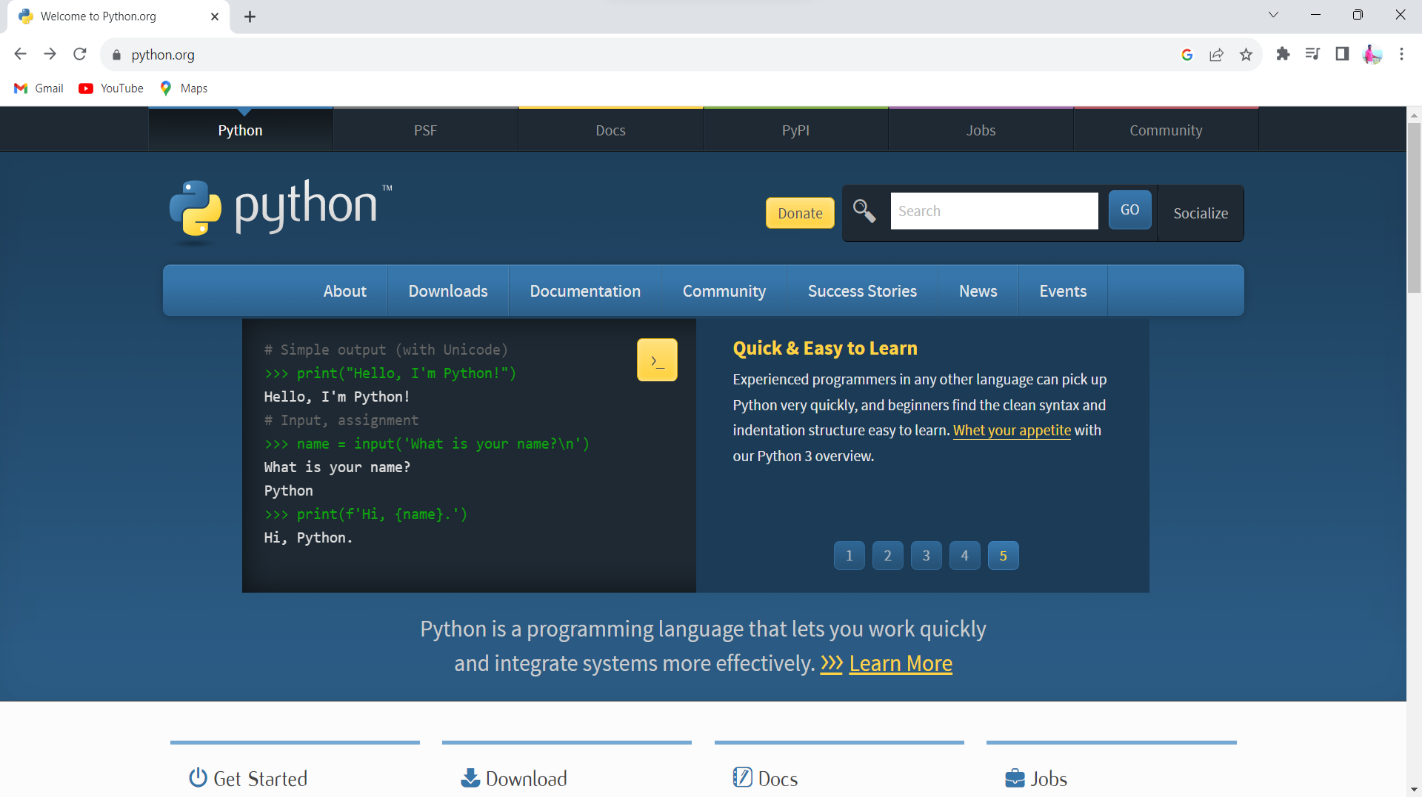
**WEEK-1**

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.**

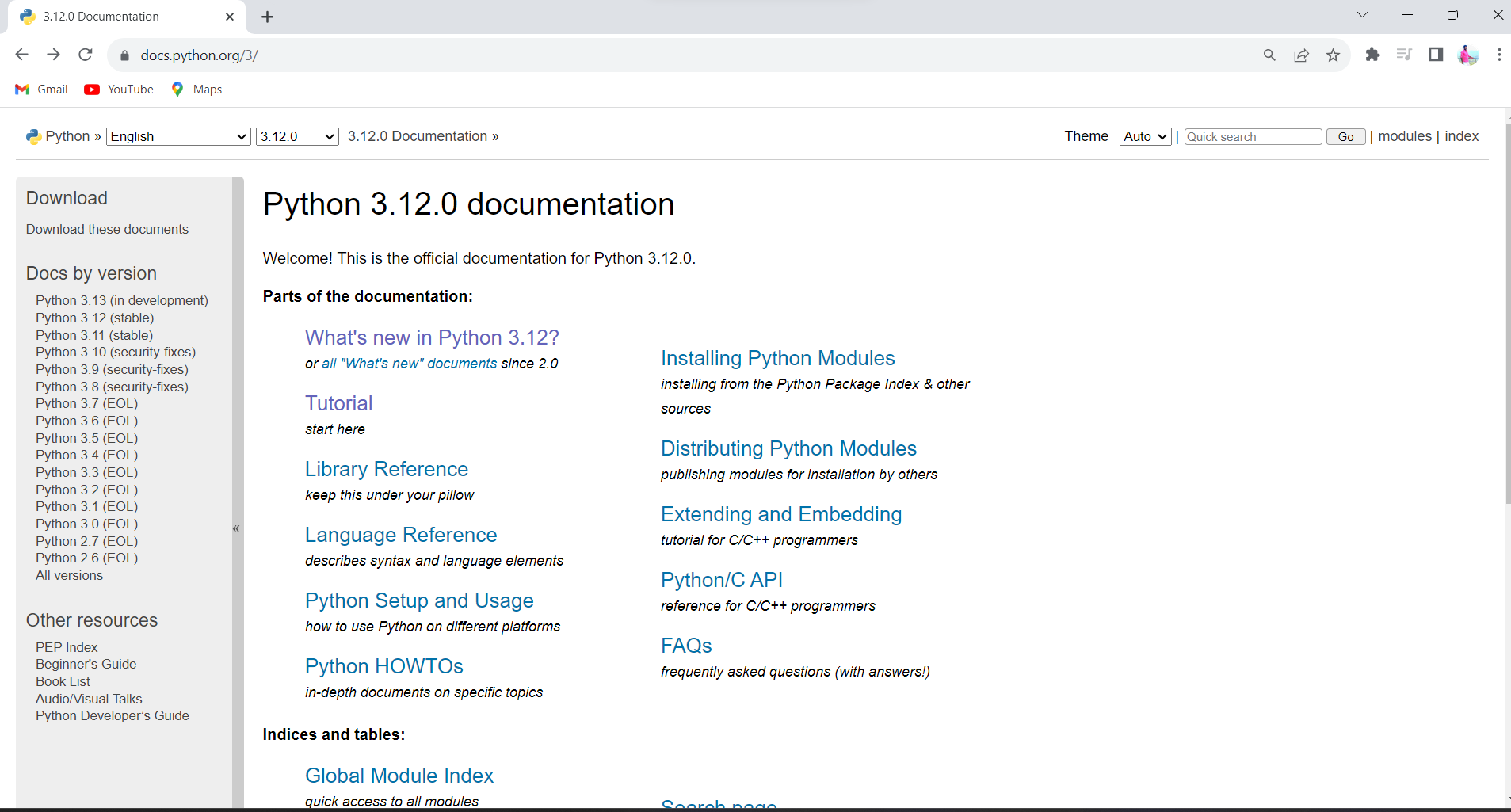
* The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python web site, <https://www.python.org/>, and may be freely distributed.
* The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.



**Python's documentation**

The Python documentation pages are available at <https://docs.python.org/3.12/>. The most helpful sections are listed here:

* The Python language reference at <https://docs.python.org/3.12/reference/index.html>
* The Python standard library documentation at <https://docs.python.org/3.12/library/index.html>



* This page contains information about Python and links to Python-related pages.
* Python Tutorials,
* Library References
* Python Setup and Usage
* Installing Python Modules etc.

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

# Python help() Method

The Python help() function invokes the interactive built-in help system. If the argument is a [string](https://www.tutorialsteacher.com/python/python-string), then the string is treated as the name of a [module](https://www.tutorialsteacher.com/python/python-module), [function](https://www.tutorialsteacher.com/python/python-user-defined-function), [class](https://www.tutorialsteacher.com/python/python-class), [keyword](https://www.tutorialsteacher.com/python/python-keywords), or documentation topic, and a help page is printed on the console. If the argument is any other kind of object, a help page on the object is displayed.

In this case, just open python interpreter and type help ( ).

