Industrial Internship Report on

” URL SHORTENER Project Using Pyhton”

Prepared by

K.ABDUL KALAM

Executive Summary

This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT).This internship was focused on a project/problem statement provided by UCT.

Description: The URL shortener is a Python project that converts long URLs into shorter, more manageable links. It takes a long URL as input, generates a unique shortened URL, and redirects users to the original URL when the shortened link is accessed.

Scope: The scope of this project involves designing a user interface to input long URLs and display the shortened links, implementing a database to store the mapping between original and shortened URLs, and developing functions to generate unique shortened URLs and handle redirection.

TABLE OF CONTENTS

1 Preface

1.1 Week 1:

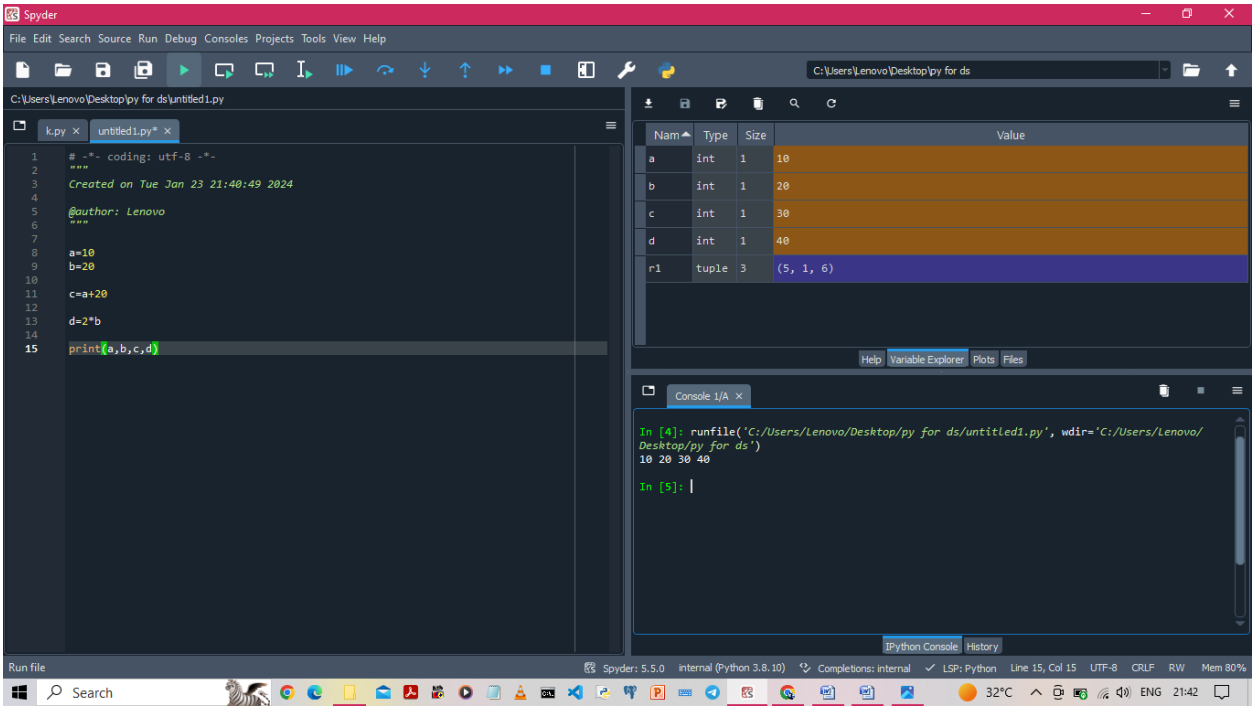
In Week 1 ,I Iearned about Basics Of Python. It is a highlevel programming language and it is easy to understand. It is open source and platform independent. It is developed by Guido Van Rossum.

Python Libraries used for providing key features for data Science. It also offers oop programming.

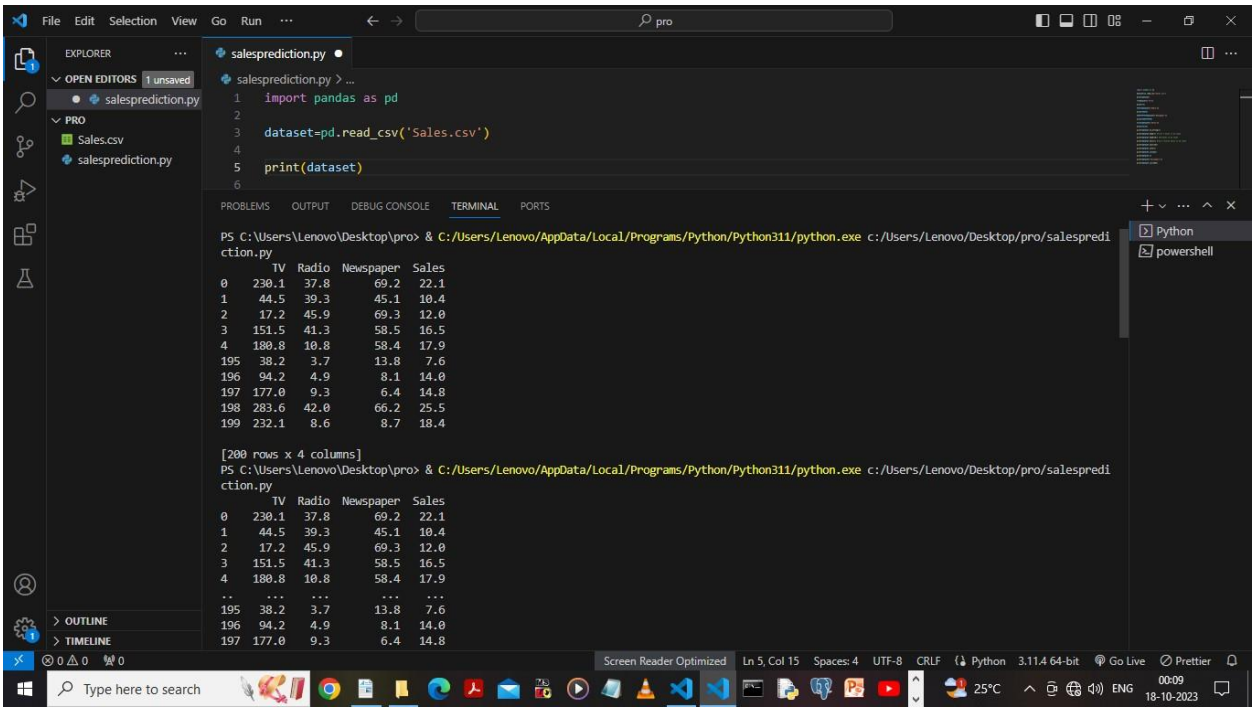
Python has a great library ecosystem.

In this I specifically works on some python libraries such as Numpy , Pandas and Matplotlib

I used Spyder IDE(Integrated Development Environment) for my coding exercises.



I have tried various code editors such as VScode for using python Libraries.



Pandas offers a varietyof functions for Data Wrangling and manipulations.

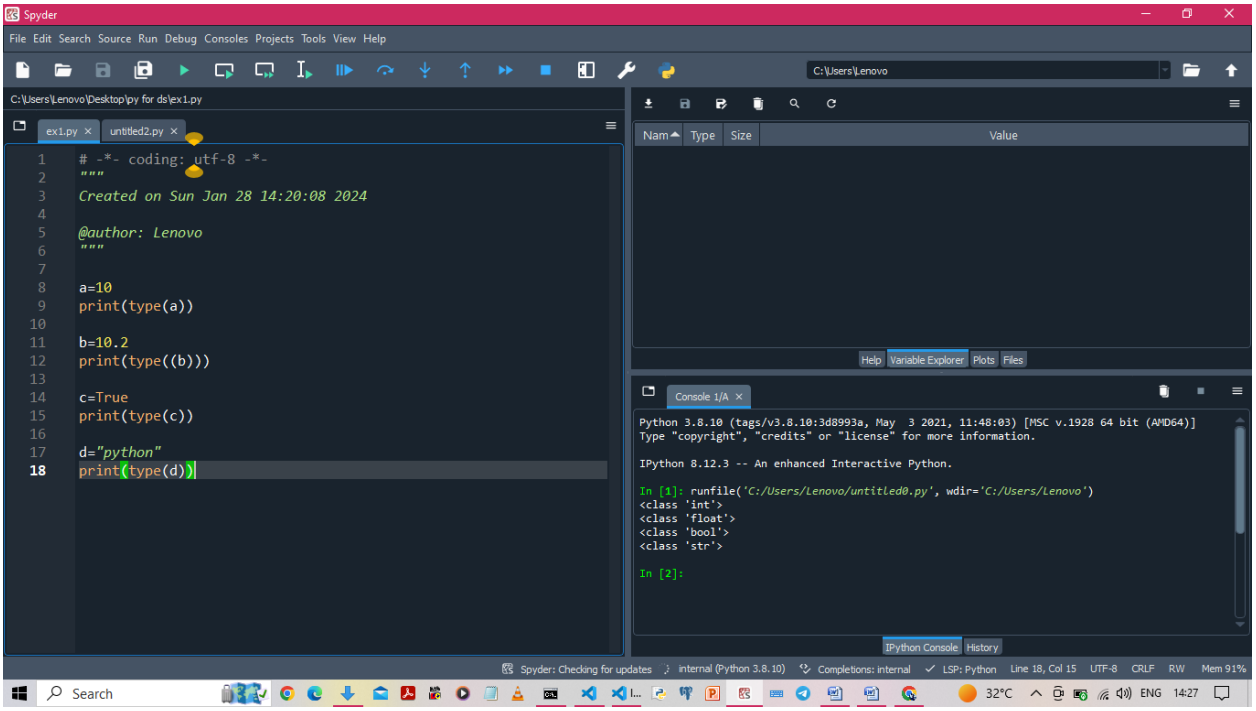
1.2 Week 2 :

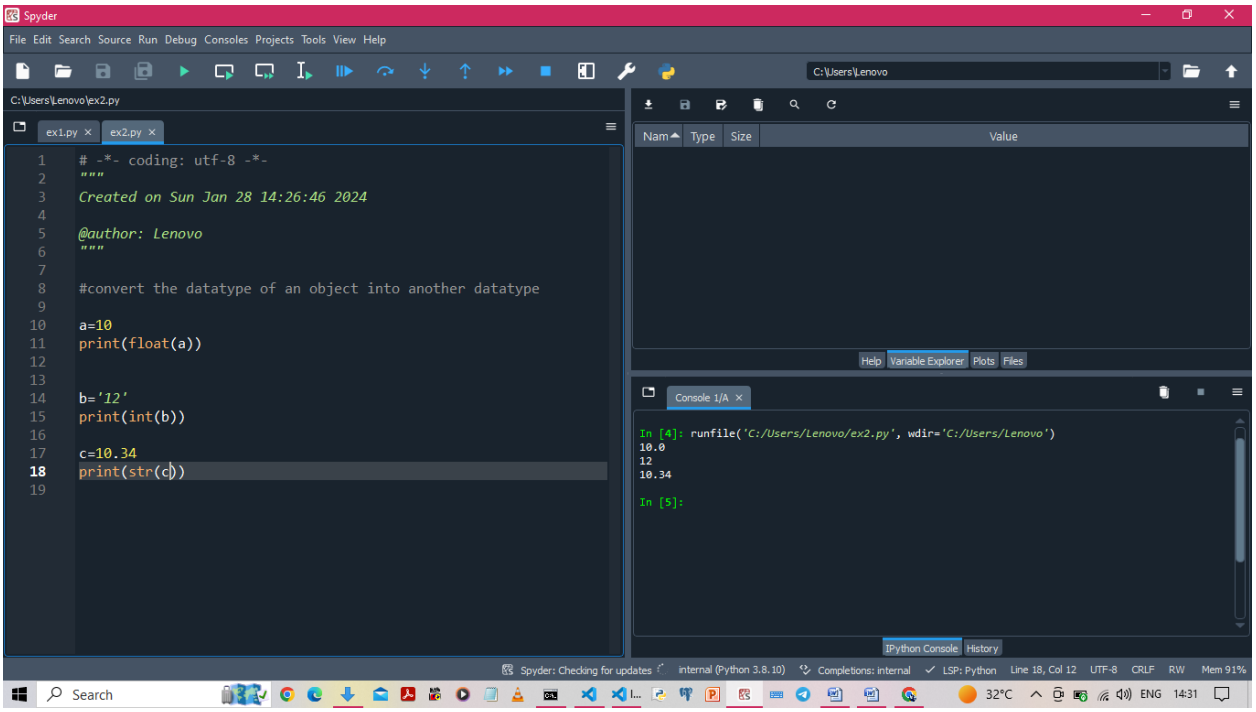
In week 2 I learned Data types, Conditional Statements, Looping Statements in Python. I also worked on these concepts.

Data types in python:

Commonly used data types in python are Boolean, Integer, Float, Complex and String.

I have Used Spyder IDE for practicing my code.





