

**Lab Report-03**

**Course Title: Artificial Intelligence**

**Course Code:** **CSE 366**

**Semester: Fall 2021**

**Section No:** **01**

**Submitted By**

**Name: Syeda Tamanna Sheme**

**ID:2018-2-60-010**

**Submitted To**

**Md Al-Imran**

**Lecturer**

**Department of Computer Science & Engineering**

**Date of Submission:** **20 November, 2021**

# Theory

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). Python is named after a TV Show called ëMonty Pythonís Flying Circusí and not after Python-the snake.

Python 3.0 was released in 2008. Although this version is supposed to be backward incompatibles, later on many of its important features have been backported to be compatible with version 2.7.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable.

**#1  
x\_val = [1, 2, 3]  
for x in x\_val: ##Looping without indices  
 print(x)**

**Codes**

### **Output**

**1**

**2**

**3**

**#2  
for i in range(len(x\_val)): ##Looping with indices  
 print(x\_val[i])**

**Output**

**1**

**2**

**3**

**#3  
cities = ('Dhaka', 'Tokyo', 'Seoul', 'Tehran', 'Doha')  
countries = ('BD', 'JP', 'SK', 'IR', 'QR')  
for city, country in zip(cities, countries):  
 print(f'The city is {city} and corresponding country {country}')**

**Output**

**The city is Dhaka and corresponding country BD**

**The city is Tokyo and corresponding country JP**

**The city is Seoul and corresponding country SK**

**The city is Tehran and corresponding country IR**

**The city is Doha and corresponding country QR**

**#4**

**for index, number in enumerate(x\_val):  
 print(f'xa\_val[{index}] = {number}')**

**Output**

**xa\_val[0] = 1**

**xa\_val[1] = 2**

**xa\_val[2] = 3**

**#5  
def f(x):  
 return x\*\*3  
print(f(3))**

**27**

**#6  
from scipy.integrate import quad  
print(quad(lambda x: x\*\*2, 0 , 3))**

**(9.000000000000002, 9.992007221626411e-14)**

**#7  
f = (lambda x: x\*\*3)(3)  
print(f)**

**27**

**#8  
import numpy as np**

**a = np.zeros(3, dtype = int)**

**print(a.shape)  
print(a)  
print(type(a))**

**[0 0 0]**

**<class 'numpy.ndarray'>**

**(3,)**

**#9  
import numpy as np  
a = np.zeros(10)  
print(a)  
print(type(a))  
print(a.shape)  
print(np.linspace(2, 4, 50))**

**[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]**

**<class 'numpy.ndarray'>**

**(10,)**

**[2. 2.04081633 2.08163265 2.12244898 2.16326531 2.20408163**

**2.24489796 2.28571429 2.32653061 2.36734694 2.40816327 2.44897959**

**2.48979592 2.53061224 2.57142857 2.6122449 2.65306122 2.69387755**

**2.73469388 2.7755102 2.81632653 2.85714286 2.89795918 2.93877551**

**2.97959184 3.02040816 3.06122449 3.10204082 3.14285714 3.18367347**

**3.2244898 3.26530612 3.30612245 3.34693878 3.3877551 3.42857143**

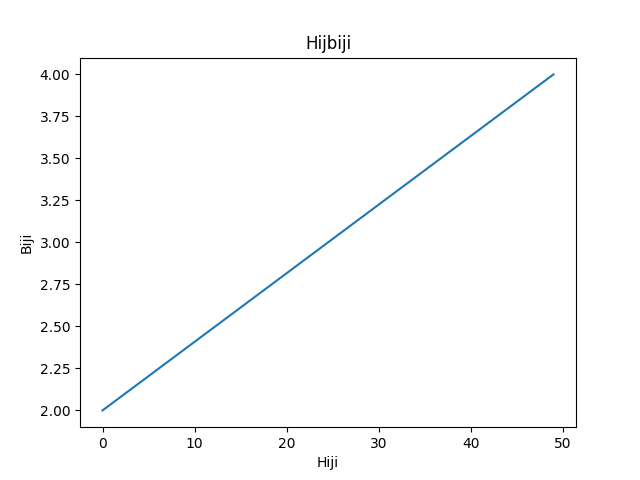
**3.46938776 3.51020408 3.55102041 3.59183673 3.63265306 3.67346939**

**3.71428571 3.75510204 3.79591837 3.83673469 3.87755102 3.91836735**

**3.95918367 4. ]**

**#10  
import matplotlib.pyplot as plt  
import numpy as np  
a = np.zeros(10)  
b= np.linspace(2, 4, 50)  
plt.plot(b)  
plt.title("Hijbiji")  
plt.xlabel("Hiji")  
plt.ylabel("Biji")  
plt.show()**