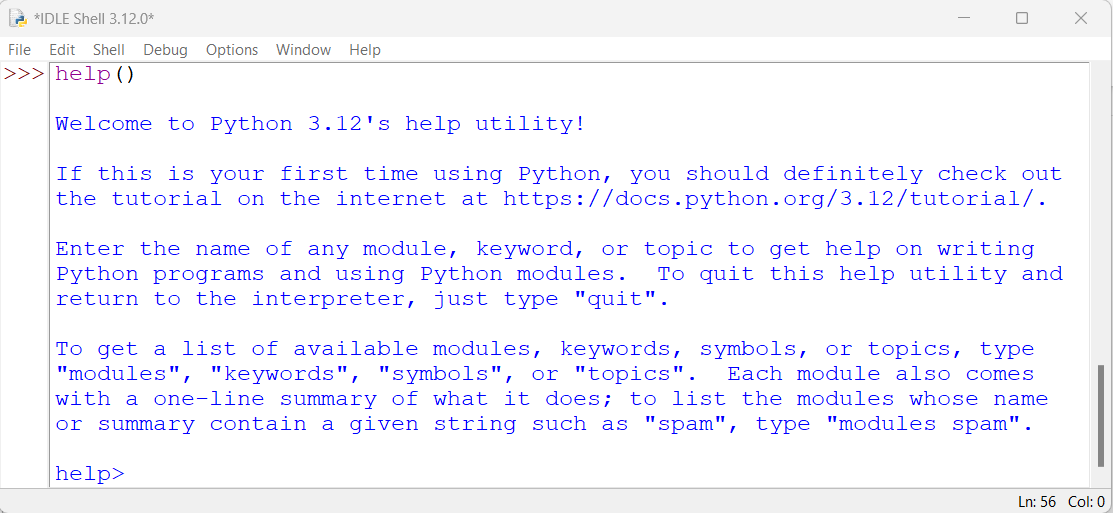
### Syntax:

help(object)

#### Parameters:

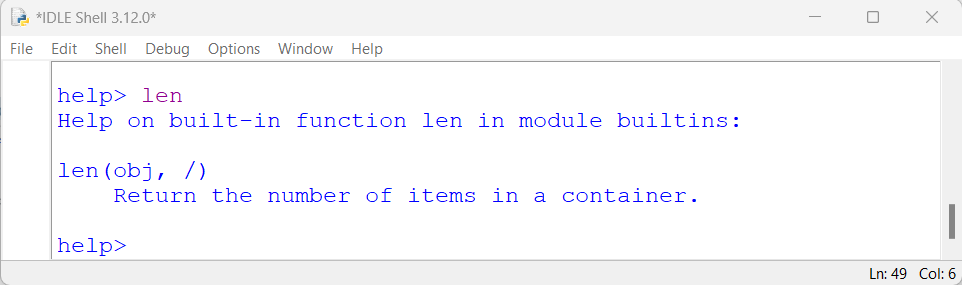
object: (Optional) The object whose documentation needs to be printed on the console.

To get help, one can enter help mode by typing help() ay python prompt(i.e. >>> prompt) we can see the help utility appearing as shown in following