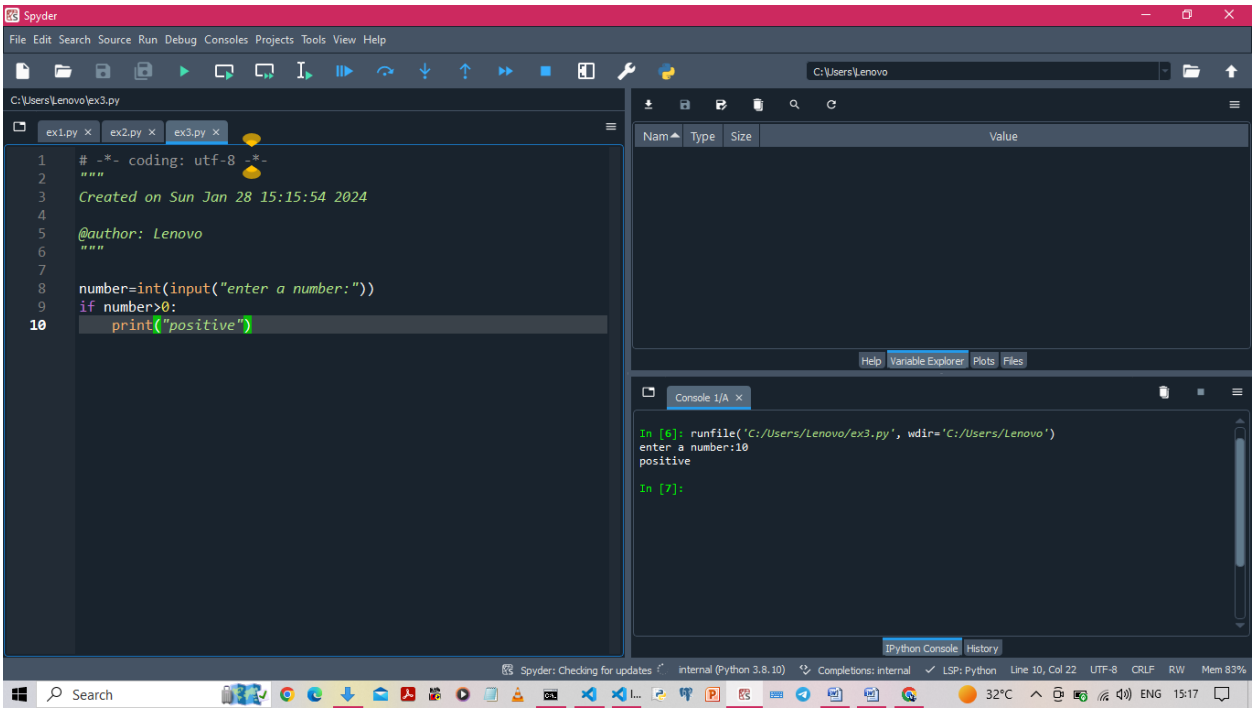
Conditional Statements in python:

Conditional statements are fundamental programming constructs that allow you to control the flow of your program based on conditions that you specify.

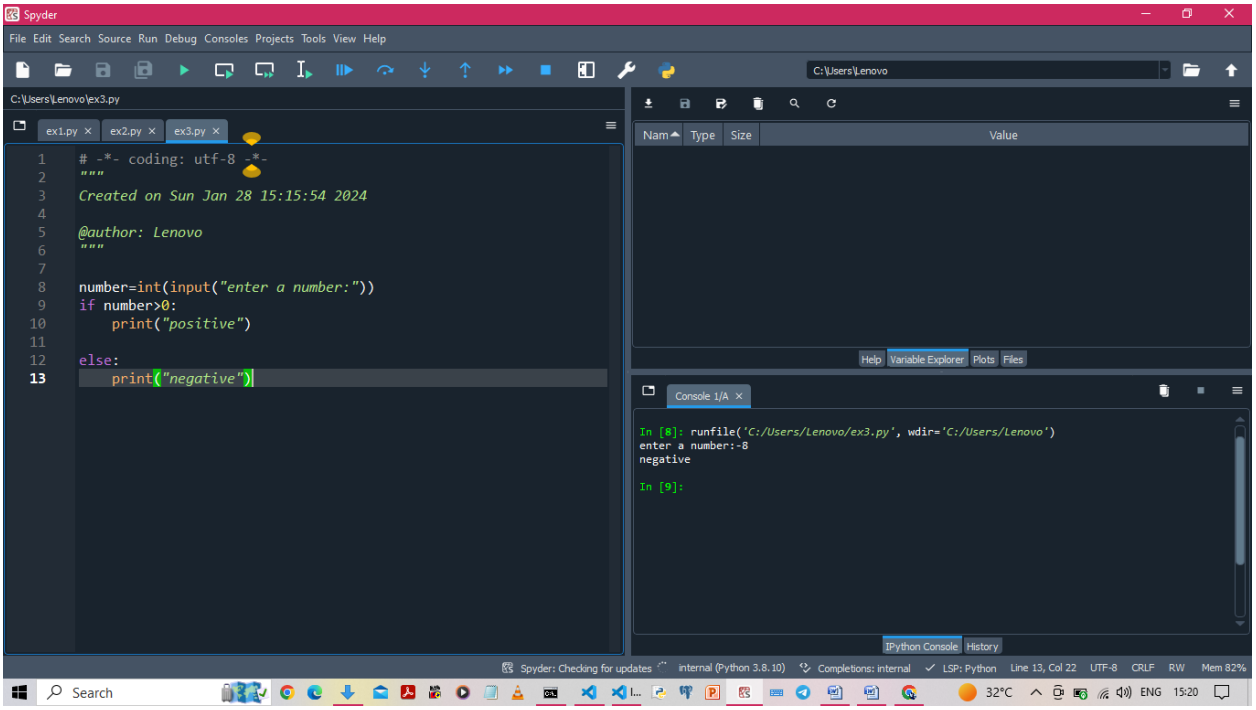
Types of Conditional Statements in python:

1. **if statement**
2. **if…else statement**
3. **if…elif…else statement**
4. **Nested if statement**

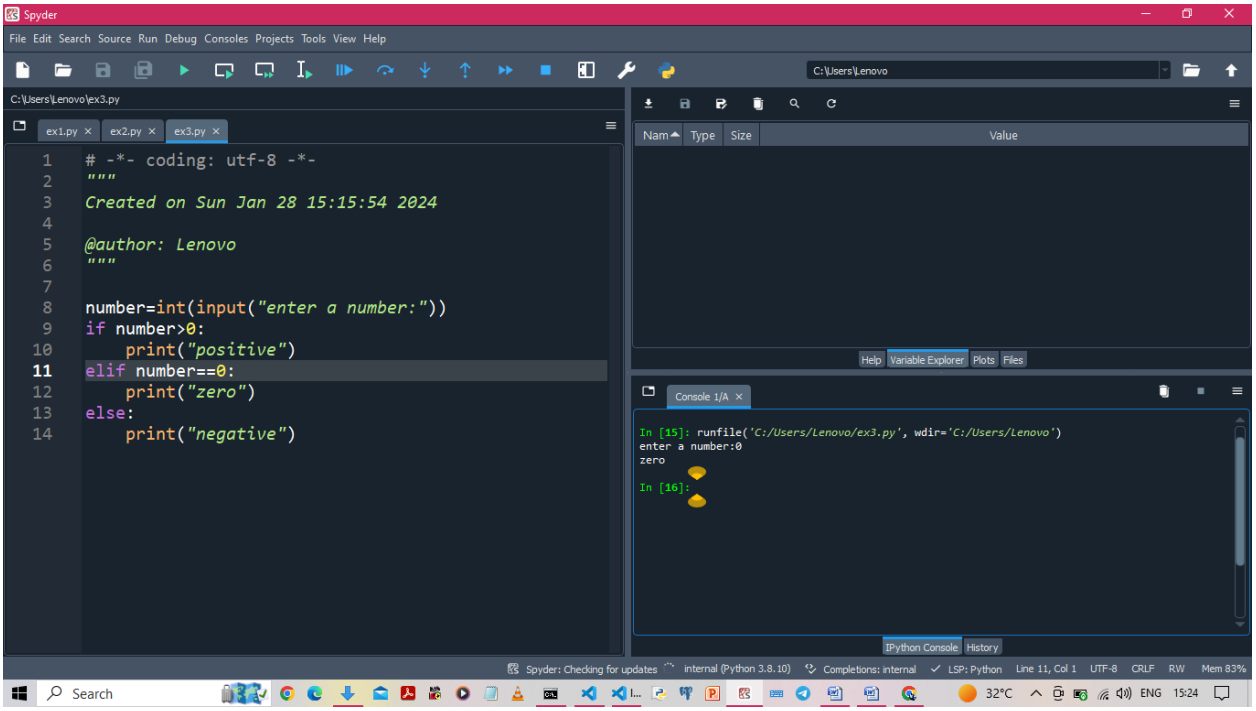
if statement :



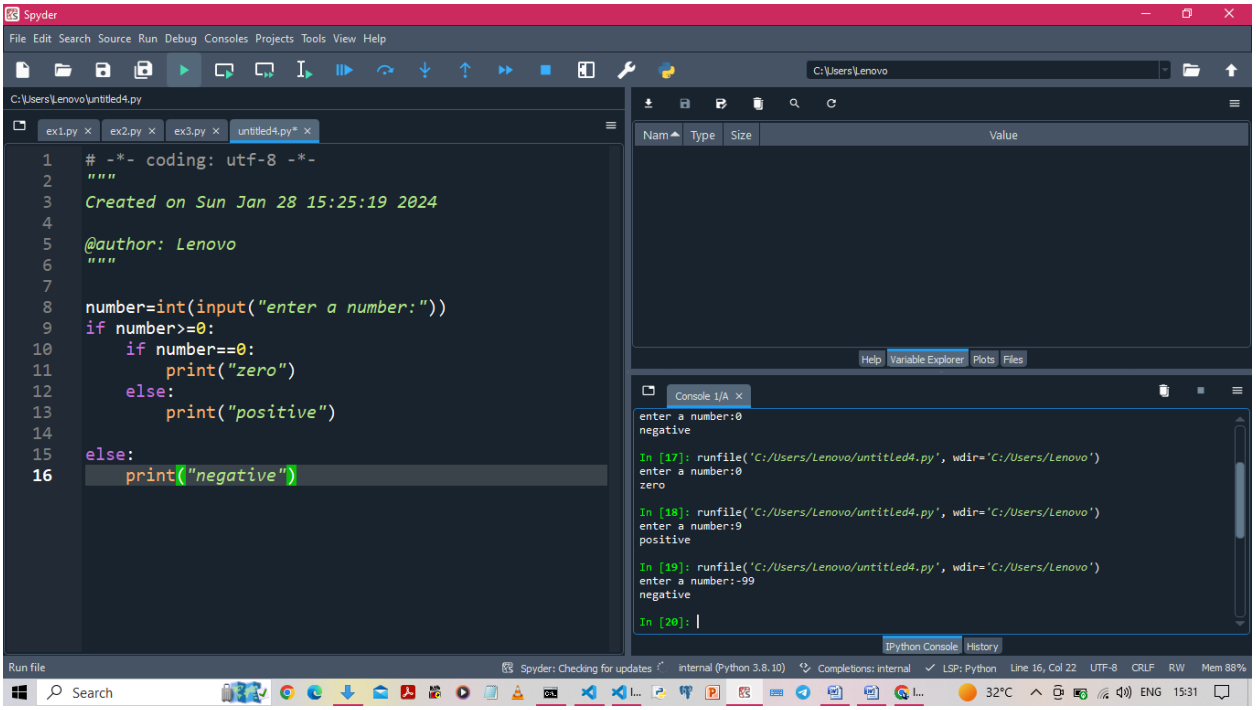
if…else..statement :



if…elif…else…statement :



Nested if statement :

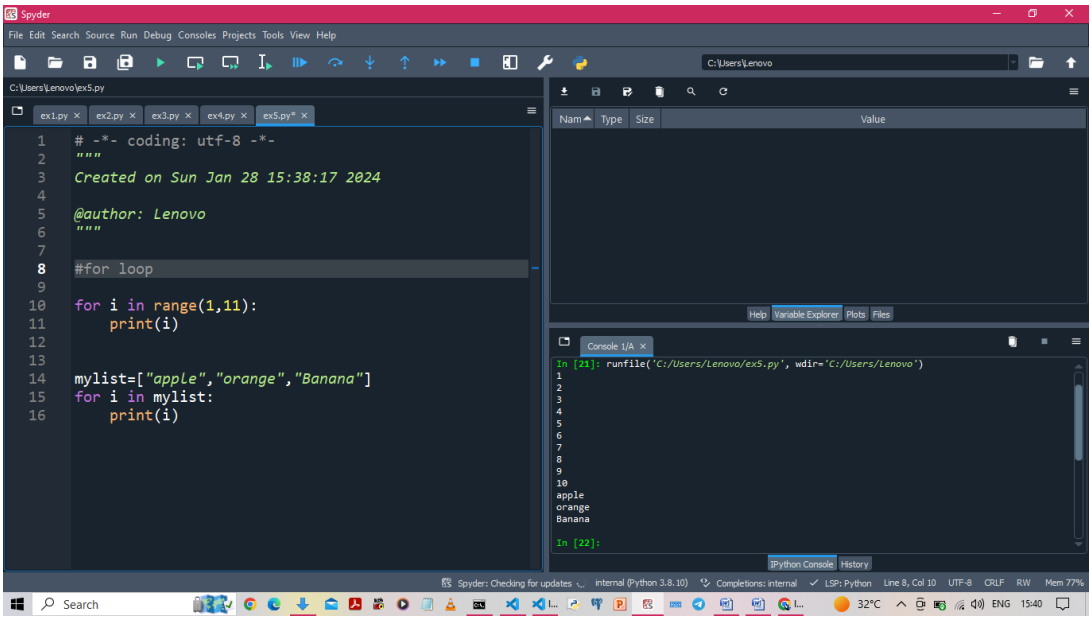


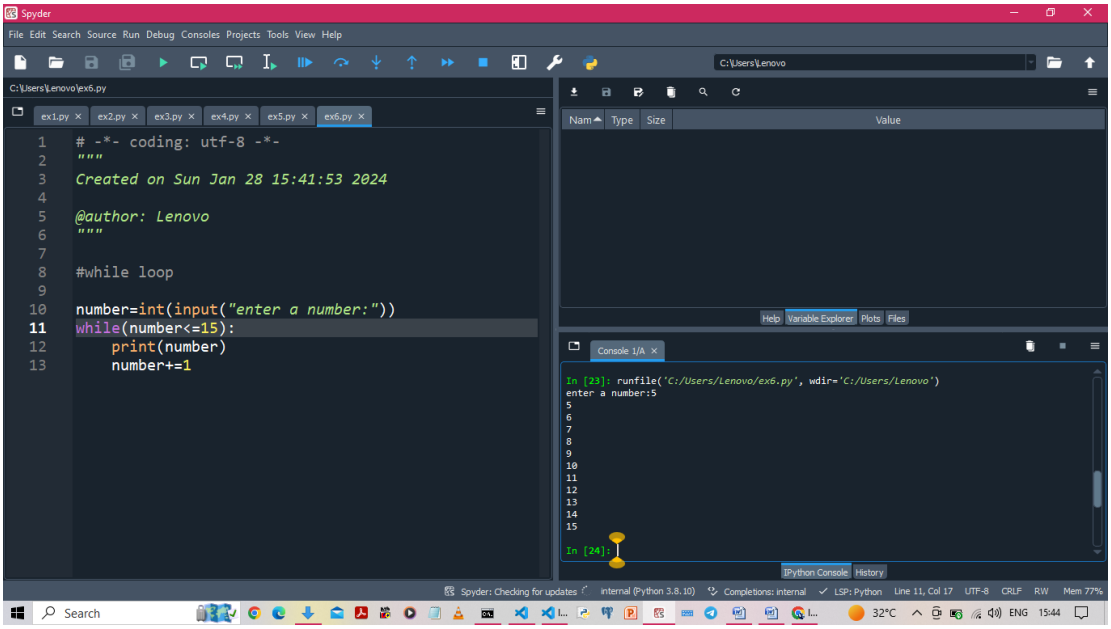
Looping Statements:

- **for loop**
- **while loop**

for loop is designed to repeatedly execute a code block.

while loop is used to execute the statements repeatedly until a given condition is satisfied.





