23 29 31 37 41 43 47

Week 3

1. i) Write a program to convert a list and tuple into arrays.

Program:

```
import numpy as np
my_list = [1, 2, 3, 4, 5, 6, 7, 8]
print("\n",type(my_list))
print(my_list)

print("List to array: ")
list_array = np.asarray(my_list)
print(type(list_array))
print(list_array)

my_tuple = ([8, 4, 6], [1, 2, 3])
print("\n",type(my_tuple))
print(my_tuple)
print("Tuple to array: ")
tuple_array = np.asarray(my_tuple)
print(type(tuple_array))
print(tuple_array)
```

Sample Input and Output 1:

```
<class 'list'>
[1, 2, 3, 4, 5, 6, 7, 8]
List to array:
<class 'numpy.ndarray'>
[1 2 3 4 5 6 7 8]

<class 'tuple'>
([8, 4, 6], [1, 2, 3])
Tuple to array:
<class 'numpy.ndarray'>
[[8 4 6]
 [1 2 3]]
```

ii) Write a program to find common values between two arrays

Program:

```
import numpy as np

ar1 = [0, 1, 2, 3, 4, 7, 10]
ar2 = [10, 3, 4, 5, 6, 7]
# Common values between two arrays
print("array1 : ",ar1)
print("array2 : ",ar2)
print("Common values : ")
print(np.intersect1d(ar1, ar2))
```

Sample Input and Output 1:

```
array1 : [0, 1, 2, 3, 4, 7, 10]
array2 : [10, 3, 4, 5, 6, 7]
Common values :
[ 3 4 7 10]
```

2. Write a function called gcd that takes parameters a and b and returns their greatest common divisor.

Program:

```
# Python code to compute gcd using Euclidean method( recursion )

def gcd(a, b):
    if(b == 0):
        return a
    else:
        return gcd(b, a % b)

"""
# Python code to demonstrate subtraction method to compute gcd ( recursion )

def gcd(a, b):

    # Everything divides 0
    if (a == 0):
        return b
    if (b == 0):
        return a

    # base case
    if (a == b):
        return a

    # a is greater
    if (a > b):
        return gcd(a-b, b)
    return gcd(a, b-a)

"""

print("Program to find gcd of two given numbers \n")
a = int(input("enter a number : "))
b = int(input("enter another number : "))

#function call
print("The gcd of ",a," and ",b," is : ", end="")
print(gcd(a, b))

Sample Input and Output 1:
enter a number : 60
enter another number : 25
```

The gcd of 60 and 25 is : 5

Sample Input and Output 2:

enter a number : 12
enter another number : 60
The gcd of 12 and 60 is : 12

Sample Input and Output 3 :

enter a number : 23
enter another number : 0
The gcd of 23 and 0 is : 23

Sample Input and Output 4 :

enter a number : 17
enter another number : 27
The gcd of 17 and 27 is : 1

3. Write a function called **palindrome** that takes a string argument and returns **True** if it is a **palindrome** and **False** otherwise.
(Remember that you can use the built-in function **len** to check the length of a string.)

Program :

Method 1:

```
# function which return reverse of a string

def Palindrome(s):
    return s == s[::-1]
```

Method 2:

```
# by comparing characters (left and right)
def Palindrome(str):

    # Run loop from 0 to len/2
    for i in range(0, int(len(str)/2)):
        if str[i] != str[len(str)-i-1]:
            return False
    return True
```

Method 3:

```
# Using predefined function to reverse the string
def Palindrome(s):

    rev = "".join(reversed(s))

    # Checking if both string are equal or not
    if (s == rev):
        return True
    return False
```

Driver Code:

```
print("Check if given string is a Palindrome or Not ")
s = input("enter a string ")

if Palindrome(s):
    print("True")
else:
    print("False")
```

Note: Any one of the function definitions can be used to check for palindrome

Sample Input and Output 1 :

```
enter a string madam
True
```

Sample Input and Output 2 :

```
enter a string xyz
False
```

Sample Input and Output 3 :

```
enter a string malayalam
True
```

Week – 4

1. Write a function called `is_sorted` that takes a list as a parameter and returns `True` if the list is sorted in ascending order and `False` otherwise.