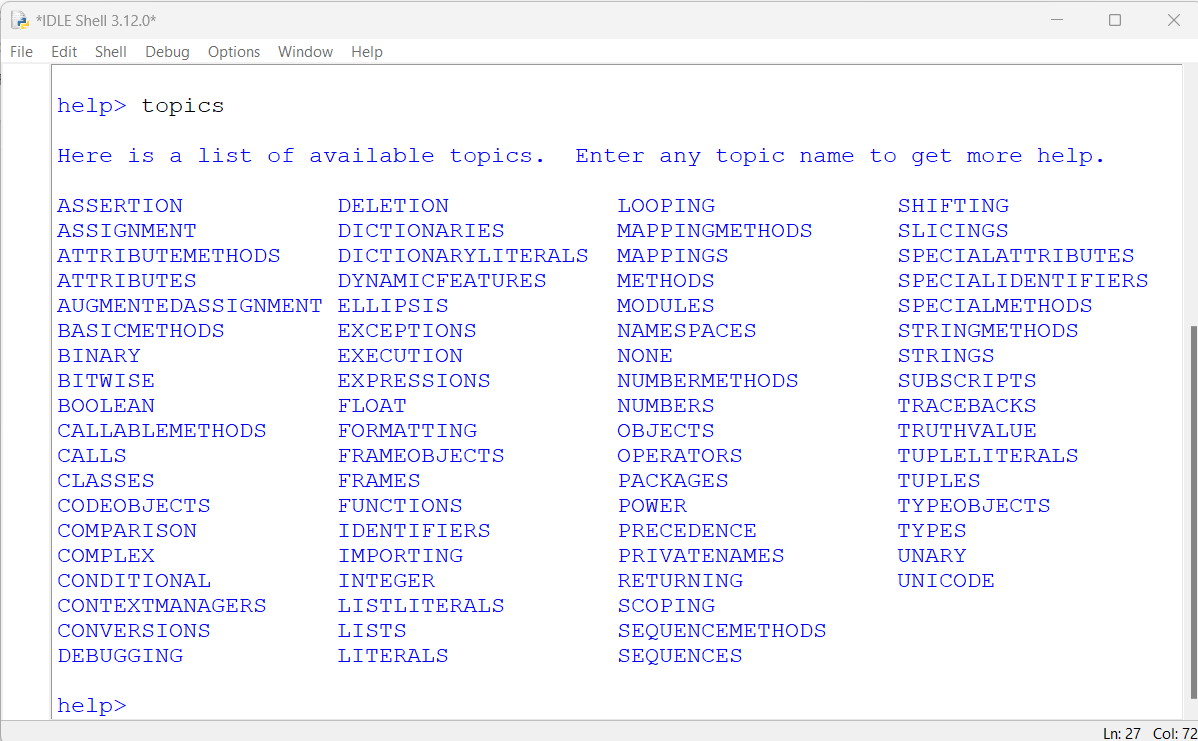


Now we can type in the name of a Python object to get helpful information about it:

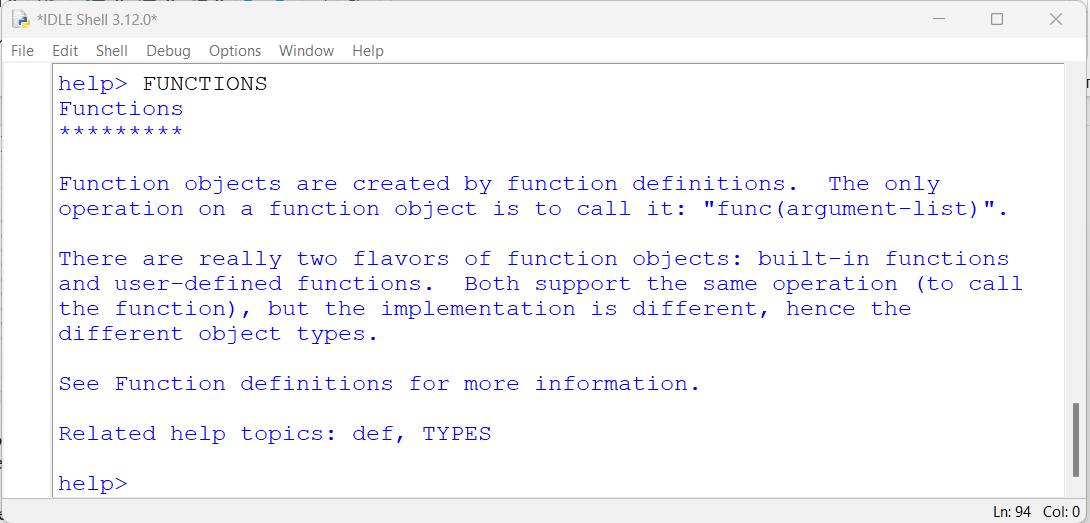
When we type the name len at the help> prompt and hit Enter, we get help content related to that built-in function.



We can type ‘topics’ to know about topics in python. Lets enter topics at help prompt.



Among the topics , suppose we want to know about functions, we should enter ‘FUNCTIONS’ in capital letters since the FUNCTIONS topic is shown in capital letters in help window.



To leave the content and get back to the help> prompt, you can press Q. To leave the help utility, you can type quit and hit Enter.

We can also use help() with the name of an object as an argument to get information about that object:

Python

>>> help(dir)

Help on built-in function dir in module builtins:

dir(...)

dir([object]) -> list of strings

...