1.3 Week 3:

In this week, I learned python libraries such as numpy (numericalpython), pandas (panel data) and basics of python.

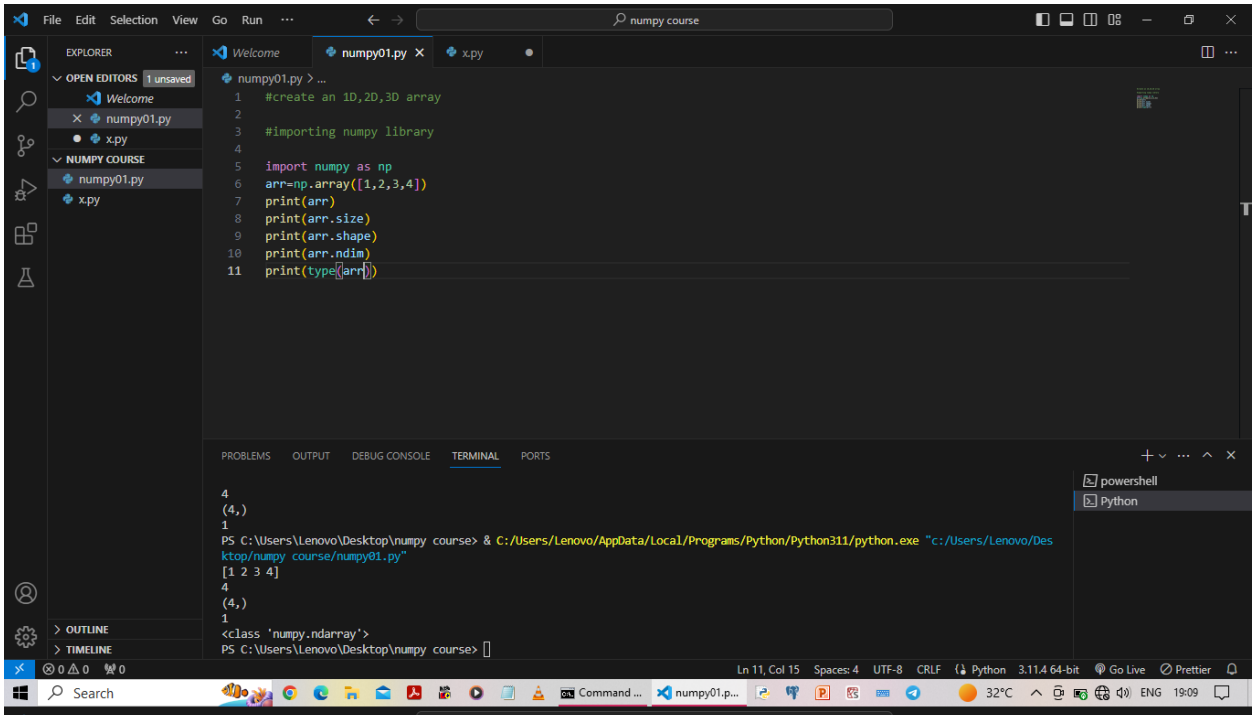
NUMPY:

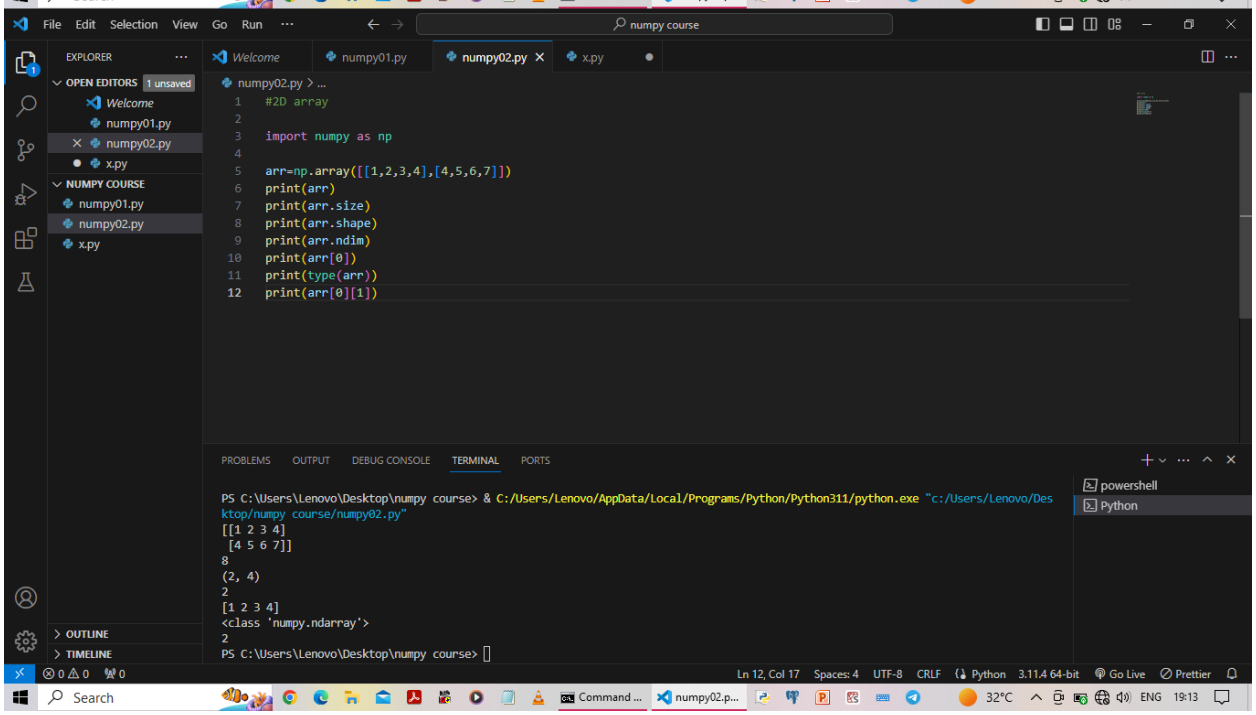
Numpy is a fundamental pacakage for numerical computations in python.

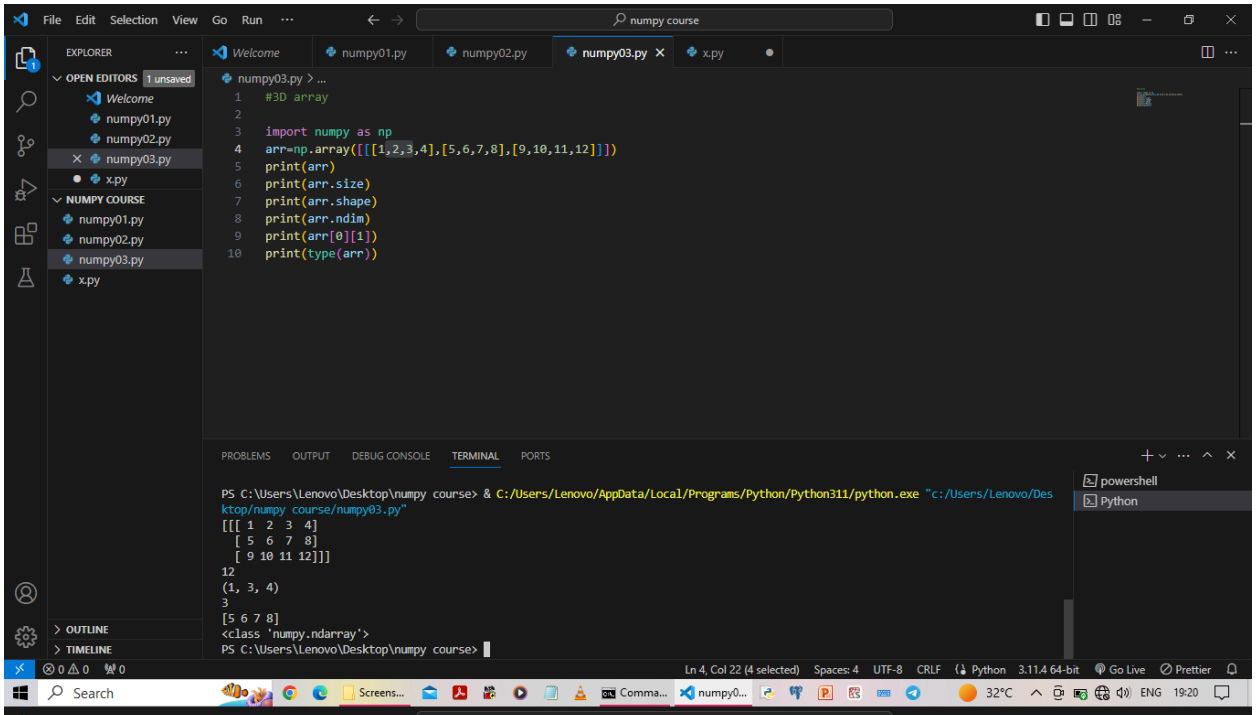
It supports N-dimensional array objects that can be used for processing multidimensional data.

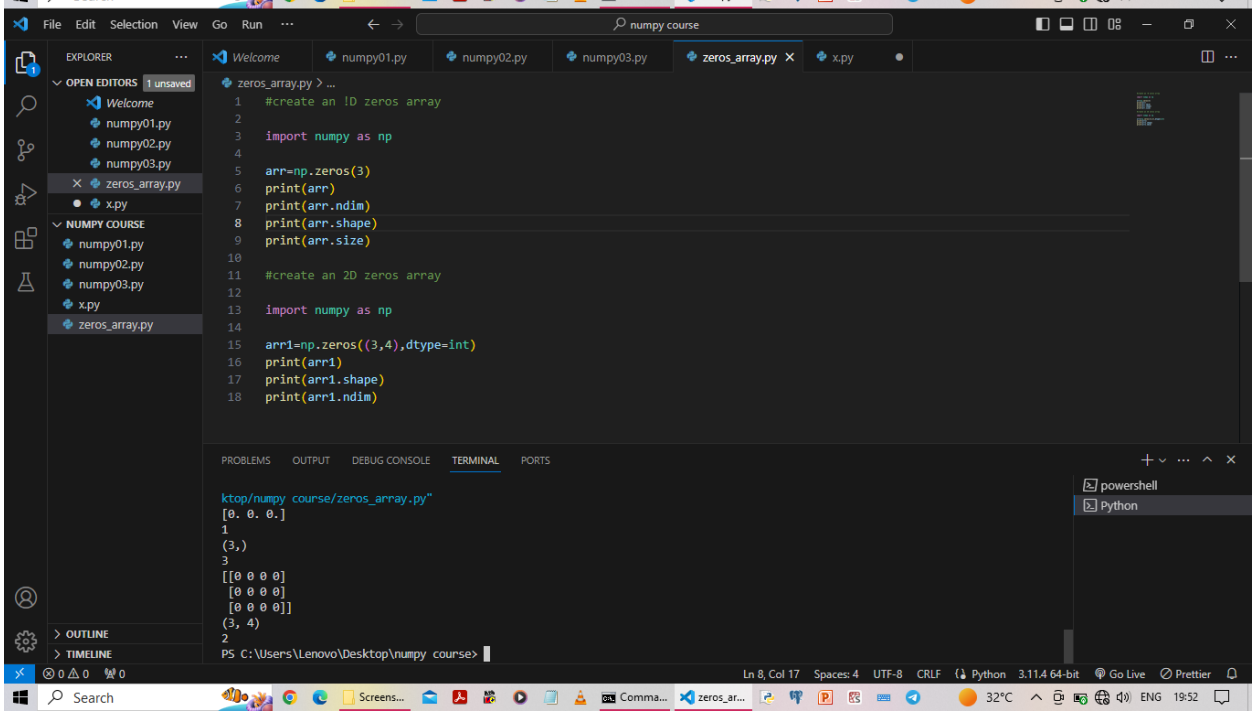
It supports different datatypes.