Program:

```
def is_sorted(a):
    print("Given List is : ",a)
    for i in range(len(a)-1):
        #print(a[i], a[i+1])
        if a[i] > a[i+1]:
            return False
    return True

if (is_sorted([31,42,53,74,95,96])):
    print("List is in sorted")
else:
    print("List is not sorted")
```

Sample Input and Output 1:

Given List is : [31, 42, 53, 74, 95, 96]
List is in sorted

Sample Input and Output 2:

Given List is : [31, 42, 53, 74, 15, 96]
List is not sorted

2. Write a function called `has_duplicates` that takes a list and returns `True` if there is any element that appears more than once. It should not modify the original list.

Program:

```
def has_duplicates(source):
    return len(set(source)) != len(source)

print(has_duplicates([1,2,3,4,4,5,6,7,8]))
```

Sample Input and Output 1:

True

Sample Input and Output 1:

False (if the function call is `print(has_duplicates([1,2,3,4,5,6,7,8]))`)

i). Write a function called `remove_duplicates` that takes a list and returns a new list with only the unique elements from the original. Hint: they don't have to be in the same order.

Program:

```
def Remove(duplicate):
    final_list = []
    for num in duplicate:
        if num not in final_list:
            final_list.append(num)
    return final_list

# Driver Code
duplicate = [2, 4, 10, 20, 5, 2, 20, 4]
print(Remove(duplicate))
```

Sample Input and Output 1:

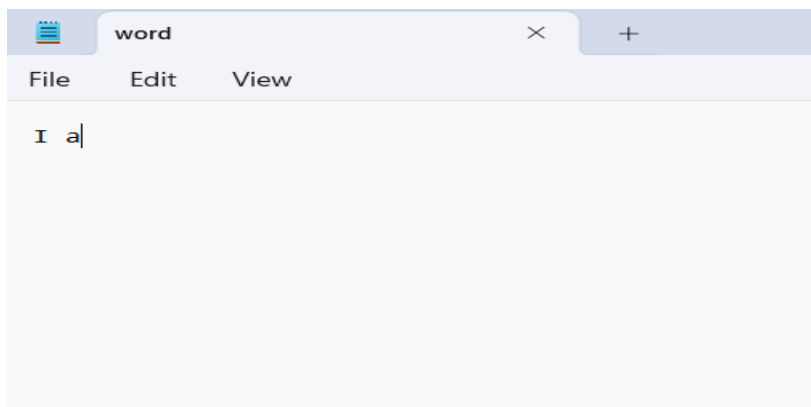
[2, 4, 10, 20, 5]

ii). The wordlist I provided, `words.txt`, doesn't contain single letter words. So you might want to add "I", "a", and the empty string.

Program:

```
with open("word.txt","w") as file:
    file.write("I ")
    file.write("a")
```

Sample Input and Output 1:



iii). Write a python code to read dictionary values from the user. Construct a function to invert its content. i.e., keys should be values and values should be keys.

Program:

```
n = int(input("Enter the no of entries you want to enter : "))
dict = {}
for i in range(n):
    key = int(input("Enter the key : "))
    value = input("Enter the value : ")
    dict[key]=value
```

```

print(dict)

newdict={}
for pair in dict.items():
    newdict[pair[1]]=pair[0]

print(newdict)

```

Sample Input and Output 1:

```

Enter the no of entries you want to enter : 4
Enter the key : 3
Enter the value : 2
Enter the key : 6
Enter the value : 4
Enter the key : 8
Enter the value : 1
Enter the key : 9
Enter the value : 5
{3: '2', 6: '4', 8: '1', 9: '5'}
{'2': 3, '4': 6, '1': 8, '5': 9}