Here [dir()](https://realpython.com/python-scope-legb-rule/#dir), we can use this function to inspect the methods and attributes that are available in a particular object:

Python

>>> dir(str)

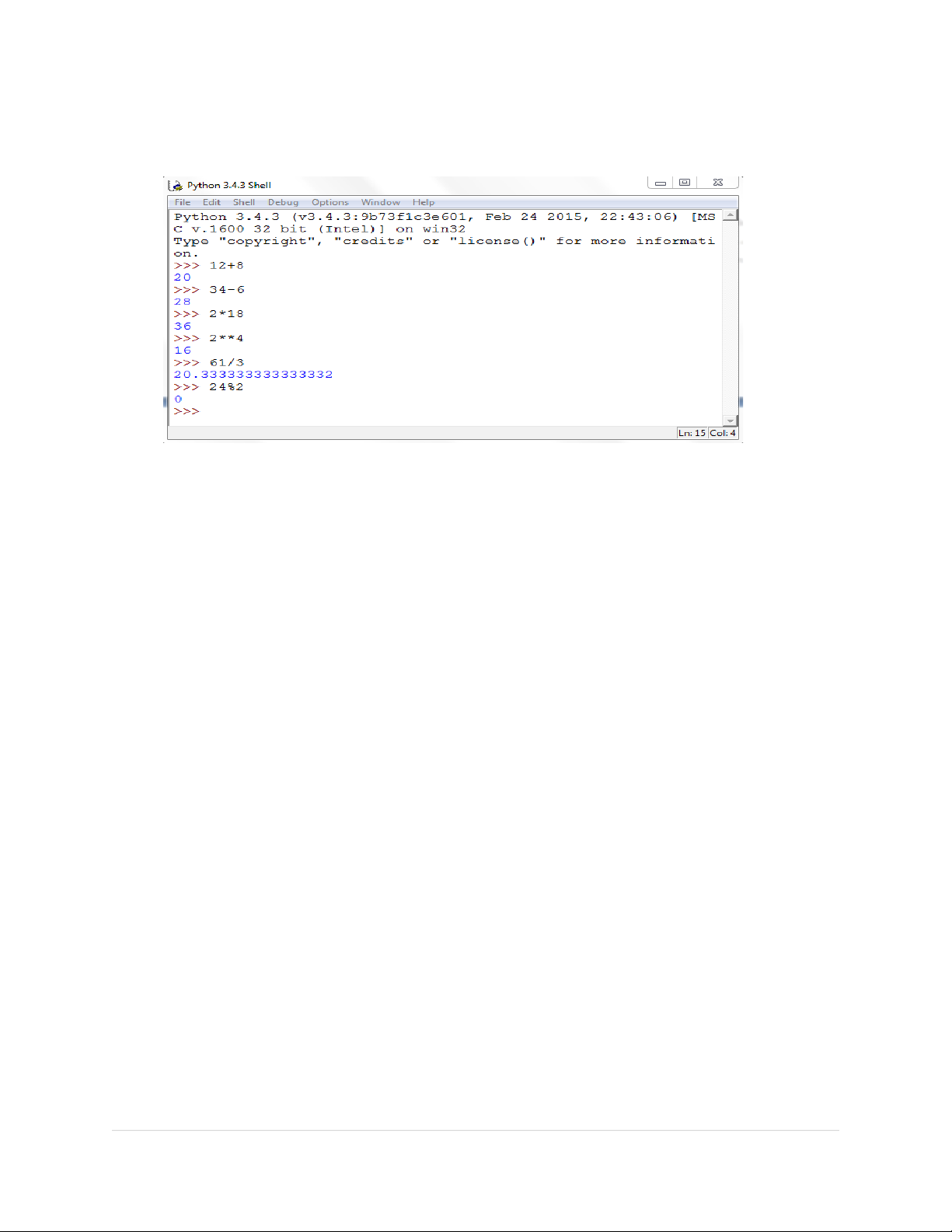
['\_\_add\_\_', '\_\_class\_\_', ..., 'title', 'translate', 'upper', 'zfill']

>>> dir(tuple)

['\_\_add\_\_', '\_\_class\_\_', ..., 'count', 'index']

When we call dir() with the name of a Python object as an argument, the function attempts to return a list of valid attributes for that specific object. This is a convenient way to get an idea of what a given object can do.