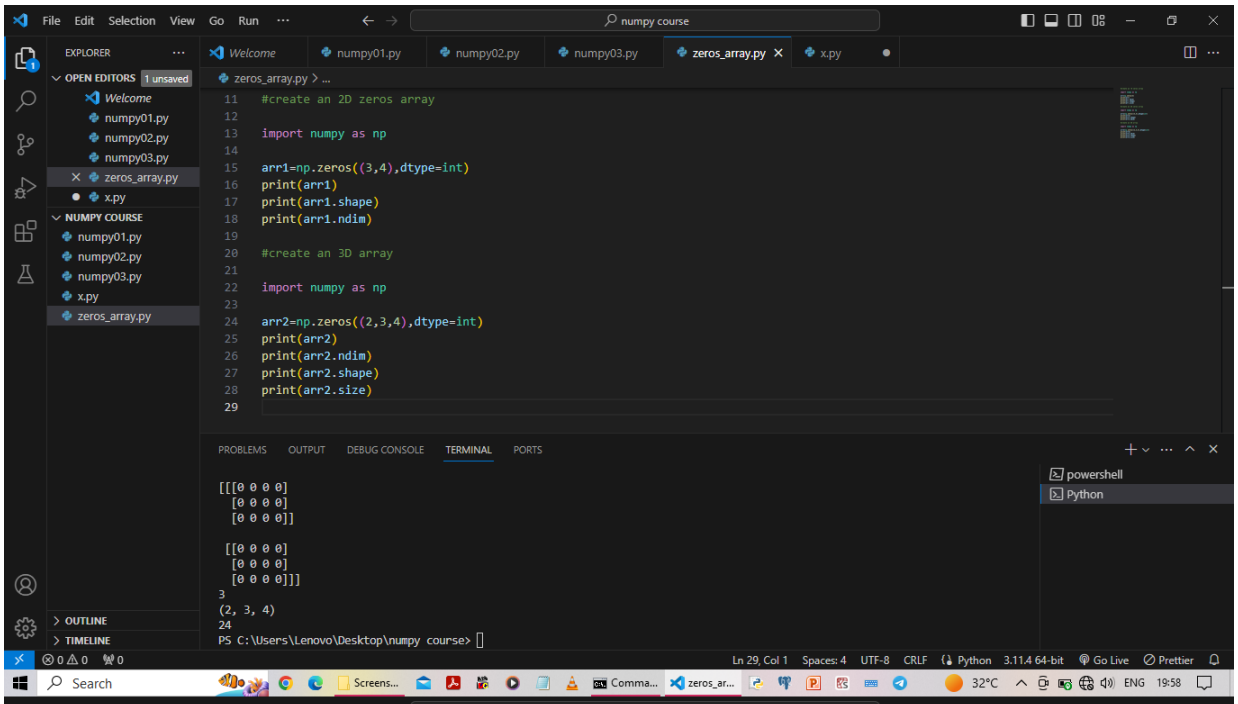
Examples:

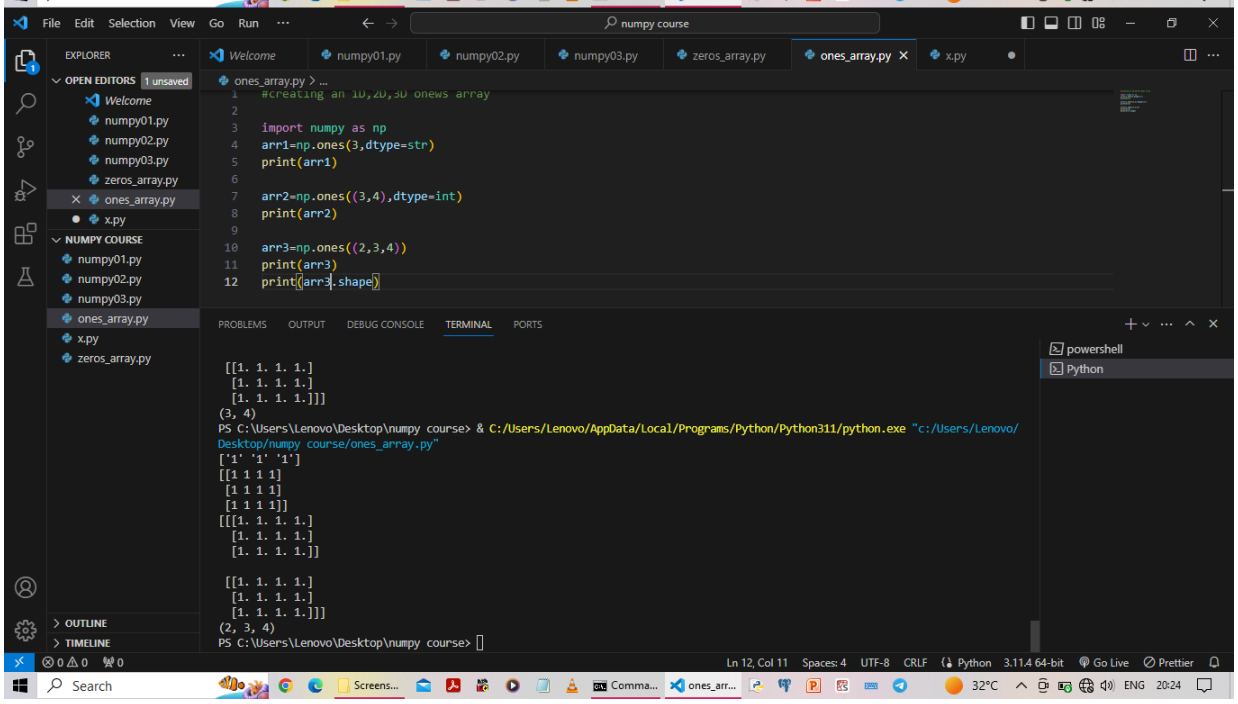

File Edit Selection View Go Run
numpy course
EXPLORER
OPEN EDITORS 1 unsaved
Welcome
numpy01.py
x.py
NUMPY COURSE
numpy01.py
x.py
Welcome
numpy01.py
x.py
numpy01.py > ...
#create an 1D,2D,3D array
#importing numpy library
import numpy as np
arr=np.array([1,2,3,4])
print(arr)
print(arr.size)
print(arr.shape)
print(arr.ndim)
print(type(arr))
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
4
(4,)
1
PS C:\Users\Lenovo\Desktop\numpy course> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/Lenovo/Desktop/numpy course/numpy01.py"
[1 2 3 4]
4
(4,)
1
<class 'numpy.ndarray'>
PS C:\Users\Lenovo\Desktop\numpy course>
powershell
Python
OUTLINE
TIMELINE
Ln 11, Col 15 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier
Search
32°C
ENG 19:09

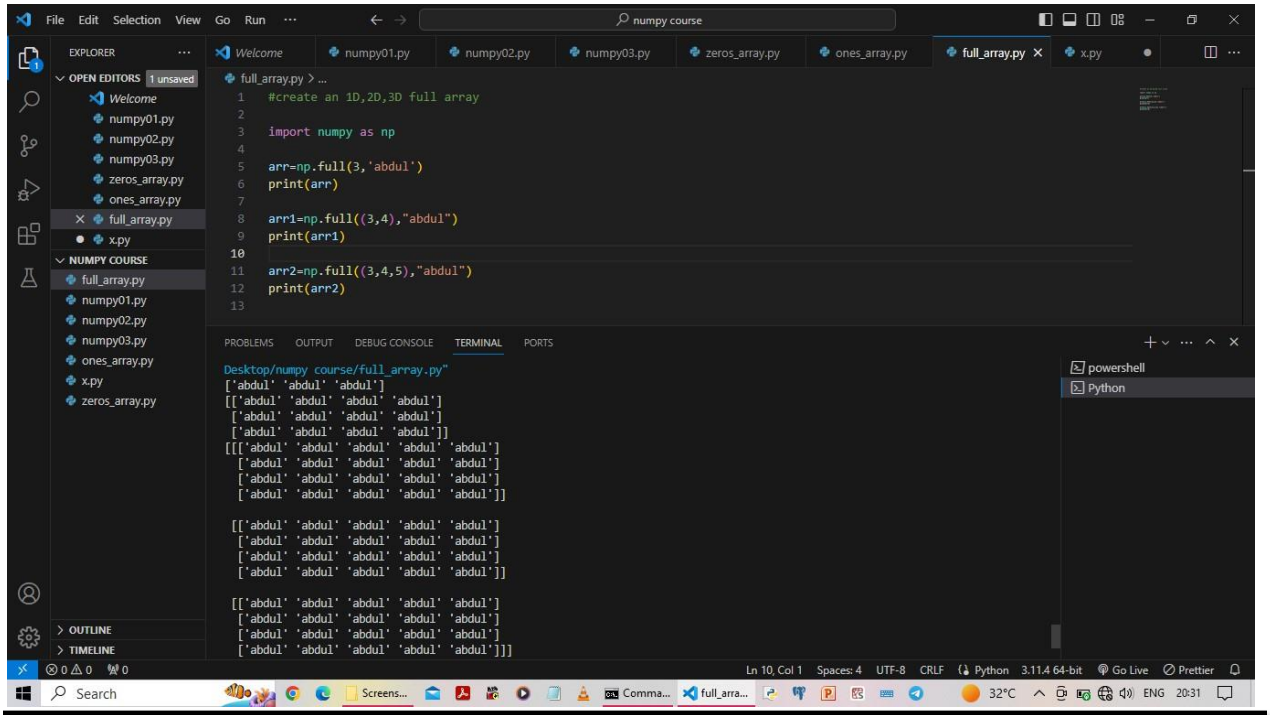

File Edit Selection View Go Run
numpy course
EXPLORER
OPEN EDITORS 1 unsaved
Welcome
numpy01.py
numpy02.py
x.py
NUMPY COURSE
numpy01.py
numpy02.py
x.py
Welcome
numpy01.py
numpy02.py
x.py
numpy02.py > ...
#2D array
import numpy as np
arr=np.array([[1,2,3,4],[4,5,6,7]])
print(arr)
print(arr.size)
print(arr.shape)
print(arr.ndim)
print(arr[0])
print(type(arr))
print(arr[0][1])
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Lenovo\Desktop\numpy course> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/Lenovo/Desktop/numpy course/numpy02.py"
[[1 2 3 4]
[4 5 6 7]]
8
(2, 4)
2
[1 2 3 4]
<class 'numpy.ndarray'>
2
PS C:\Users\Lenovo\Desktop\numpy course>
powershell
Python
OUTLINE
TIMELINE
Ln 12, Col 17 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier
Search
32°C
ENG 19:13

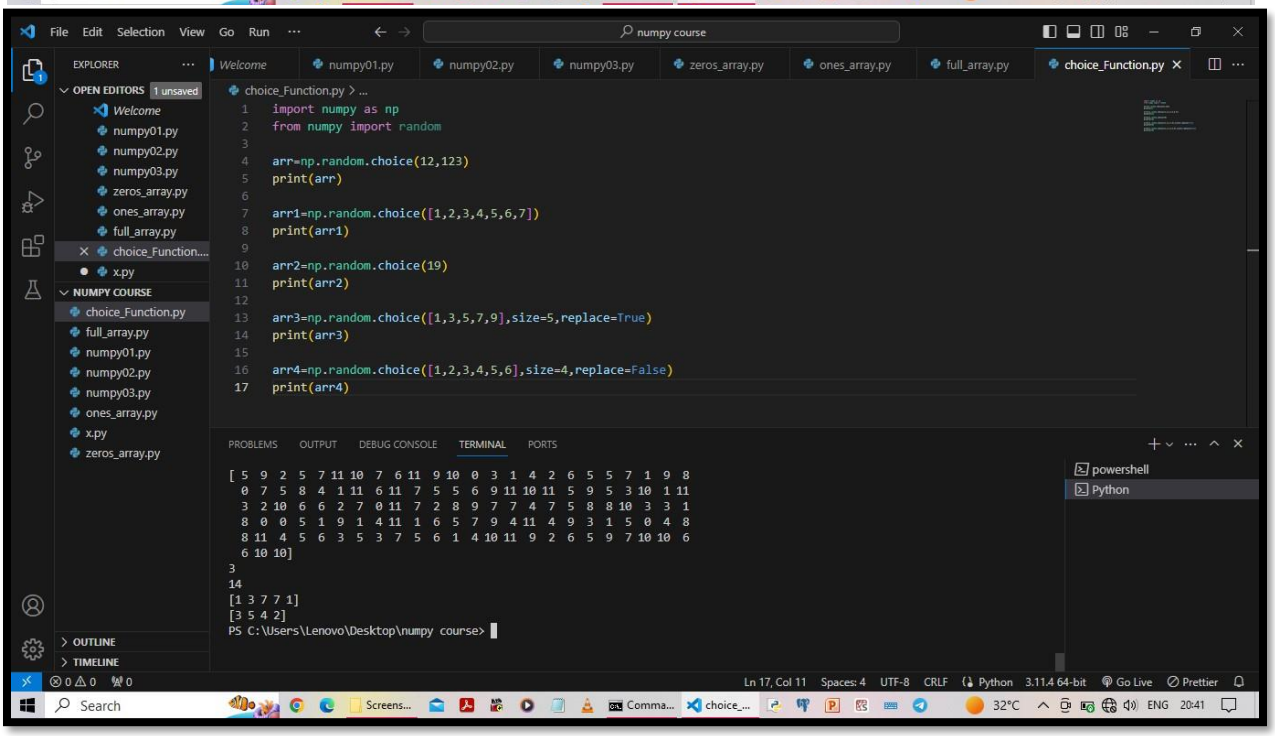

File Edit Selection View Go Run
numpy course
EXPLORER
OPEN EDITORS 1 unsaved
Welcome
numpy01.py
numpy02.py
numpy03.py
x.py
NUMPY COURSE
numpy01.py
numpy02.py
numpy03.py
x.py
OUTLINE
TIMELINE
numpy03.py > ...
1 #3D array
2
3 import numpy as np
4 arr=np.array([[[1,2,3,4],[5,6,7,8],[9,10,11,12]]])
5 print(arr)
6 print(arr.size)
7 print(arr.shape)
8 print(arr.ndim)
9 print(arr[0][1])
10 print(type(arr))
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Lenovo\Desktop\numpy course> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/Lenovo/Des
ktop/numpy course/numpy03.py"
[[[1 2 3 4]
[5 6 7 8]
[9 10 11 12]]]
12
(1, 3, 4)
3
[5 6 7 8]
<class 'numpy.ndarray'>
PS C:\Users\Lenovo\Desktop\numpy course>
powershell
Python
Ln 4, Col 22 (4 selected) Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier
Search
32°C
ENG 19:20