```

3. i) Add a comma between the characters. If the given word is 'Apple', it should become 'A,p,p,l,e'

Program:

```

txt = "a" "p" "p" "l" "e"
commas_added = ','.join(txt.split())
print(commas_added)
output:
"a", "p", "p", "l", "e"

(or)
import re

# initializing string
test_str = input("Enter a word:")

# Distance between occurrences
# Using re.findall()
res = re.findall(r'\w+|\S', test_str)

# printing result
print("Result : " + str(res))
'''
test_str = input("Enter a word:")
res = [s.strip() for s in test_str.split(',')]

```

Sample Input and Output 1:

```

Enter a word:a p l e
Result : ['a', 'p', 'p', 'l', 'e']

```

ii) Remove the given word in all the places in a string?

Program:

```
print("Enter the String: ")
text = input()

print("Enter a Word to Delete: ")
word = input()

text = text.replace(word, "")

print()
print(text)
```

Sample Input and Output 1:

```
Enter the String:
hello welcome to RISHI
Enter a Word to Delete:
hello
```

```
welcome to RISHI
```

iii) Write a function that takes a sentence as an input parameter and replaces the first letter of every word with the corresponding upper case letter and the rest of the letters in the word by corresponding letters in lower case without using a built-in function?

Program:

```
txt=input("Enter a sentence:")
print(txt.title())
```

Sample Input and Output 1:

```
Enter a sentence:hello, welcome to rishi
Hello, Welcome To Rishi
```

4. Writes a recursive function that generates all binary strings of n-bit length

Program:

```
def printTheArray(arr, n):

    for i in range(0, n):
        print(arr[i], end = " ")

    print()

# Function to generate all binary strings
def generateAllBinaryStrings(n, arr, i):

    if i == n:
        printTheArray(arr, n)
        return
```



```
arr[i] = 0
generateAllBinaryStrings(n, arr, i + 1)

arr[i] = 1
generateAllBinaryStrings(n, arr, i + 1)

# Driver Code
if __name__ == "__main__":

    n = 4
    arr = [None] * n

    # Print all binary strings
    generateAllBinaryStrings(n, arr, 0)
```

Sample Input and Output 1:

```
0 0 0 0
0 0 0 1
0 0 1 0
0 0 1 1
0 1 0 0
0 1 0 1
0 1 1 0
0 1 1 1
1 0 0 0
1 0 0 1
1 0 1 0
1 0 1 1
1 1 0 0
1 1 0 1
1 1 1 0
1 1 1 1
```

Week - 5:

1 i) Write a python program that defines a matrix and prints

Program:

```
row=int(input("Enter No of Rows for 1st Matrix:"))
column=int(input("Enter No of column for 1nd Matrix:"))
row1=int(input("Enter No of Rows for 2st Matrix:"))
column1=int(input("Enter No of column for 2nd Matrix:"))
X = [[int(input(("Enter value for X["i","j"]")))
for j in range(column)] for i in range(row)]
Y = [[int(input(("Enter value for Y["i","j"]")))
for j in range(column1)] for i in range(row1)]
print("1st Matrix X:",X)
print("2st Matrix Y:",Y)
```

Sample Input and Output:

```
Enter No of Rows for 1st Matrix:3
Enter No of column for 1nd Matrix:3
Enter No of Rows for 2st Matrix:3
Enter No of column for 2nd Matrix:3
('Enter value for X[' 0, '][' 0, ']:')2
('Enter value for X[' 0, '][' 1, ']:')1
('Enter value for X[' 0, '][' 2, ']:')4
('Enter value for X[' 1, '][' 0, ']:')5
('Enter value for X[' 1, '][' 1, ']:')2
('Enter value for X[' 1, '][' 2, ']:')7
('Enter value for X[' 2, '][' 0, ']:')6
('Enter value for X[' 2, '][' 1, ']:')8
('Enter value for X[' 2, '][' 2, ']:')9
('Enter value for Y[' 0, '][' 0, ']:')4
('Enter value for Y[' 0, '][' 1, ']:')10
('Enter value for Y[' 0, '][' 2, ']:')5
('Enter value for Y[' 1, '][' 0, ']:')7
('Enter value for Y[' 1, '][' 1, ']:')21
('Enter value for Y[' 1, '][' 2, ']:')34
('Enter value for Y[' 2, '][' 0, ']:')11
('Enter value for Y[' 2, '][' 1, ']:')6
('Enter value for Y[' 2, '][' 2, ']:')3
```