1. **Start Python interpreter and use it as Calculator**



1. **I) Write a program to calculate compound interest when principal, rate and number of periods are given**

The formula to calculate compound interest annually is given by:

*A = P(1 + R/100) t*

*Compound Interest = A – P*

*Where,*

* *A is amount*
* *P is the principal amount*
* *R is the rate and*
* *T is the time span*

**# Calculates compound interest**

**principal=int(input(“Enter the Principal amount:”))**

**rate=int(input(“Enter the Rate:”))**

**time=int(input(“Enter the time span:”))**

**Amount =principal \*((1+rate /100)\*\* time)**

**CI =Amount -principal**

**print("Compound interest is", CI)**

**3. II) Given coordinates (x1, y1), (x2, y2) find the distance between two points**

Calculate the distance between two points.

The distance formula derived from Pythagorean theorem.

The formula for distance between two point (x1, y1) and (x2, y2) is √((x2– x1)² + (y2– y1)²).

***Program to find distance between two points of coordinates***

**print(" To Find the distance between to (x1,y1),(x2,y2) coordinates")**

**x1=int(input("enter x1 value:"))**

**y1=int(input("enter y1 value:"))**

**x2=int(input("enter x2 value:"))**

**y2=int(input("enter y2 value:"))**

**pnts1=(x1,y1)**

**pnts2=(x2,y2)**

**print("\n you are given two coordinates are:",pnts1,'&',pnts2)**

**Distance=(((pnts1[0]-pnts2[0])\*\*2)+((pnts1[1]-pnts2[1])\*\*2))\*\*0.5**

**print("The Distance Between two coordiantes are",Distance)**

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

# read name, contact, email and birthday from user and print them

name = input("Enter your name: ")

address = input("Enter your Address: ")

email = input("Enter your email: ")

phone\_number = input("Enter your Phone Number: ")

print("Name: ", name)

print("Address: ", address)

print("Email: ", email)

print("Phone Number: ", phone\_number)

**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**

**num=int(input("Please Enter the Number of Rows"))**

**for i in range(1,num+1):**

**for j in range(1,i+1):**

**print(num,end=' ')**

**num=num-1**

**print()**

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

**Program-1:**

**ch=input("Please Enter any Character: \t")**

**if (ch >= 'A' and ch <= 'Z'):**

**print(ch,"is an UpperCase character");**

**elif (ch >= 'a' and ch <= 'z'):**

**print(ch,"is an LowerCase character");**

**elif(ch>='0' and ch<='9'):**

**print(ch,"is a digit")**

**else:**

**print(ch,"is not an alphabetic character");**

**Program-2:**

**ch=input("Please Enter any Character: \t")**

**if(ch>='0' and ch<='9'):**

**print(ch,"is a digit")**

**elif (ch.isUpper):**

**print(ch,"is an UpperCase character");**

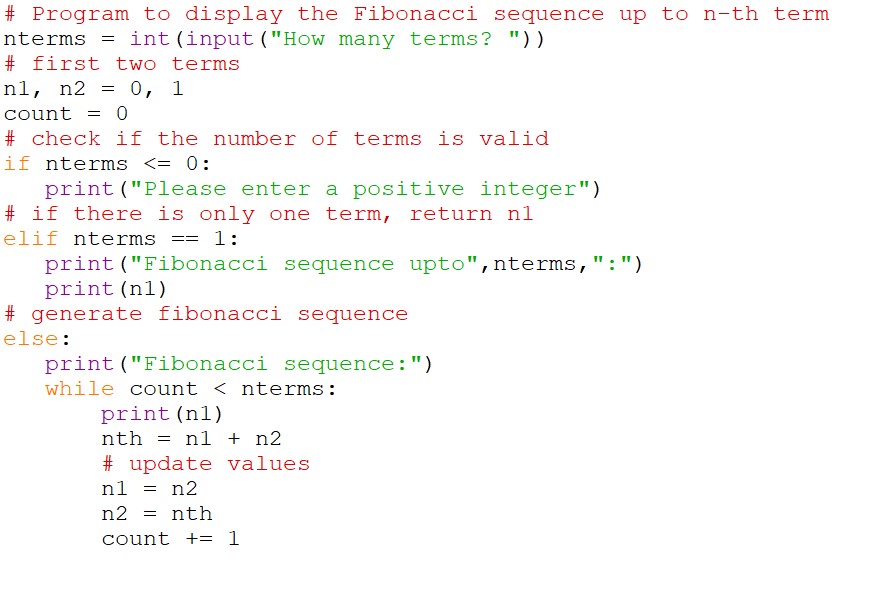
**elif (ch.isLower):**

**print(ch,"is an LowerCase character");**

**else:**

**print(ch,"is not an alphabetic character");**

1. **Python Program to Print the Fibonacci sequence using while loop**



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

# First, we will take the input:

lower\_value = int(input ("Please, Enter the Lowest Range Value: "))

upper\_value = int(input ("Please, Enter the Upper Range Value: "))

print ("The Prime Numbers in the range are: ", lower\_value,’and’, upper\_value)

for number in range (lower\_value, upper\_value + 1):