**Output**

****

**#11  
import matplotlib.pyplot as plt  
import numpy as np  
d = np.array((12, 16 , 14, 18), dtype = float)  
e = np.array((13, 17, 19, 21))  
print(d@e)**

**Output**

**1072.0**

**#12  
import matplotlib.pyplot as plt  
import numpy as np  
a = np.random.randn(5)  
print(a)**

**[0.26872293 0.30217011 0.69324888 0.30444703 0.7248253 ]**

**#13  
import matplotlib.pyplot as plt  
import numpy as np  
b = np.copy(a)  
print(b)**

**[ 0.10041079 -0.58889527 0.47072522 1.44076452 1.74354263]**

**#14  
import matplotlib.pyplot as plt  
import numpy as np  
  
a = np.sqrt(2 \* np.pi)  
print(a)**

**2.5066282746310002**

**#15  
import matplotlib.pyplot as plt  
import numpy as np  
x = np.array([1,2,3])  
b = np.sin(x)  
print(b)**

**[0.84147098 0.90929743 0.14112001]**

**#16  
import matplotlib.pyplot as plt  
import numpy as np  
def f(x):  
 return 1 if x > 0 else 0  
print(f(1))**

**1**

**#Exercise-01**

**"""Find root of ax^2 + bx + c, consider fixed x value but take multiple values for co-efficient. Hints: def f(x, co-efficient), co-efficient = (2, 1)."""**

**import math  
# function for finding roots  
def findRoots(a, b, c):  
  
 dis\_form = b \* b - 4 \* a \* c  
 sqrt\_val = math.sqrt(abs(dis\_form))  
  
  
 if dis\_form > 0:  
 print(" real and different roots ")  
 print((-b + sqrt\_val) / (2 \* a))  
 print((-b - sqrt\_val) / (2 \* a))  
  
 elif dis\_form == 0:  
 print(" real and same roots")  
 print(-b / (2 \* a))  
  
 else:  
 print("Complex Roots")  
 print(- b / (2 \* a), " + i", sqrt\_val)  
 print(- b / (2 \* a), " - i", sqrt\_val)  
  
  
a = int(input('Enter a:'))  
b = int(input('Enter b:'))  
c = int(input('Enter c:'))  
  
# If a is 0, then incorrect equation  
if a == 0:  
 print("Input correct quadratic equation")  
  
else:  
 findRoots(a, b, c)**

**Output**

**Enter a:7**

**Enter b:5**

**Enter c:2**

**Complex Roots**

**-0.35714285714285715 + i 5.5677643628300215**

**-0.35714285714285715 - i 5.5677643628300215**

**#Exercise-02**

**"""Write a function in Python which takes two sequences as arguments and returns True if every element in a sequence is also an element of second sequence, else False."""  
  
def common\_data(list1, list2):  
 result = False  
 for x in list1:  
 for y in list2:  
 if x == y:  
 result = True  
 return result  
 elif x != y:  
 result = False  
 return result  
print(common\_data([1,2,3,4,5], [1,2,3,4,5]))  
print(common\_data([1,2,3,4,5], [6,7,8,9,10]))**

**Output**

**True**

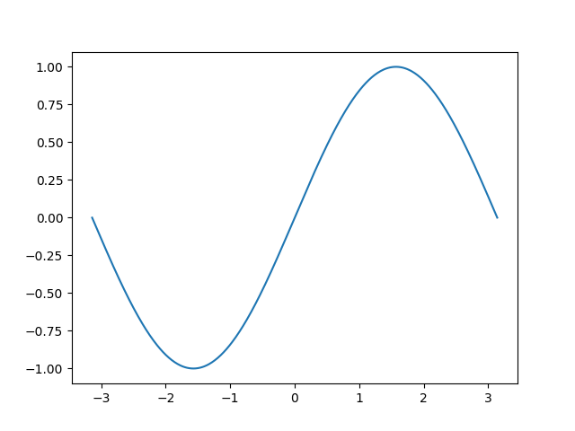
**False**

**#Exercise-03**

**"""Plot sin using np library. Use linspace to generate values."""""**

**#Exercise-03  
import matplotlib.pyplot as pl  
import numpy as np  
  
x = np.linspace(-np.pi, np.pi, 556, endpoint=True)  
y = np.sin(x)  
pl.plot(x,y)  
pl.show()**