1st Matrix X: [[2, 1, 4], [5, 2, 7], [6, 8, 9]]

2st Matrix Y: [[4, 10, 5], [7, 21, 34], [11, 6, 3]]

ii) Write a python program to perform addition of two square matrices

Program:

```
import numpy as np

X = [[1,2,3],
     [4,5,6],
     [7,8,9]]

Y = [[9,8,7],
     [6,5,4],
     [3,2,1]]

result = np.array(X) + np.array(Y)

print(result)
```

Sample Input and Output:

```
[[10 10 10]
 [10 10 10]
 [10 10 10]]
```

iii) Write a python program to perform multiplication of two square matrices

Program:

```
# 3x3 matrix
X = [[12,7,3],
     [4,5,6],
     [7,8,9]]
# 3x4 matrix
Y = [[5,8,1,2],
     [6,7,3,0],
     [4,5,9,1]]
# result is 3x4
result = [[0,0,0,0],
          [0,0,0,0],
          [0,0,0,0]]

# iterate through rows of X
for i in range(len(X)):
    # iterate through columns of Y
    for j in range(len(Y[0])):
        # iterate through rows of Y
        for k in range(len(Y)):
            result[i][j] += X[i][k] * Y[k][j]
for r in result:
    print(r)
```

Sample Input and Output:

[114, 160, 60, 27]
[74, 97, 73, 14]
[119, 157, 112, 23]

2. Use the structure of exception handling all general purpose exceptions**Program:**

```
try:  
    even_numbers = [2,4,6,8]  
    print(even_numbers[5])  
  
except ZeroDivisionError:  
    print("Denominator cannot be 0.")  
  
except IndexError:  
    print("Index Out of Bound.")
```

Sample Input and Output:

Index Out of Bound.

Week-6:

1.
 - a. Write a function called `draw_rectangle` that takes a `Canvas` and a `Rectangle` as arguments and draws a representation of the `Rectangle` on the `Canvas`.
 - b. Add an attribute named `color` to your `Rectangle` objects and modify `draw_rectangle` so that it uses the `color` attribute as the fill color.
 - c. Write a function called `draw_point` that takes a `Canvas` and a `Point` as arguments and draws a representation of the `Point` on the `Canvas`.
 - d. Define a new class called `Circle` with appropriate attributes and instantiate a few `Circle` objects. Write a function called `draw_circle` that draws circles on the canvas

Program:

```
import tkinter as tk

root = tk.Tk()

canvas = tk.Canvas(root, width=400, height=400)
canvas.pack()

canvas.create_rectangle(10, 10, 100, 100, fill='red')
root.mainloop()
```

2. Write a Python program to demonstrate the usage of Method Resolution Order (MRO) in multiple levels of Inheritances.

Program:

```
class ClassA:
    def m(self):
        print("In ClassA")

class ClassB(ClassA):
    def m(self):
        print("In ClassB")

class ClassC(ClassA):
    def m(self):
        print("In ClassC")

class ClassD(ClassB, ClassC):
    def m(self):
        print("In ClassD")
        ClassB.m(self)
        ClassC.m(self)
        ClassA.m(self)

obj = ClassD()
obj.m()
```

Sample Input and Output:

