**PYTHON**

* **Installation:**
  + Download Python from its official website.
  + Download plugin Python by Microsoft in VS Code.
* **Key Points for arithmetic:**
  + 22/7 = 3.14……
  + 22//7 = 3 (floor division. Discards numbers after decimal)
* **There are no characters. Only Strings are available**
  + To add “ ” in strings, write
    - print(‘Hello “World”’) Output: Hello “World”
* **Variables and Arithmetic:**
  + To check the type of variables, type(variablename)
  + To get exponential, var1\*\*var2
* To take input from users, var=input(). Input always returns a string.
* There are no {} for functions or loops in python.
* **String:**
  + .format function:
    - It is used to format a string.
      * msg=var1+’ ‘+var2+’. How are you?’ can be written as
        + msg='{} {}. How are you?'.format(var1,var2)
        + msg=f'{var1} {var2}. How are you?' (f function)
  + To lower case – msg.lower()
  + To upper case – msg.upper()
  + To get the length – msg.\_\_len\_\_() or can also use len(msg)
  + To count how many times a character repeats, msg.count(‘character’)
  + To check if the word exists, msg.find(‘word’). Returns index.
* **Functions:**
  + Syntax:
    - def function\_name():
* **Modules:** 
  + Files can be created as a modules so that it can be shared. (Packages in flutter).
  + If the name of module file is “modules.py”
  + It can be imported as
    - import modules
    - To call the functions, module\_name.function\_name()
  + It can also be imported as
    - from module\_name import \*
    - To call the function, function\_name()
* **Conditional Statements:**
  + **If-else**

name = input("Please enter your name: ")

age = int(input("{}, Can you also enter your age: ".format(name)))

if age < 18:

    print("\n{}, sorry!! You cannont vote.".format(name))

else:

    print("\nGreat!! {}. You can vote.".format(name))

* + **If-elif-else**

if age < 18:

    print("\n{}, sorry!! You cannont vote.".format(name))

elif age > 18 and age < 21:

    print("\nGreat!! {}. You can vote because your age is {}".format(name, age))

else:

    print("\nGreat!! {}. You can vote.".format(name))

* **break and continue:**
  + Same syntax as C++/Java
* **For Loop:**
  + Syntax:
    - for i in range(1,20):

print(“Hello”)

* + - Here, 1 is the starting index and 20 is the ending index.
    - If we don’t provide starting index, by default is 0
    - If we want to provide how much ‘i’ should be incremented after each iterations, we can provide as
      * for i in range (1,20,3) where 3 is the incrementation
      * If we use this, we will have to compulsory use the starting index.
  + If we print the value of i as:
    - print(‘Value of i is “+i), it will give error. Concatenation is possible only for string. So, print(‘Value of i is “+str(i))
  + **To print pattern:**

for i in range(0, 10):

    for j in range(0, i):

        print('\*', end=" ")

    print('\r')

**\r moves to the next line and \n leaves a line**

**end is used to remove the next line character and end the output with a custom character.**

* **To check if a char/number is in a particular range,** 
  + If number in ‘0123456789’
* **while loop**
  + Syntax:

a = int(input("Enter a number: "))

b = 0

while b < a:

    print(b)

    b += 1

**there’s no ++ or -- operator in Python**

* **Lists**
  + A type of array but can store different type of data types.
  + Syntax:
    - list\_name = [1,’2’,’a’,3.14,True,None]
  + Slicing is the process of displaying or accessing a sublist from a list,
    - list\_name[starting\_index : ending\_index]
  + To access the list from the last index, you can use -ve. To access last element, list\_name[-1]
  + If two lists have same data but different order, both lists are counted as different
  + Multi-Dimensional List:

list\_name = [[1, 2, 3], ['a', "Abhishek"], [3.14], [True, None]]

for i in list\_name:

    print(i)

* + **Inserting into list**

spam = ['cat', 'rat', 'bat']

spam.append('mat')

spam.insert(0, 'mat')

print(spam)

append adds the item at the end of the list and insert adds the item at the specified position.