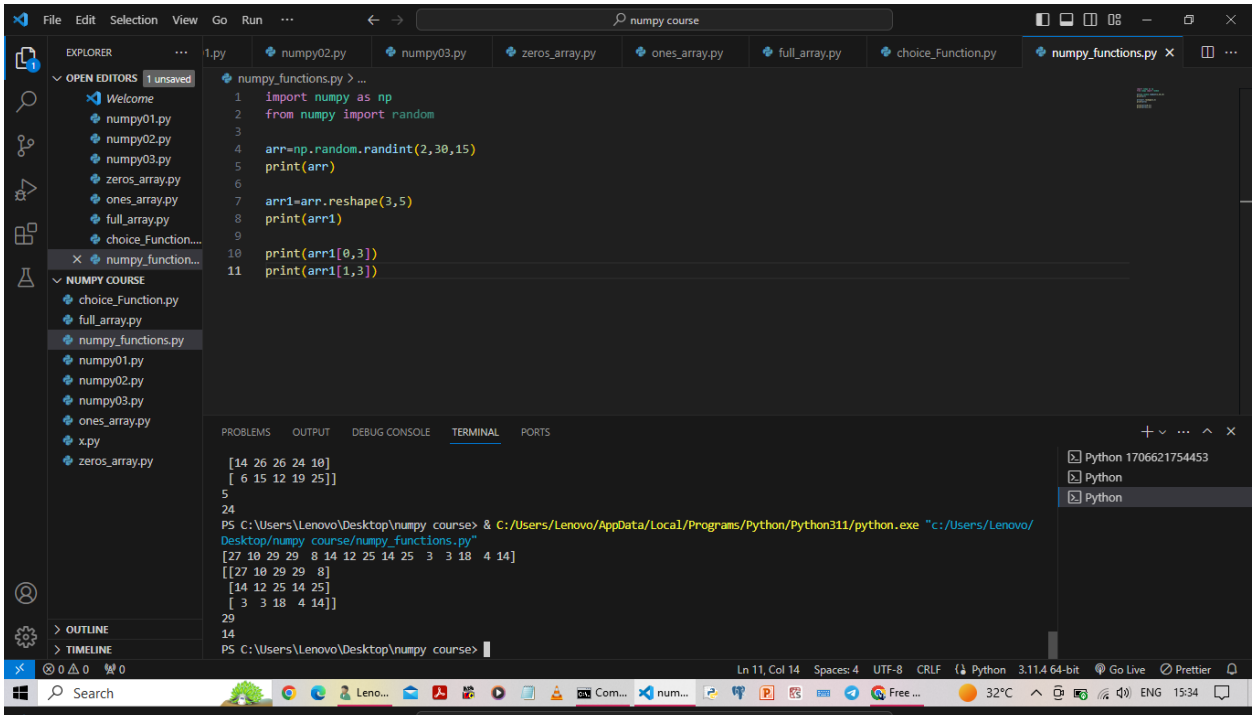

File Edit Selection View Go Run
numpy course
EXPLORER
OPEN EDITORS 1 unsaved
Welcome
numpy01.py
numpy02.py
numpy03.py
zeros_array.py
x.py
NUMPY COURSE
numpy01.py
numpy02.py
numpy03.py
x.py
zeros_array.py
OUTLINE
TIMELINE
zeros_array.py > ...
1 #create an !D zeros array
2
3 import numpy as np
4
5 arr=np.zeros(3)
6 print(arr)
7 print(arr.ndim)
8 print(arr.shape)
9 print(arr.size)
10
11 #create an 2D zeros array
12
13 import numpy as np
14
15 arr1=np.zeros((3,4),dtype=int)
16 print(arr1)
17 print(arr1.shape)
18 print(arr1.ndim)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
ktop/numpy course/zeros_array.py"
[0. 0. 0.]
1
(3,)
3
[[0 0 0 0]
[0 0 0 0]
[0 0 0 0]]
(3, 4)
2
PS C:\Users\Lenovo\Desktop\numpy course>
powershell
Python
Ln 8, Col 17 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier
Search
32°C
ENG 19:52

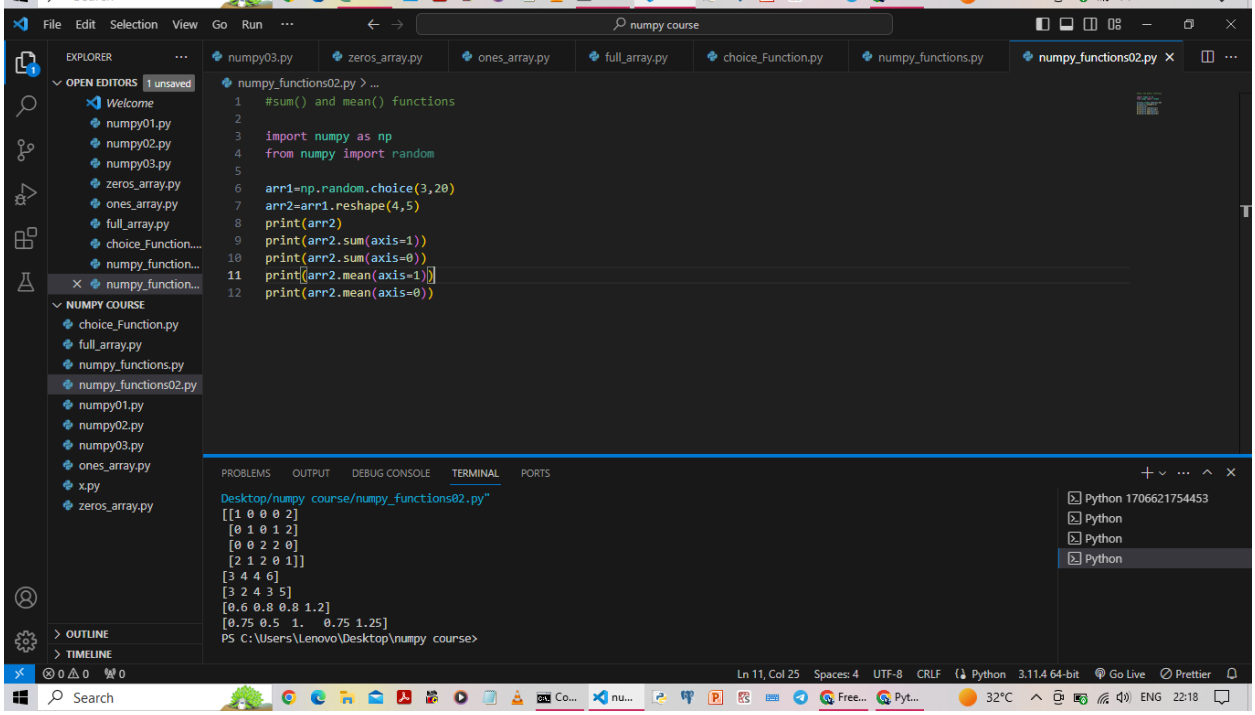

File Edit Selection View Go Run
numpy course
EXPLORER
OPEN EDITORS 1 unsaved
Welcome
numpy01.py
numpy02.py
numpy03.py
zeros_array.py
x.py
NUMPY COURSE
numpy01.py
numpy02.py
numpy03.py
x.py
zeros_array.py
OUTLINE
TIMELINE
zeros_array.py > ...
#create an 2D zeros array
import numpy as np
arr1=np.zeros((3,4),dtype=int)
print(arr1)
print(arr1.shape)
print(arr1.ndim)
#create an 3D array
import numpy as np
arr2=np.zeros((2,3,4),dtype=int)
print(arr2)
print(arr2.ndim)
print(arr2.shape)
print(arr2.size)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[[[0 0 0 0]
[0 0 0 0]
[0 0 0 0]]
[[0 0 0 0]
[0 0 0 0]
[0 0 0 0]]]
3
(2, 3, 4)
24
PS C:\Users\Lenovo\Desktop\numpy course>
powershell
Python
Ln 29, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier
Search
32°C ENG 19:58

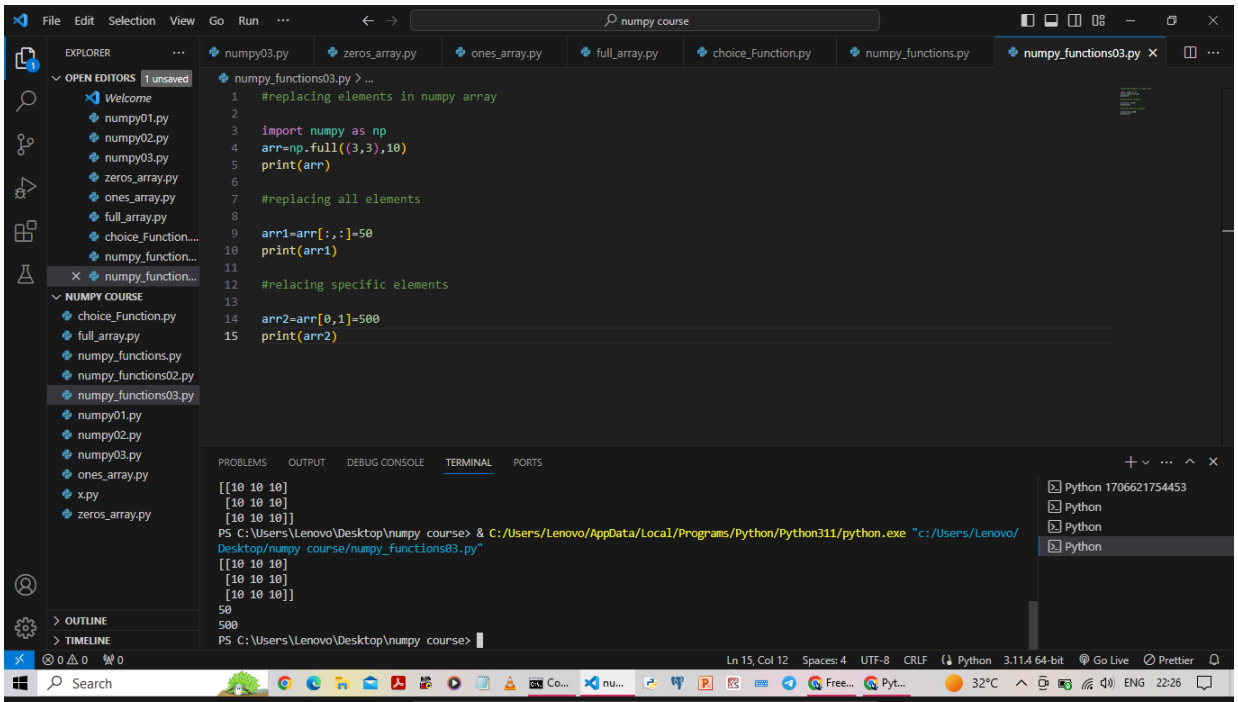

File Edit Selection View Go Run
numpy course
EXPLORER
OPEN EDITORS 1 unsaved
Welcome
numpy01.py
numpy02.py
numpy03.py
zeros_array.py
ones_array.py
x.py
NUMPY COURSE
numpy01.py
numpy02.py
numpy03.py
ones_array.py
x.py
zeros_array.py
OUTLINE
TIMELINE
ones_array.py > ...
#creating an 1D,2D,3D onews array
import numpy as np
arr1=np.ones(3,dtype=str)
print(arr1)
arr2=np.ones((3,4),dtype=int)
print(arr2)
arr3=np.ones((2,3,4))
print(arr3)
print(arr3.shape)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[[1. 1. 1. 1.]
[1. 1. 1. 1.]
[1. 1. 1. 1.]]]
(3, 4)
PS C:\Users\Lenovo\Desktop\numpy course> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/Lenovo/Desktop/numpy course/ones_array.py"
['1' '1' '1']
[[1 1 1 1]
[1 1 1 1]
[1 1 1 1]]
[[[1. 1. 1. 1.]
[1. 1. 1. 1.]
[1. 1. 1. 1.]]
[[1. 1. 1. 1.]
[1. 1. 1. 1.]
[1. 1. 1. 1.]]]
(2, 3, 4)
PS C:\Users\Lenovo\Desktop\numpy course>
powershell
Python
Ln 12, Col 11 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier
Search
32°C ENG 20:24

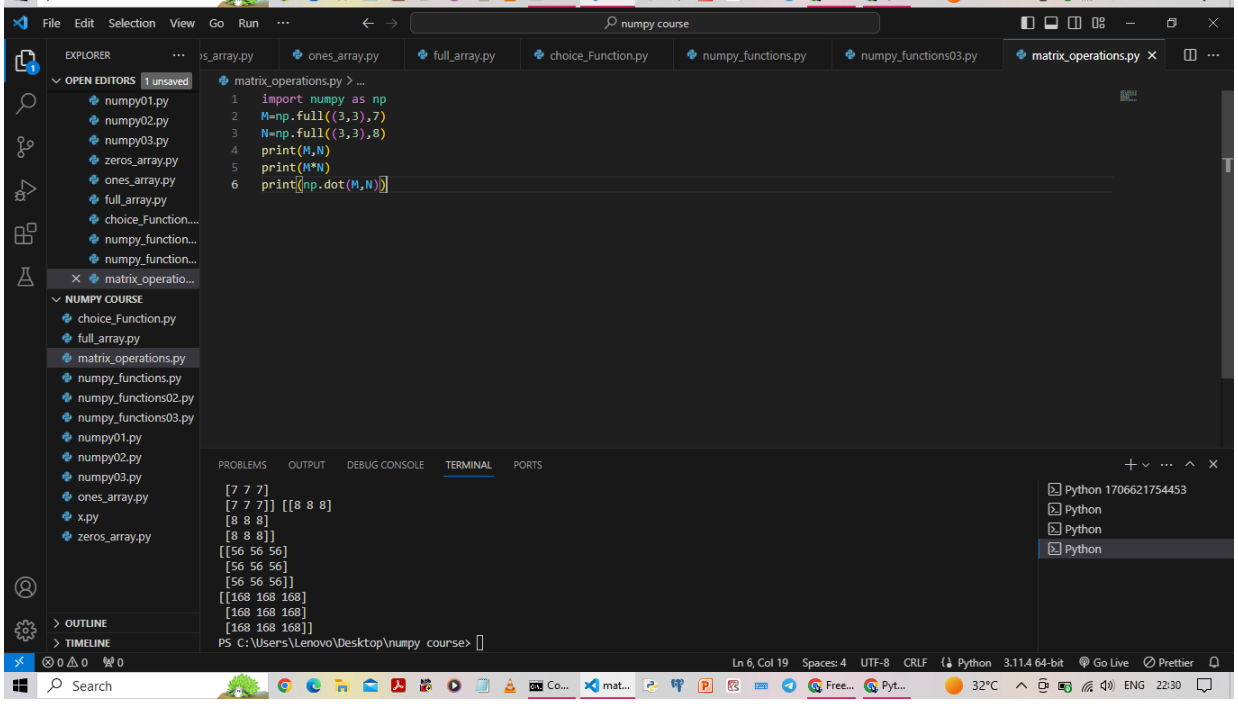

File Edit Selection View Go Run
numpy course
EXPLORER
OPEN EDITORS 1 unsaved
Welcome
numpy01.py
numpy02.py
numpy03.py
zeros_array.py
ones_array.py
full_array.py
x.py
NUMPY COURSE
full_array.py
numpy01.py
numpy02.py
numpy03.py
ones_array.py
x.py
zeros_array.py
OUTLINE
TIMELINE
full_array.py > ...
#create an 1D,2D,3D full array
import numpy as np
arr=np.full(3,'abdul')
print(arr)
arr1=np.full((3,4),"abdul")
print(arr1)
arr2=np.full((3,4,5),"abdul")
print(arr2)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Desktop/numpy course/full_array.py"
['abdul' 'abdul' 'abdul']
[['abdul' 'abdul' 'abdul' 'abdul']
['abdul' 'abdul' 'abdul' 'abdul']
['abdul' 'abdul' 'abdul' 'abdul']]
[[['abdul' 'abdul' 'abdul' 'abdul' 'abdul']
['abdul' 'abdul' 'abdul' 'abdul' 'abdul']
['abdul' 'abdul' 'abdul' 'abdul' 'abdul']
['abdul' 'abdul' 'abdul' 'abdul' 'abdul']]
[['abdul' 'abdul' 'abdul' 'abdul' 'abdul']
['abdul' 'abdul' 'abdul' 'abdul' 'abdul']
['abdul' 'abdul' 'abdul' 'abdul' 'abdul']
['abdul' 'abdul' 'abdul' 'abdul' 'abdul']]
[['abdul' 'abdul' 'abdul' 'abdul' 'abdul']
['abdul' 'abdul' 'abdul' 'abdul' 'abdul']
['abdul' 'abdul' 'abdul' 'abdul' 'abdul']
['abdul' 'abdul' 'abdul' 'abdul' 'abdul']]]
powershell
Python
Ln 10, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier
Search
32°C ENG 20:31