In ClassD
In ClassB
In ClassC
In ClassA

2. Write a python code to read a phone number and email-id from the user and validate it for correctness.**Program:**

```
import re

regex = re.compile(r'([A-Za-z0-9]+[-_])*[A-Za-z0-9]+@[A-Za-z0-9-]+(\.[A-Z|a-z]{2,})+')

def isValid(email):
    if re.fullmatch(regex, email):
        print("Valid email")
    else:
        print("Invalid email")
isValid("name.surname@gmail.com")
isValid("anonymous123@yahoo.co.uk")
isValid("anonymous123@...uk")
isValid("...@domain.us")
```

Sample Input and Output:

Valid email
Valid email
Invalid email
Invalid email

For mobilenumber:

```
import re

def validate_phone_number(regex, phone_number):
    match = re.search(regex, phone_number)
    if match:
        return True
    return False

pattern = re.compile(r"(\+|d{1,3})?s?(?d{1,4})?[\s.-]?d{3}[\s.-]?d{4}")

test_phone_numbers = [
    "+1 (555) 123-4567",
    "555-123-4567",
    "555 123 4567",
    "+44 (0) 20 1234 5678",
    "02012345678",
```

```
    "invalid phone number"  
]
```

```
for number in test_phone_numbers:  
    print(f"{number}: {validate_phone_number(pattern, number)}")
```

Sample Input and Output:

```
+1 (555) 123-4567: True  
555-123-4567: True  
555 123 4567: True  
+44 (0) 20 1234 5678: True  
02012345678: True  
invalid phone number: False
```

Week- 7

1. Write a Python code to merge two given file contents into a third file.

Program:

```
with open("file1.txt","r") as fh1:
    with open("file2.txt","r") as fh2:
        with open("mergefile.txt","w") as fh3:
            q=fh1.readlines()+fh2.readlines()
            fh3.writelines(q)
```

2. Write a Python code to open a given file and construct a function to check for given words present in it and display on found.

Program:

```
with open('file1.txt') as file:
    contents = file.read()
    search_word = input("enter a word you want to search in file: ")
    if search_word in contents:
        print ('word found')
    else:
        print ('word not found')
```

Sample Input and output1:

```
enter a word you want to search in file: interpreted
word found
```

Sample Input and output2:

```
enter a word you want to search in file: hello
word not found
```

3. Write a Python code to Read text from a text file, find the word with most number of occurrences

Program:

```
file = open("file1.txt","r")
frequent_word = ""
frequency = 0
words = []

# Traversing file line by line
for line in file:
```



```

# splits each line into
# words and removing spaces
# and punctuations from the input
line_word = line.lower().replace(',').replace('.', '').split(" ");

# Adding them to list words
for w in line_word:
    words.append(w);

# Finding the max occurred word
for i in range(0, len(words)):

    # Declaring count
    = 1;

    # Count each word in the file
    for j in range(i+1, len(words)):
        if(words[i] == words[j]):
            count = count + 1;

    # If the count value is more
    # than highest frequency then
    if(count > frequency):
        frequency = count;
        frequent_word = words[i];

print("Most repeated word: " + frequent_word)
print("Frequency: " + str(frequency))
file.close();

```

Sample Input and output:

```

Most repeated word: python
Frequency: 3

```

- 4. Write a function that reads a file file1 and displays the number of words, number of vowels, blank spaces, lower case letters and uppercase letters.**

Program:

```

f=open("file1.txt","r")
r=f.read()
u=l=v=c=b=0
for i in r:
    if i.isalpha():
        if i.isupper():
            u=u+1

```

```
        if i.islower():
            l=l+1
        if i.isspace():
            b=b+1
        if i.lower() in ["a","e","i","o","u"]:
            v=v+1
        else:
            c=c+1
    print("Number of Uppercase characters are:",u)
    print("Number of Lowercase characters are:",l)
    print("Number of blankspaces:",b)
    print("Number of Vowels characters are:",v)
    print("Number of Consonants characters are:",u)
```

Sample Output:

```
Number of Uppercase characters are: 10
Number of Lowercase characters are: 250
Number of blankspaces: 0
Number of Vowels characters are: 98
Number of Consonants characters are: 10
```