# all prime numbers are greater than 1,if number is less than or equal to 1, it is not prime

if number> 1:

for i in range (2, number):

# Check for factors

if (number % i) == 0:

# not a prime number so break inner loop and look for next number

break

else:

print (number)

**WEEK-3**

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

**import numpy as np**

**my\_list = [1, 2, 3, 4, 5, 6, 7, 8]**

**print("List to array: ")**

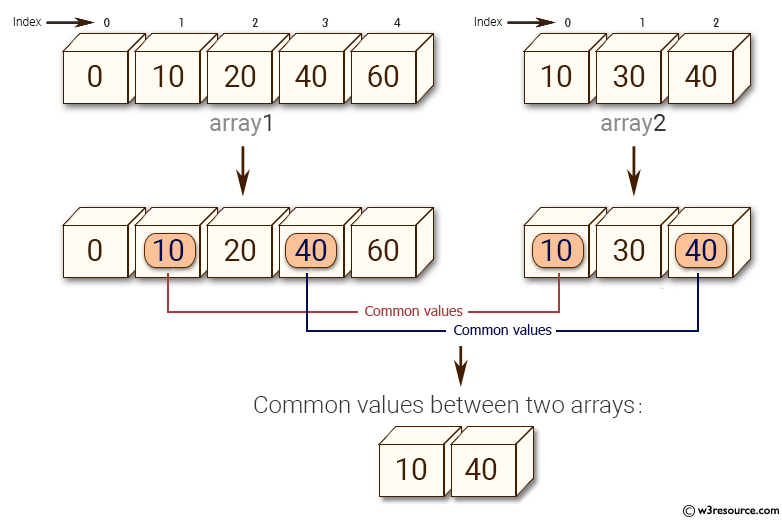
**print(np.asarray(my\_list))**

**my\_tuple = ([8, 4, 6], [1, 2, 3])**

**print("Tuple to array: ")**

**print(np.asarray(my\_tuple))**

1. **Write a program to find common values between two arrays.**

****

import numpy as np

array1 = np.array([0,10,20,40,60])

print("Array1: ",array1)

array2 =[10,30,40]

print("Array2: ",array2)

print("Common values between two arrays:")

print(np.intersect1d(array1, array2))