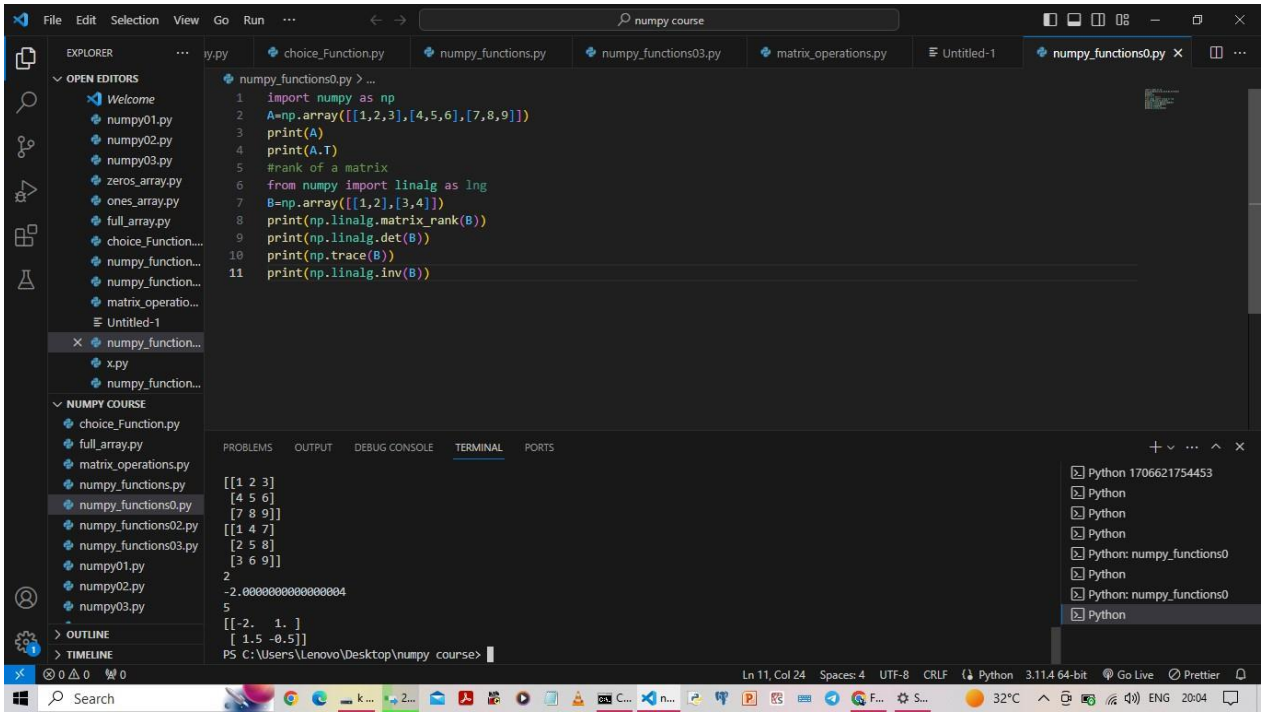

File Edit Selection View Go Run
numpy course
EXPLORER
OPEN EDITORS 1 unsaved
Welcome
numpy01.py
numpy02.py
numpy03.py
zeros_array.py
ones_array.py
full_array.py
choice_Function....
x.py
NUMPY COURSE
choice_Function.py
full_array.py
numpy01.py
numpy02.py
numpy03.py
ones_array.py
x.py
zeros_array.py
OUTLINE
TIMELINE
choice_Function.py > ...
import numpy as np
from numpy import random
arr=np.random.choice(12,123)
print(arr)
arr1=np.random.choice([1,2,3,4,5,6,7])
print(arr1)
arr2=np.random.choice(19)
print(arr2)
arr3=np.random.choice([1,3,5,7,9],size=5,replace=True)
print(arr3)
arr4=np.random.choice([1,2,3,4,5,6],size=4,replace=False)
print(arr4)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[5 9 2 5 7 11 10 7 6 11 9 10 0 3 1 4 2 6 5 5 7 1 9 8
0 7 5 8 4 1 11 6 11 7 5 5 6 9 11 10 11 5 9 5 3 10 1 11
3 2 10 6 6 2 7 0 11 7 2 8 9 7 7 4 7 5 8 8 10 3 3 1
8 0 0 5 1 9 1 4 11 1 6 5 7 9 4 11 4 9 3 1 5 0 4 8
8 11 4 5 6 3 5 3 7 5 6 1 4 10 11 9 2 6 5 9 7 10 10 6
6 10 10]
3
14
[1 3 7 7 1]
[3 5 4 2]
PS C:\Users\Lenovo\Desktop\numpy course>
powershell
Python
Ln 17, Col 11 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier
Search
32°C ENG 20:41


File Edit Selection View Go Run
numpy course
EXPLORER
numpy02.py
numpy03.py
zeros_array.py
ones_array.py
full_array.py
choice_Function.py
numpy_functions.py
OPEN EDITORS 1 unsaved
Welcome
numpy01.py
numpy02.py
numpy03.py
zeros_array.py
ones_array.py
full_array.py
choice_Function....
numpy_function...
NUMPY COURSE
choice_Function.py
full_array.py
numpy_functions.py
numpy01.py
numpy02.py
numpy03.py
ones_array.py
x.py
zeros_array.py
OUTLINE
TIMELINE
numpy_functions.py > ...
1 import numpy as np
2 from numpy import random
3
4 arr=np.random.randint(2,30,15)
5 print(arr)
6
7 arr1=arr.reshape(3,5)
8 print(arr1)
9
10 print(arr1[0,3])
11 print(arr1[1,3])
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[14 26 26 24 10]
[6 15 12 19 25]]
5
24
PS C:\Users\Lenovo\Desktop\numpy course> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/Lenovo/Desktop/numpy course/numpy_functions.py"
[27 10 29 29 8 14 12 25 14 25 3 3 18 4 14]
[[27 10 29 29 8]
[14 12 25 14 25]
[3 3 18 4 14]]
29
14
PS C:\Users\Lenovo\Desktop\numpy course>
Python 1706621754453
Python
Python
Ln 11, Col 14 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier
Search
32°C
ENG 15:34


File Edit Selection View Go Run
numpy course
EXPLORER
numpy03.py
zeros_array.py
ones_array.py
full_array.py
choice_Function.py
numpy_functions.py
numpy_functions02.py
OPEN EDITORS 1 unsaved
Welcome
numpy01.py
numpy02.py
numpy03.py
zeros_array.py
ones_array.py
full_array.py
choice_Function....
numpy_function...
numpy_function...
NUMPY COURSE
choice_Function.py
full_array.py
numpy_functions.py
numpy_functions02.py
numpy01.py
numpy02.py
numpy03.py
ones_array.py
x.py
zeros_array.py
OUTLINE
TIMELINE
numpy_functions02.py > ...
1 #sum() and mean() functions
2
3 import numpy as np
4 from numpy import random
5
6 arr1=np.random.choice(3,20)
7 arr2=arr1.reshape(4,5)
8 print(arr2)
9 print(arr2.sum(axis=1))
10 print(arr2.sum(axis=0))
11 print(arr2.mean(axis=1))
12 print(arr2.mean(axis=0))
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Desktop/numpy course/numpy_functions02.py"
[[1 0 0 0 2]
[0 1 0 1 2]
[0 0 2 2 0]
[2 1 2 0 1]]
[3 4 4 6]
[3 2 4 3 5]
[0.6 0.8 0.8 1.2]
[0.75 0.5 1. 0.75 1.25]
PS C:\Users\Lenovo\Desktop\numpy course>
Python 1706621754453
Python
Python
Python
Ln 11, Col 25 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier
Search
32°C
ENG 22:18

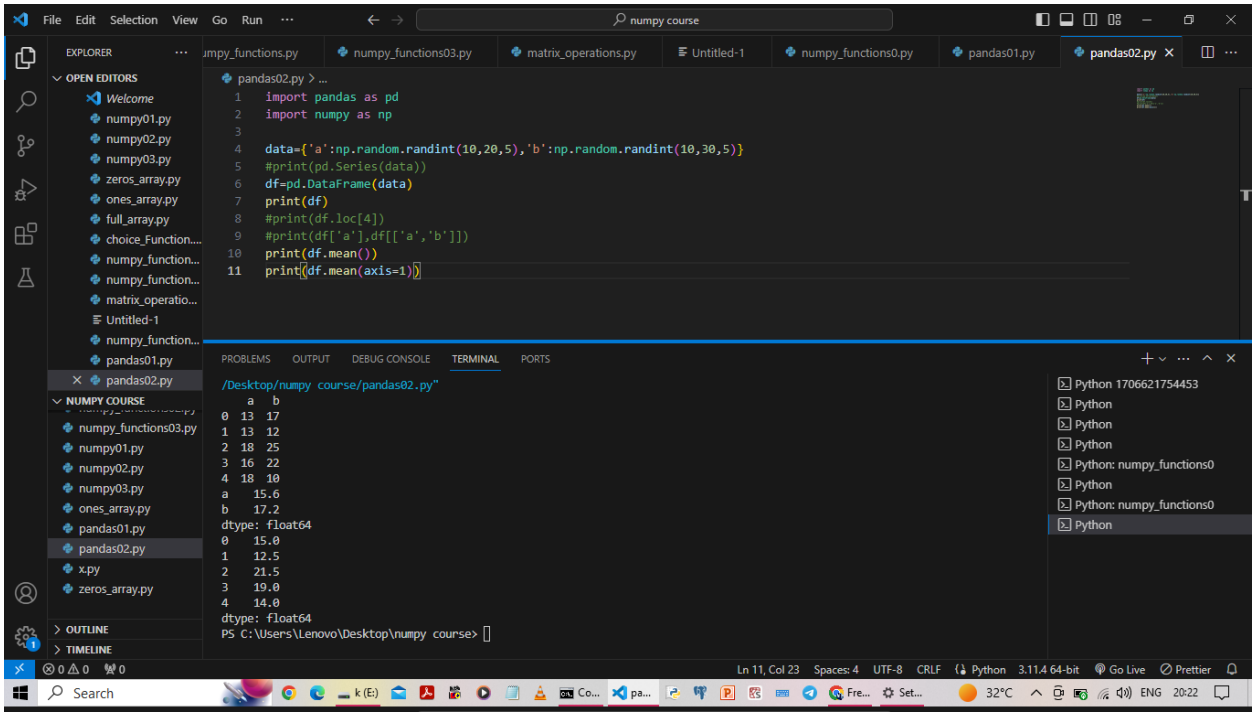

File Edit Selection View Go Run
numpy course
EXPLORER
OPEN EDITORS 1 unsaved
Welcome
numpy01.py
numpy02.py
numpy03.py
zeros_array.py
ones_array.py
full_array.py
choice_Function....
numpy_function...
numpy_function...
NUMPY COURSE
choice_Function.py
full_array.py
numpy_functions.py
numpy_functions02.py
numpy_functions03.py
numpy01.py
numpy02.py
numpy03.py
ones_array.py
x.py
zeros_array.py
OUTLINE
TIMELINE
numpy03.py
zeros_array.py
ones_array.py
full_array.py
choice_Function.py
numpy_functions.py
numpy_functions03.py
numpy_functions03.py
numpy_functions03.py > ...
1 #replacing elements in numpy array
2
3 import numpy as np
4 arr=np.full((3,3),10)
5 print(arr)
6
7 #replacing all elements
8
9 arr1=arr[:,:]=50
10 print(arr1)
11
12 #relacing specific elements
13
14 arr2=arr[0,1]=500
15 print(arr2)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[[10 10 10]
[10 10 10]
[10 10 10]]
PS C:\Users\Lenovo\Desktop\numpy course> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/Lenovo/Desktop/numpy course/numpy_functions03.py"
[[10 10 10]
[10 10 10]
[10 10 10]]
50
500
PS C:\Users\Lenovo\Desktop\numpy course>
Python 1706621754453
Python
Python
Python
Ln 15, Col 12 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier
Search
32°C
ENG
22:26