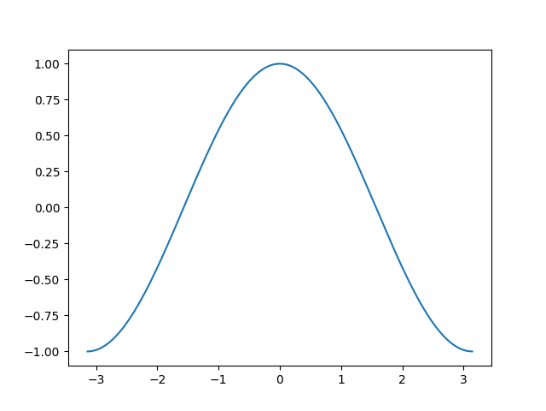
**Output**

****

**"""Plot cosine function using np library. Use linspace to generate values."""""**

**#Exercise  
import matplotlib.pyplot as pl  
import numpy as np  
  
x = np.linspace(-np.pi, np.pi, 556, endpoint=True)  
y = np.cos(x)  
pl.plot(x,y)  
pl.show()**

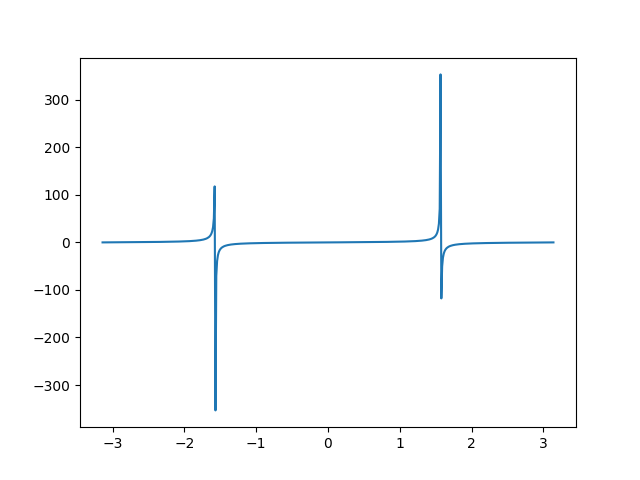
**Output**

****

**"""Plot tan function using np library. Use linspace to generate values."""""**

**#Exercise  
import matplotlib.pyplot as pl  
import numpy as np  
  
x = np.linspace(-np.pi, np.pi, 556, endpoint=True)  
y = np.tan(x)  
pl.plot(x,y)  
pl.show()**

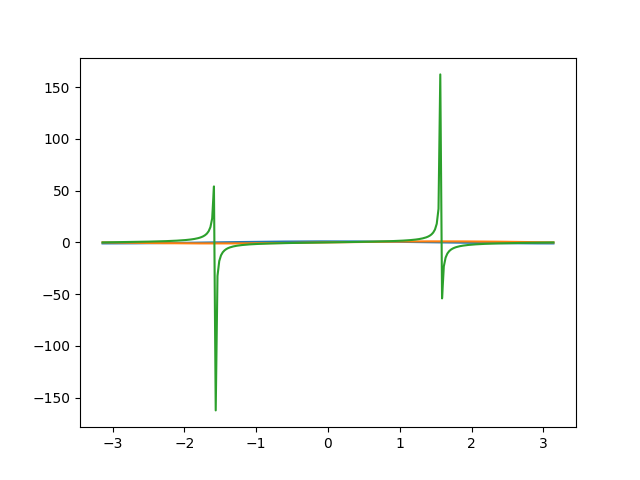
**Output**

****

**"""Plot sin, cosine, and tan function using np library. Use linspace to generate values."""""**

**#Exercise  
import matplotlib.pyplot as pl  
import numpy as np  
  
x = np.linspace(-np.pi, np.pi, 256, endpoint=True)  
  
y = np.cos(x)  
y1 = np.sin(x)  
z1 = np.tan(x)  
pl.plot(x,y)  
pl.plot(x, y1)  
pl.plot(x, z1)  
pl.show()**

**Output**

****

## **Results**

After doing the lab task and lab work we are able to learn -Function Python, arguments and

Ploting, Slicing, ‘in’ operator, looping and counting, Comparison operator, String Methods, parsing, Lists and operation, Variables, expressions, and statements Conditional Executions, Functions, Loops. Now we are able to do programs related to these topics. After doing the lab task, I did some problems related to my task. Now we are able to solve many problems.

## **Discussion**

Python is a general-purpose, versatile and popular programming language. It’s great as a first language because it is concise and easy to read, and it is also a good language to have in any programmer’s stack as it can be used for everything from web development to software development and data science applications.

This lab task is a great introduction to both fundamental programming concepts and the Python programming language. Python 3 is the most up-to-date version of the language with many improvements made to increase the efficiency and simplicity of the code that we write. Dayby-day, python new version are realising and all the new versions have new features and these new features are better than previous one. So, finally I can say that python will be more user friendly in future.