Week - 8:

1. Import numpy, Plotpy and Scipy and explore their functionalities.

Numpy:

```
import numpy as np

# Initial Array
arr = np.array([[ -1, 2, 0, 4],
                [ 4, -0.5, 6, 0],
                [ 2.6, 0, 7, 8],
                [ 3, -7, 4, 2.0]])
print("Initial Array: ")
print(arr)

# Printing a range of Array
# with the use of slicing method
sliced_arr = arr[:2, ::2]
print ("Array with first 2 rows and"
       " alternate columns(0 and 2):\n", sliced_arr)

# Printing elements at
# specific Indices
Index_arr = arr[[1, 1, 0, 3],
                 [3, 2, 1, 0]]
print ("\nElements at indices (1, 3), "
       "(1, 2), (0, 1), (3, 0):\n", Index_arr)
```

Sample Input and Output:

```
Initial Array:
[[-1.  2.  0.  4.]
 [ 4. -0.5  6.  0.]
 [ 2.6  0.  7.  8.]
 [ 3. -7.  4.  2.]]
Array with first 2 rows and alternate columns(0 and 2):
[[-1.  0.]
 [ 4.  6.]]

Elements at indices (1, 3), (1, 2), (0, 1), (3, 0):
[0.  6.  2.  3.]
```

Pyplot

```
import matplotlib.pyplot as plt
plt.plot([1, 2, 3, 4])
plt.ylabel('some numbers')
plt.show()

import numpy as np
import matplotlib.pyplot as plt

# evenly sampled time at 200ms intervals
t = np.arange(0., 5., 0.2)

# red dashes, blue squares and green triangles
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
plt.show()
```

2. a) Install NumPy package with pip and explore it.

Python NumPy is a general-purpose array processing package which provides tools for handling the n-dimensional arrays.

NumPy installation through pip:

Syntax:

Pip install numpy

Program:

```
import numpy as np
a=np.array([10,20,30]) #creates a one dimensional array
print("one dimensional array:",a)
b=np.array([[10,20,30], [40,50,60]]) #creates a 2D array
print("Two dimensional array:",b)
c=np.zeros((3,4)) #creates array with all zeros
print("array with all zeros:",c)
d=np.ones((3,4)) #creates array with all ones.
print("array with all zeros:",d)
e=np.full((3,3),5) #creates an array with specified number
print("array with specific number:",e)
g=np.eye(3,dtype=int) #creates an identity matrix
print("Identity Matrix:",g)

print("Dimension:",b.ndim) #gives the dimension of the array
print("Shape of array:",b.shape) #gives the sequence of integers indicating the size of the
array for each dimension
```

```

print("Size of array:",b.size) #gives the total number of elements of the array
print("Data type of array elements:",b.dtype) #gives data type of the elements of the
array
print("Size of each array element in bytes:",b.itemsize) #It specifies the size in bytes of
each element of the array

```

Sample Input and Output:

```

one dimensional array: [10 20 30]
Two dimensional array: [[10 20 30]
[40 50 60]]
array with all zeros: [[0. 0. 0. 0.]
[0. 0. 0. 0.]
[0. 0. 0. 0.]]
array with all ones: [[1. 1. 1. 1.]
[1. 1. 1. 1.]]
array with specific number: [[5 5 5]
[5 5 5]
[5 5 5]]
Identity Matrix: [[1 0 0]
[0 1 0]
[0 0 1]]
Dimension: 2
Shape of array: (2, 3)
Size of array: 6
Data type of array elements: int32
Size of each array element in bytes: 4

```

3. Write a program to implement Digital Logic Gates – AND, OR, NOT, EX-OR

Program:

```

#Python3 program to illustrate working of AND gate
def AND (a, b):

    if a == 1 and b == 1:
        return True
    else:
        return False

# Driver code
if __name__=='_main_':
    print(AND(1, 1))

    print("+.....+")
    print("| AND Truth Table | Result |")
    print(" A = False, B = False | A AND B =",AND(False,False)," | ")
    print(" A = False, B = True | A AND B =",AND(False,True)," | ")