* + **Removing from list –** spam.remove(‘cat’)
  + **Sorting the list –** spam.sort()
* **Iterators:**
  + Iterators are known as data streams from which data can be extracted one by one.

string = '0123456789'

for i in iter(string):

    print(i)

days = ['Mon', 'Tue', 'Wed', 'Thurs', 'Fri', 'Sat', 'Sun']

days\_iter = iter(days)

for char in range(0, len(days)):

    next\_day = next(days\_iter)

    print(next\_day)

* **Dictionaries:**
  + To store different types of data, list is not normally used.
  + Dictionary is used to store different types of data.
  + Values in dictionary is stored in key-value pair
  + Two dictionaries with same values but different orders are counted as equal.
  + Syntax:

myCat = {'size': 'fat', 'color': 'black'}

print(myCat['size'])

* + If you want to print only values,

details = {'First Name': 'Abhishek', 'Surname': 'Doshi'}

for i in details.values():

    print(i)

* + If you want to print only key values, use details.keys()
  + To display both, use details.items()
  + Example:

details = {'Abhishek': '26/01/1999', 'Kush': '07/05/2003'}

while True:

    name = input("Enter Name: ")

    if name in details:

        print('Birthdate: '+details[name])

        break

    else:

        print('Name not in the list!!\n')

        bday = input('Enter the birthdate: ')

        details[name] = bday

        print("List Updated\n\n")

**Numpy**

* **Installation:**
  + Go to C:\Users\adosh\AppData\Local\Programs\Python\Python38\Scripts.
  + Open CMD
  + Write pip install numpy
* **Numpy is basically used for mathematical calculations especially related to arrays and matrices. Numerical Python Library.**
* **np.arrange(starting\_index,ending\_index)**
  + This gives an array that contains numbers starting from starting\_index and ending to ending\_index
  + To get even number np.arrange(0,10,2) where 2 is the number of incrementation.
* To generate 0 matrix, np.zeros(index) where index is the dimension of array.
* To generate m\*n matrix, np.zeros((m,n))
* To generate 1 matrix, np.ones(index)
* To generate random numbers:
  + np.random.randint(start,end)
  + To generate array of random numbers:
    - np.random.randint(start,end,(m\*n))
  + To generate constant random numbers:

np.random.seed(1)

print(np.random.randint(0, 100, 10))

where 10 is the number of random numbers required.

* To get maximum of the array, array\_name.max()
* To get minimum, array\_name.min()
* To get average from array, array\_name.mean()
* To get the index of max value, array\_name.argmax()
* To get the index of min value, array\_name.argmin()
* To reshape an array into a matrix, eg from 1\*1 to 2\*5, use array\_name.reshape(m\*n)
* For getting a specific range of matrix, array\_name[start1:end1,start2:end2]

**MatPlotLib**

* Matplotlib is a library that is used to plot graphs in python.
* **Installation:**
  + Go to C:\Users\adosh\AppData\Local\Programs\Python\Python38\Scripts
  + Open cmd here.
  + Write pip install matplotlib
* **Importing matplotlib:**
  + import matplotlib.pyplot as plt
  + where matplotlib is the module and pyplot is the class inside matplotlib
* **Example:**

import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5, 6]

y = [2, 9, 8, 10, 20, 30]

plt.plot(x, y)

plt.xlabel('x axis')

plt.ylabel('y axis')

plt.title('First Plot')

plt.show()

* + Here, **plt.plot()** takes up the data for x and y axis
  + **plt.xlabel()** sets the label text for x axis
  + **plt.ylabel()** sets the label text for y axis
  + **plt.title** sets the title of the plot
  + **plt.show()** makes the graph visible.
* **Matplotlib and Numpy:**

import numpy as np

import matplotlib.pyplot as plt

time = np.arange(0.0, 2.0, 0.01)

s = 1 + np.cos(2\*np.pi\*time)

plt.plot(time, s,)

plt.grid()

plt.xlabel('Time')

plt.ylabel('Voltage (mV)')

plt.title('Cosine Wave Plot (cos(x))')

plt.show()

* + **plt.grid()** provides grid in the background
* **Subplot**
  + Making multiple plots in same plane

import numpy as np

import matplotlib.pyplot as plt

x1 = np.linspace(0.0, 5.0)

x2 = np.linspace(0.0, 2.0)

y1 = np.cos(2\*np.pi\*x1)\*np.exp(-x1)

y2 = np.cos(2\*np.pi\*x2)

plt.subplot(2, 1, 1)

plt.plot(x1, y1, 'o-')

plt.title('Subplot 1')

plt.xlabel('x1')

plt.ylabel('y1')

plt.subplot(2, 1, 2)

plt.plot(x2, y2, '--')

plt.title('Subplot 2')

plt.xlabel('x2')

plt.ylabel('y2')

plt.show()

* + **np.linspace()** provides a linear range starting from 0
  + **plt.subplot(row,column,index)** defines where to plot the subplots