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

**Program-1**

**# Python code to demonstrate naive**

**# method to compute gcd ( recursion )**

**def gcd(a, b):**

**if(b == 0):**

**return a**

**else:**

**return gcd(b, a % b)**

**# Driver program to test above function**

**m = int(input("Enter a Number: "))**

**n = int(input("Enter b Number: "))**

**print("The gcd of {0} and {1} is : ".format(m,n),end=' ')**

**print(gcd(m, n))**

**Program-2**

**# Recursive function to return gcd of a and b**

**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)**

**# Driver program to test above function**

**m = int(input ("Please, Enter the First Value: "))**

**n = int(input ("Please, Enter the Second Value: "))**

**if(gcd(m, n)):**

**print('GCD of', m, 'and', n, 'is', gcd(m, n))**

**else:**

**print('not found')**

1. **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.**

**def palindrome(str):**

**for i in range(0, int(len(str)/2)):**

**if str[i] != str[len(str)-i-1]:**

**return False**

**return True**

**# main function**

**s = input("Enter a String: ")**

**ans =palindrome(s)**

**if (ans):**

**print("TRUE ")**

**else:**

**print("FALSE")**

**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.**

**def is\_sorted(list):**

**x=list[:]**

**x.sort()**

**if(list==x):**

**return True**

**else:**

**return False**

**n = int (input ("Enter number of elements in the List: "))**

**# Declare an empty list**

**list1 = []**

**# Iterate for n times take inputs**

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

**# Take input of ith element as x.**

**print(i+1,end=' ')**

**x = int(input(" Element"))**

**# Append 'x' to the list1.**

**list1.append(x)**

**print("Elements are : ",list1)**

**print(is\_sorted(list1))**

1. **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.**

**def has\_duplicate(list):**

**for i in range(len(list)):**

**for j in range(0, i):**

**if list[i] == list[j]:**

**return 'true'**

**return 'false'**

**n = int (input ("Enter number of elements: "))**

**# Declare an empty list**

**list1 = []**

**# Iterate for n times take inputs**

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

**# Take input of Ith element as x.**

**print(i+1,end=' ')**

**x = int(input(" Element"))**

**# Append 'x' to the list1.**

**list1.append(x)**

**print("Elements are : ",list1)**

**print(has\_duplicate(list1))**

**2.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.**

**def remove\_duplicates(L):**

**LS = sorted(L)**

**res = []**

**prev = None**

**for el in LS:**

**if el != prev:**

**res.append(el)**

**prev = el**

**return res**

**n = int (input ("Enter number of elements: "))**

**# Declare an empty list**

**L = []**

**# Iterate for n times take inputs**

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

**# Take input of ith element as x.**

**print(i+1,end=' ')**

**x = int(input(" Element"))**

**# Append 'x' to the list1.**

**L.append(x)**

**print("Elements are : ",L)**

**print("\n ",remove\_duplicates(L))**

2.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.

2. 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

1. 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**

**# A basic code for matrix input from user**

**R = int(input("Enter the number of rows:"))**

**C = int(input("Enter the number of columns:"))**

**# Initialize matrix**

**matrix = []**

**print("Enter the entries row wise:")**

**# For user input**

**for i in range(R): # A for loop for row entries**

**a =[]**

**for j in range(C): # A for loop for column entries**

**a.append(int(input()))**

**matrix.append(a)**

**# For printing the matrix**

**for i in range(R):**

**for j in range(C):**

**print(matrix[i][j], end = "")**

**print()**

1. **II) Write a python program to perform multiplication of two square matrices**

**matOne = []**

**print("Enter 9 Elements for First Matrix: ")**

**for i in range(3):**

**matOne.append([])**

**for j in range(3):**

**num = int(input())**

**matOne[i].append(num)**

**matTwo = []**

**print("Enter 9 Elements for Second Matrix: ")**

**for i in range(3):**

**matTwo.append([])**

**for j in range(3):**

**num = int(input())**

**matTwo[i].append(num)**

**# For printing the matrix**

**print("First Matrix: ")**

**for i in range(3):**

**for j in range(3):**

**print(matOne[i][j], end = "")**

**print()**

**# For printing the matrix**

**print("Second Matrix: ")**

**for i in range(3):**

**for j in range(3):**

**print(matTwo[i][j], end = "")**

**print()**

**matThree = []**

**for i in range(3):**

**matThree.append([])**

**for j in range(3):**

**matThree[i].append(matOne[i][j]+matTwo[i][j])**

**print("\nAddition of Two Given Matrix is:")**

**for i in range(3):**

**for j in range(3):**

**print(matThree[i][j], end="")**

**print()**

**.III) Write a python program to perform multiplication of two square matrices**

**matOne = []**

**print("Enter 9 Elements for First Matrix: ")**

**for i in range(3):**

**matOne.append([])**

**for j in range(3):**

**num = int(input())**

**matOne[i].append(num)**

**matTwo = []**

**print("Enter 9 Elements for Second Matrix: ")**

**for i in range(3):**

**matTwo.append([])**

**for j in range(3):**

**num = int(input())**

**matTwo[i].append(num)**

**# For printing the matrix**

**print("First Matrix: ")**

**for i in range(3):**

**for j in range(3):**

**print(matOne[i][j], end = "")**

**print()**

**# For printing the matrix**

**print("Second Matrix: ")**

**for i in range(3):**

**for j in range(3):**

**print(matTwo[i][j], end = "")**

**print()**

**matThree = []**

**for i in range(3):**

**matThree.append([])**

**for j in range(3):**

**sum = 0**

**for k in range(3):**

**sum = sum + (matOne[i][k] \* matTwo[k][j])**

**matThree[i].append(sum)**

**print("\nMultiplication of Two Given Matrix is:")**

**for i in range(3):**

**for j in range(3):**

**print(matThree[i][j], end="")**

**print()**

1. **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.**
2. **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 anddraws a representation of the Rectangle on the Canvas.

b. Add an attribute named color to your Rectangle objects and modify draw\_rectangle

so that ituses 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 arepresentation 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

multiplelevels of Inheritances.

3. Write a python code to read a phone number and email-id from the user and

validate it forcorrectness.

**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 init 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, blankspaces, 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 buttonsas Submit and Reset.