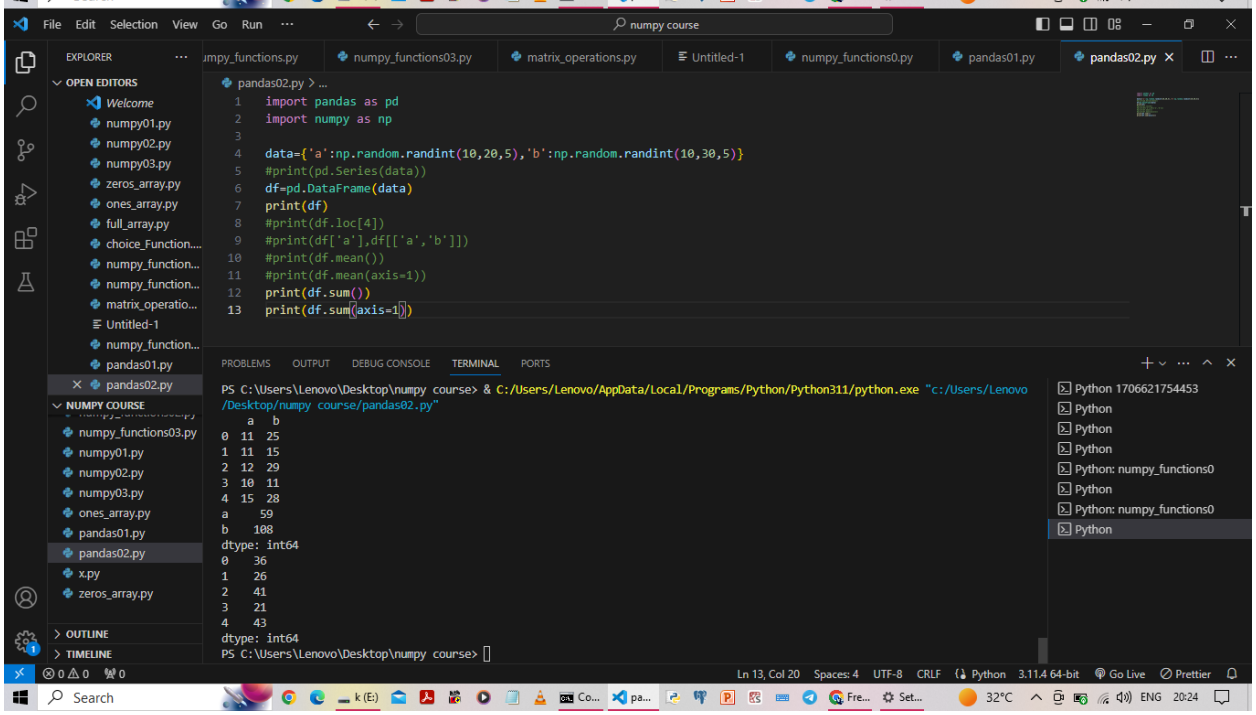

File Edit Selection View Go Run
numpy course
EXPLORER
OPEN EDITORS 1 unsaved
numpy01.py
numpy02.py
numpy03.py
zeros_array.py
ones_array.py
full_array.py
choice_Function....
numpy_function...
numpy_function...
matrix_operatio...
NUMPY COURSE
choice_Function.py
full_array.py
matrix_operations.py
numpy_functions.py
numpy_functions02.py
numpy_functions03.py
numpy01.py
numpy02.py
numpy03.py
ones_array.py
x.py
zeros_array.py
OUTLINE
TIMELINE
ones_array.py
full_array.py
choice_Function.py
numpy_functions.py
numpy_functions03.py
matrix_operations.py
matrix_operations.py > ...
1 import numpy as np
2 M=np.full((3,3),7)
3 N=np.full((3,3),8)
4 print(M,N)
5 print(M*N)
6 print(np.dot(M,N))
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[7 7 7]
[7 7 7]] [[8 8 8]
[8 8 8]
[8 8 8]]
[[56 56 56]
[56 56 56]
[56 56 56]]
[[168 168 168]
[168 168 168]
[168 168 168]]
PS C:\Users\Lenovo\Desktop\numpy course>
Python 1706621754453
Python
Python
Python
Ln 6, Col 19 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier
Search
32°C
ENG
22:30

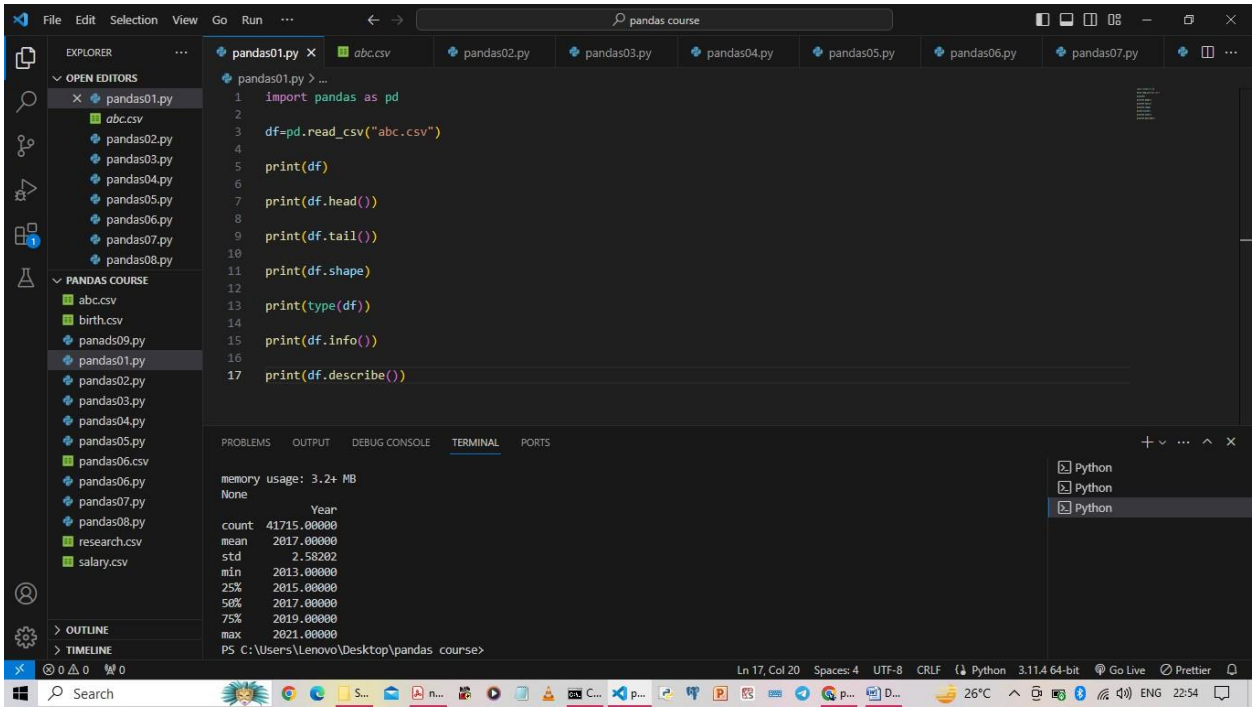


PANDAS :

Pandas is a python library used for working with Datasets. It has functions for analyzing, cleaning, exploring, and manipulating data.


File Edit Selection View Go Run
numpy course
EXPLORER
OPEN EDITORS
NUMPY COURSE
pandas02.py
import pandas as pd
import numpy as np
data={'a':np.random.randint(10,20,5),'b':np.random.randint(10,30,5)}
#print(pd.Series(data))
df=pd.DataFrame(data)
print(df)
#print(df.loc[4])
#print(df['a'],df[['a','b']])
print(df.mean())
print(df.mean(axis=1))
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
/Desktop/numpy course/pandas02.py"
a b
0 13 17
1 13 12
2 18 25
3 16 22
4 18 10
a 15.6
b 17.2
dtype: float64
0 15.0
1 12.5
2 21.5
3 19.0
4 14.0
dtype: float64
PS C:\Users\Lenovo\Desktop\numpy course>
Ln 11, Col 23 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier
Search
32°C ENG 20:22


File Edit Selection View Go Run
numpy course
EXPLORER
OPEN EDITORS
NUMPY COURSE
pandas02.py
import pandas as pd
import numpy as np
data={'a':np.random.randint(10,20,5),'b':np.random.randint(10,30,5)}
#print(pd.Series(data))
df=pd.DataFrame(data)
print(df)
#print(df.loc[4])
#print(df['a'],df[['a','b']])
#print(df.mean())
#print(df.mean(axis=1))
print(df.sum())
print(df.sum(axis=1))
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Lenovo\Desktop\numpy course> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/Lenovo
/Desktop/numpy course/pandas02.py"
a b
0 11 25
1 11 15
2 12 29
3 10 11
4 15 28
a 59
b 108
dtype: int64
0 36
1 26
2 41
3 21
4 43
dtype: int64
PS C:\Users\Lenovo\Desktop\numpy course>
Ln 13, Col 20 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier
Search
32°C ENG 20:24


File Edit Selection View Go Run
pandas course
EXPLORER
OPEN EDITORS
pandas01.py
abc.csv
pandas02.py
pandas03.py
pandas04.py
pandas05.py
pandas06.py
pandas07.py
pandas08.py
PANDAS COURSE
abc.csv
birth.csv
panads09.py
pandas01.py
pandas02.py
pandas03.py
pandas04.py
pandas05.py
pandas06.csv
pandas06.py
pandas07.py
pandas08.py
research.csv
salary.csv
OUTLINE
TIMELINE
pandas01.py
abc.csv
pandas02.py
pandas03.py
pandas04.py
pandas05.py
pandas06.py
pandas07.py
1 import pandas as pd
3 df=pd.read_csv("abc.csv")
5 print(df)
7 print(df.head())
9 print(df.tail())
11 print(df.shape)
13 print(type(df))
15 print(df.info())
17 print(df.describe())
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
memory usage: 3.2+ MB
None
Year
count 41715.00000
mean 2017.00000
std 2.58202
min 2013.00000
25% 2015.00000
50% 2017.00000
75% 2019.00000
max 2021.00000
PS C:\Users\Lenovo\Desktop\pandas course>
Python
Python
Python
Ln 17, Col 20 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier
Search
26°C
ENG
22:54

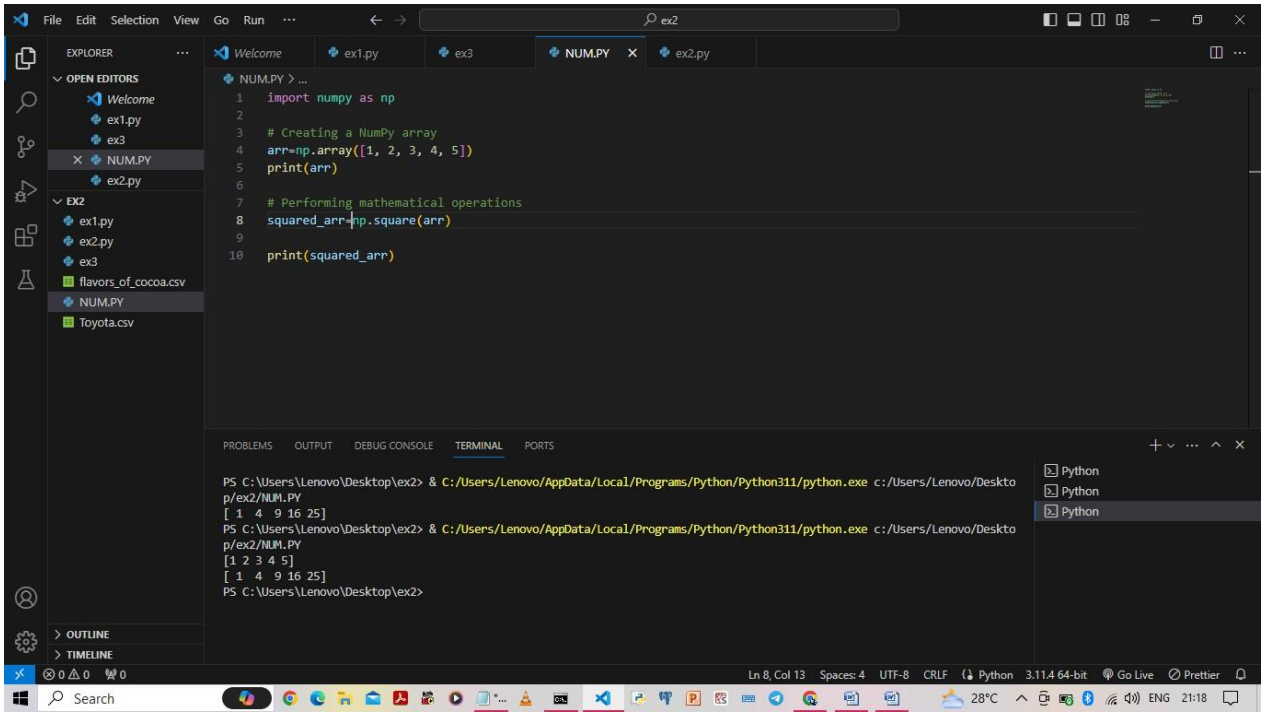
1.4 Week 4 :

-Relation between numpy and pandas:

NumPy and pandas are two popular libraries in Python that are commonly used for data manipulation, analysis, and numerical computations. While they serve different purposes ,they are often used together in data science and analysis workflows.

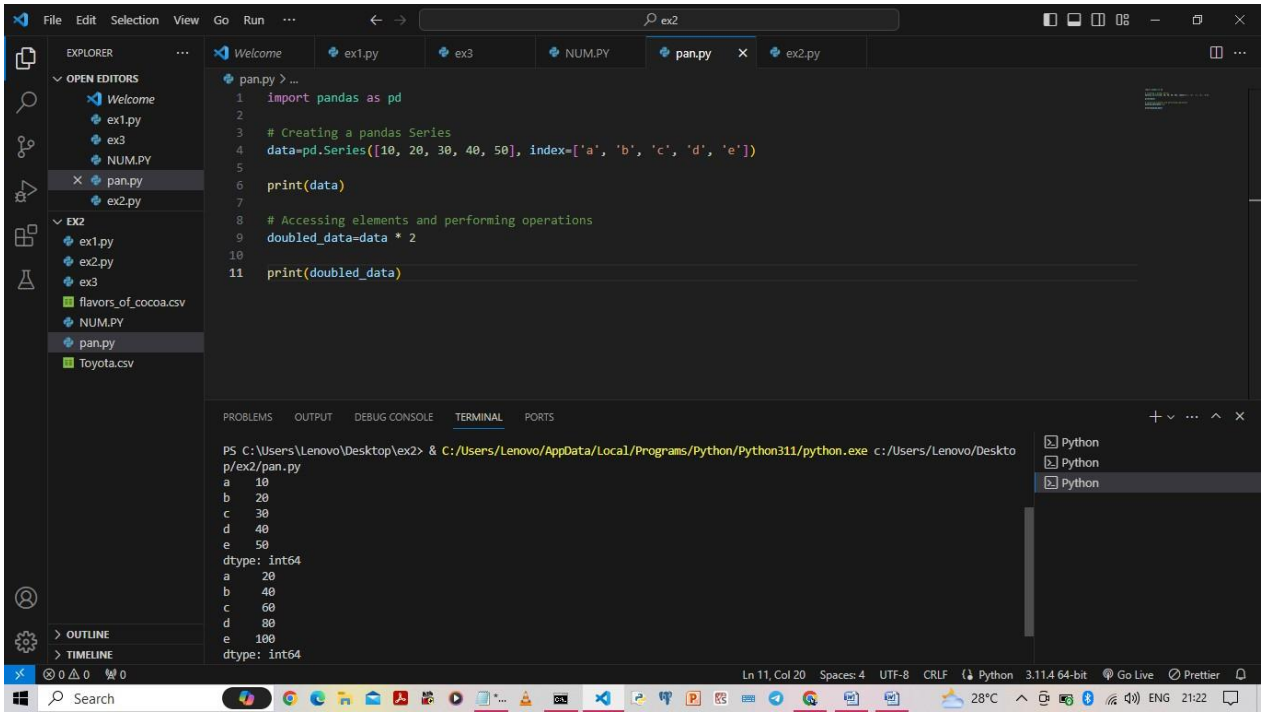
2 NUMPY:

NumPy (Numerical Python) provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these arrays. It forms the foundation for many other scientific computing libraries in Python.