```

```

print(" A = True, B = False | A AND B =",AND(True,False)," | ")
print(" A = True, B = True | A AND B =",AND(True,True)," | ")

```

Python3 program to illustrate working of OR gate

```

def OR(a, b):
    if a == 1 or b ==1:
        return True
    else:
        return False

# Driver code
if __name__=='_main_':
    print(OR(0, 0))

    print("+.....+")
    print(" | OR Truth Table | Result |")
    print(" A = False, B = False | A OR B =",OR(False,False)," | ")
    print(" A = False, B = True | A OR B =",OR(False,True)," | ")
    print(" A = True, B = False | A OR B =",OR(True,False)," | ")
    print(" A = True, B = True | A OR B =",OR(True,True)," | ")

```

Python3 program to illustrate working of Xor gate

```

def XOR (a, b):
    if a != b:
        return 1
    else:
        return 0

# Driver code
if __name__=='_main_':
    print(XOR(5, 5))

    print("+.....+")
    print(" | XOR Truth Table | Result |")
    print(" A = False, B = False | A XOR B =",XOR(False,False)," | ")
    print(" A = False, B = True | A XOR B =",XOR(False,True)," | ")
    print(" A = True, B = False | A XOR B =",XOR(True,False)," | ")
    print(" A = True, B = True | A XOR B =",XOR(True,True)," | ")

```

Python3 program to illustrate working of Not gate

```

def NOT(a):
    return not a
# Driver code

```

```

if __name__=='_main_':
    print(NOT(0))

    print("+.....+")
    print(" | NOT Truth Table | Result |")
    print(" A = False | A NOT =",NOT(False)," | ")
    print(" A = True, | A NOT =",NOT(True)," | ")

```

Sample input and Output:

```

True
+.....+
| AND Truth Table | Result |
A = False, B = False | A AND B = False |
A = False, B = True | A AND B = False |
A = True, B = False | A AND B = False |
A = True, B = True | A AND B = True |
False
+.....+
| OR Truth Table | Result |
A = False, B = False | A OR B = False |
A = False, B = True | A OR B = True |
A = True, B = False | A OR B = True |
A = True, B = True | A OR B = True |
0
+.....+
| XOR Truth Table | Result |
A = False, B = False | A XOR B = 0 |
A = False, B = True | A XOR B = 1 |
A = True, B = False | A XOR B = 1 |
A = True, B = True | A XOR B = 0 |
True
+.....+
| NOT Truth Table | Result |
A = False | A NOT = True |
A = True, | A NOT = False |

```

4. Write a program to implement Half Adder, Full Adder, and Parallel Adder

Program:

```

# Half adder
import numpy as np
def half_adder(A, B):
    Sum = np.bitwise_xor(A, B)
    Carry = np.bitwise_and(A, B)
    return Sum, Carry

```

```
# Driver code
A = 0
B = 1
Sum, Carry = half_adder(A, B)
print("Half Adder:")
print("Sum:", Sum)
print("Carry:", Carry)
```

Sample Input and Output:

```
Half Adder:
Sum: 1
Carry: 0
```

5. Write a GUI program to create a window wizard having two text labels, two text fields and two buttons as Submit and Reset

Program:

```
from tkinter import *
window=Tk()
usernameLabel1=Label(window,text="User Name:").grid(row=0,column=0)
usernameEntry=Entry(window).grid(row=0,column=1)
passwordLabel1=Label(window,text="Password:").grid(row=1,column=0)
passwordEntry=Entry(window).grid(row=1,column=1)
submitButton=Button(window,text="Submit").grid(row=2,column=0)
resetButton=Button(window,text="Reset").grid(row=2,column=1)

window.mainloop()
```

Sample Input and Output :