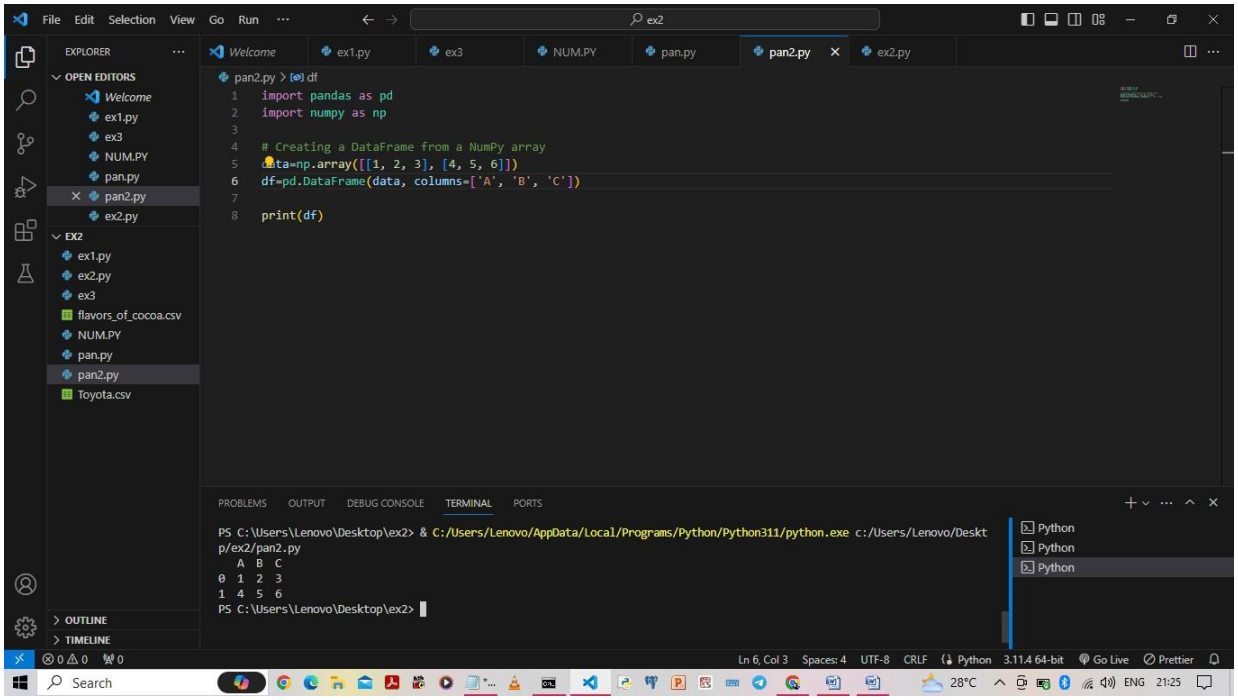

File Edit Selection View Go Run
ex2
EXPLORER
OPEN EDITORS
Welcome
ex1.py
ex3
NUM.PY
ex2.py
EX2
ex1.py
ex2.py
ex3
flavors_of_cocoa.csv
NUM.PY
Toyota.csv
OUTLINE
TIMELINE
Welcome
ex1.py
ex3
NUM.PY
ex2.py
NUM.PY > ...
1 import numpy as np
2
3 # Creating a NumPy array
4 arr=np.array([1, 2, 3, 4, 5])
5 print(arr)
6
7 # Performing mathematical operations
8 squared_arr=np.square(arr)
9
10 print(squared_arr)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Lenovo\Desktop\ex2> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python311/python.exe c:/Users/Lenovo/Desktop/ex2/NUM.PY
[1 4 9 16 25]
PS C:\Users\Lenovo\Desktop\ex2> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python311/python.exe c:/Users/Lenovo/Desktop/ex2/NUM.PY
[1 2 3 4 5]
[1 4 9 16 25]
PS C:\Users\Lenovo\Desktop\ex2>
Python
Python
Python
Ln 8, Col 13 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier
Search
28°C
ENG
21:18

3 PANDAS:

pandas is a powerful library for data manipulation and analysis. It introduces two key data structures: Series (1- dimensional labeled array) and DataFrame (2-dimensional table). Pandas is built on top of NumPy and often used for handling and analyzing structured data.



Pandas can also work seamlessly with NumPy arrays. Forexample, you can create a DataFrame using a NumPy array:



[

In the example above, pandas DataFrame df is created using a NumPy array, showcasing the interoperability between the two libraries. NumPy provides the underlying numerical operations, while pandas offers high-level data manipulation capabilities, making the powerful combination in data analysis workflows.

[

2 Introduction

2.1 About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.

i. UCT IoT Platform(UCT Insight)

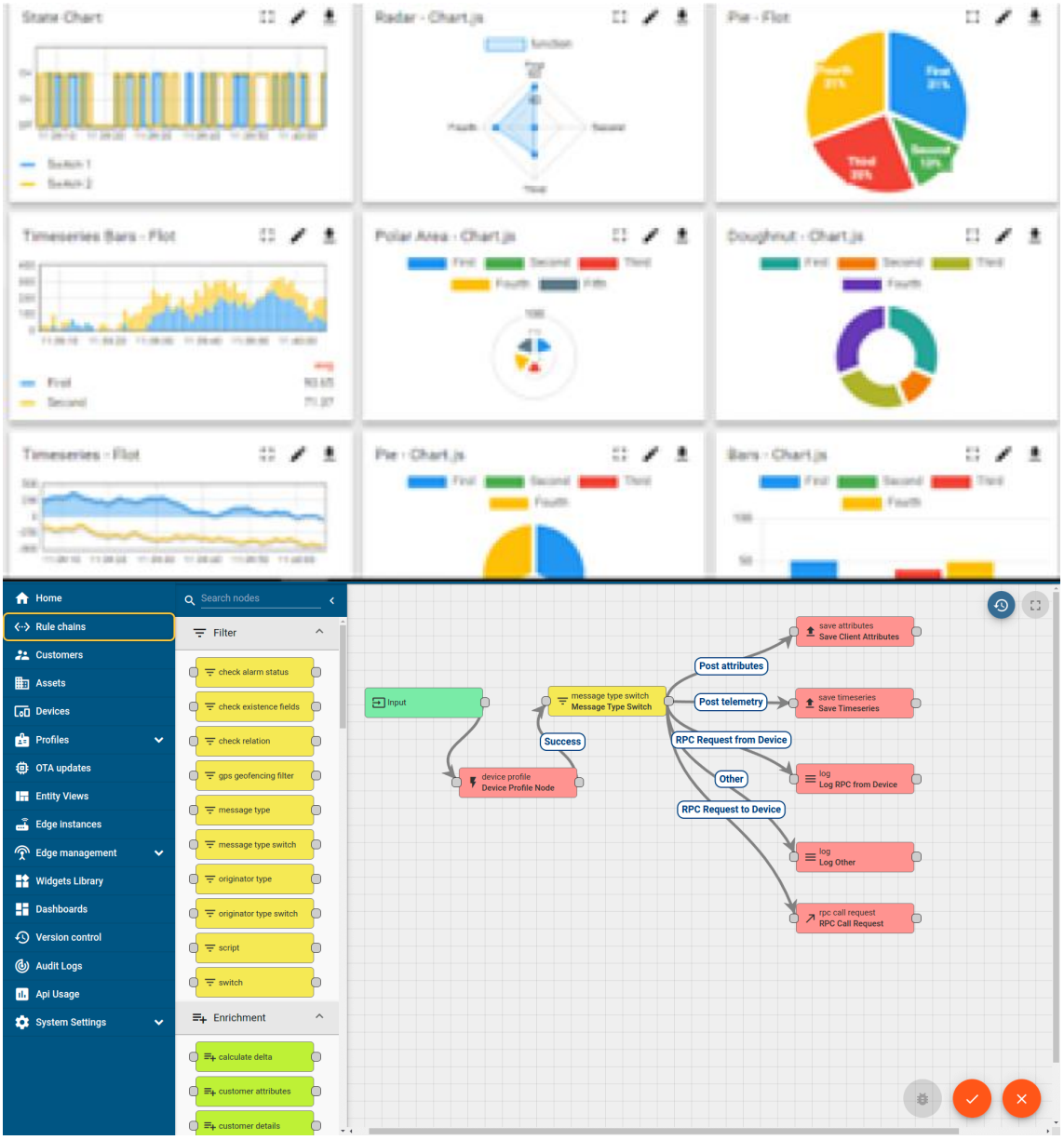
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
- It supports both cloud and on-premises deployments.

[

It has features to

- Build Your own dashboard
- Analytics and Reporting
- Alert and Notification
- Integration with third party application(Power BI, SAP, ERP)
- Rule Engine



[

ii. Smart Factory Platform (FACTORY WATCH)

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.

[

Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i

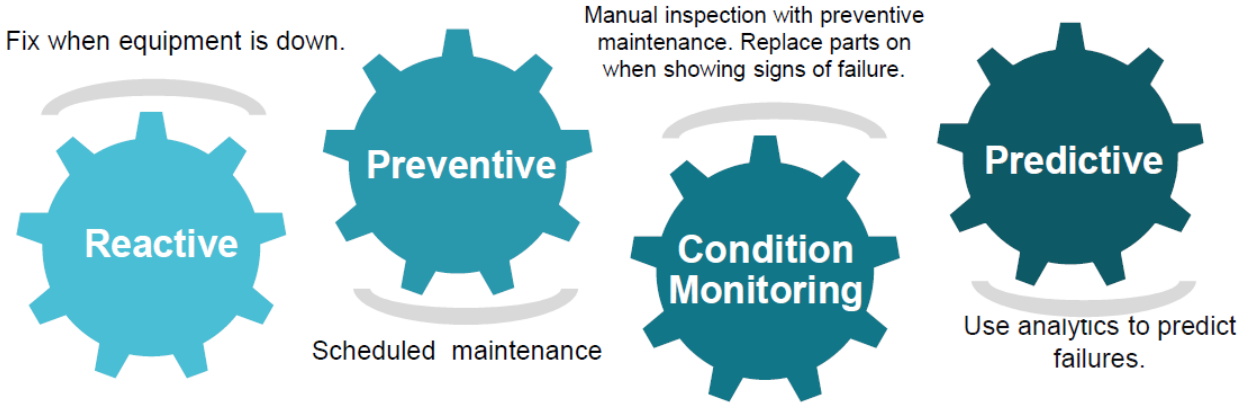
[

iii. LoRaWAN based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

UCT isproviding Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.

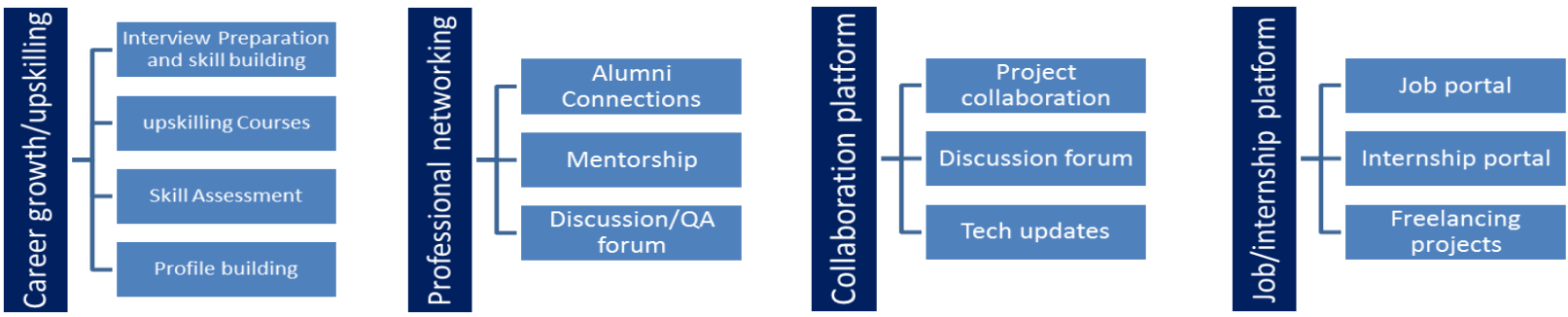


2.1 About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.

[



2.2 The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

[

2.3 Objectives of this Internship program

The objective for this internship program was to

☛ get practical experience of working in the industry.

☛ to solve real world problems.

☛ to have improved job prospects.

☛to have Improved understanding of our field and its applications.

☛to have Personal growth like better communication and problem solving.

[

3 MY PROJECT : URL SHORTENER

First of all, let us understand what is a URL shortener. Users have relied on connections as their primary form of transit since the internet's inception. URL shortening is a Web method for making a Uniform Resource Locator (URL) significantly shorter while still directing to the desired website. This is accomplished by the use of a redirect, which points to a web page with a lengthy URL. They let users navigate from one website to another in a matter of seconds by just clicking on a URL. Unfortunately, connections typically become lengthy and complex when tracking parameters are added and site structures expand to handle hundreds of pages. Long URLs became a concern as social media grew in popularity. Twitter used to limit messages to a maximum of 140 characters, and every character in a link was tallied. As a result, a long URL may take up the entirety of your tweet. URL shortener programs have evolved as a response to this difficulty of sharing.

[

3.1 Benefits of URL shortener

- The importance of content in any marketing plan cannot be overstated. You'll need the correct URLs to distribute that material. An URL shortener guarantees that the relevant messages reach your target audience without taking up too much room in your social media postings.
- Make it simpler for people to share your material by doing the following: Customers may learn everything they need to know about your site by using simple and branded URLs. Reduced URLs with random letter and number combinations are no longer necessary.
- Make your URLs more attractive: Shorter URLs are more appealing from an aesthetic standpoint. Although it may not appear to be significant, a shorter URL may be the deciding factor in encouraging someone to click on your links.
- Allows for traffic monitoring: Bit.ly includes tracking monitors that track your tweet or post's sharing activity over time.

[

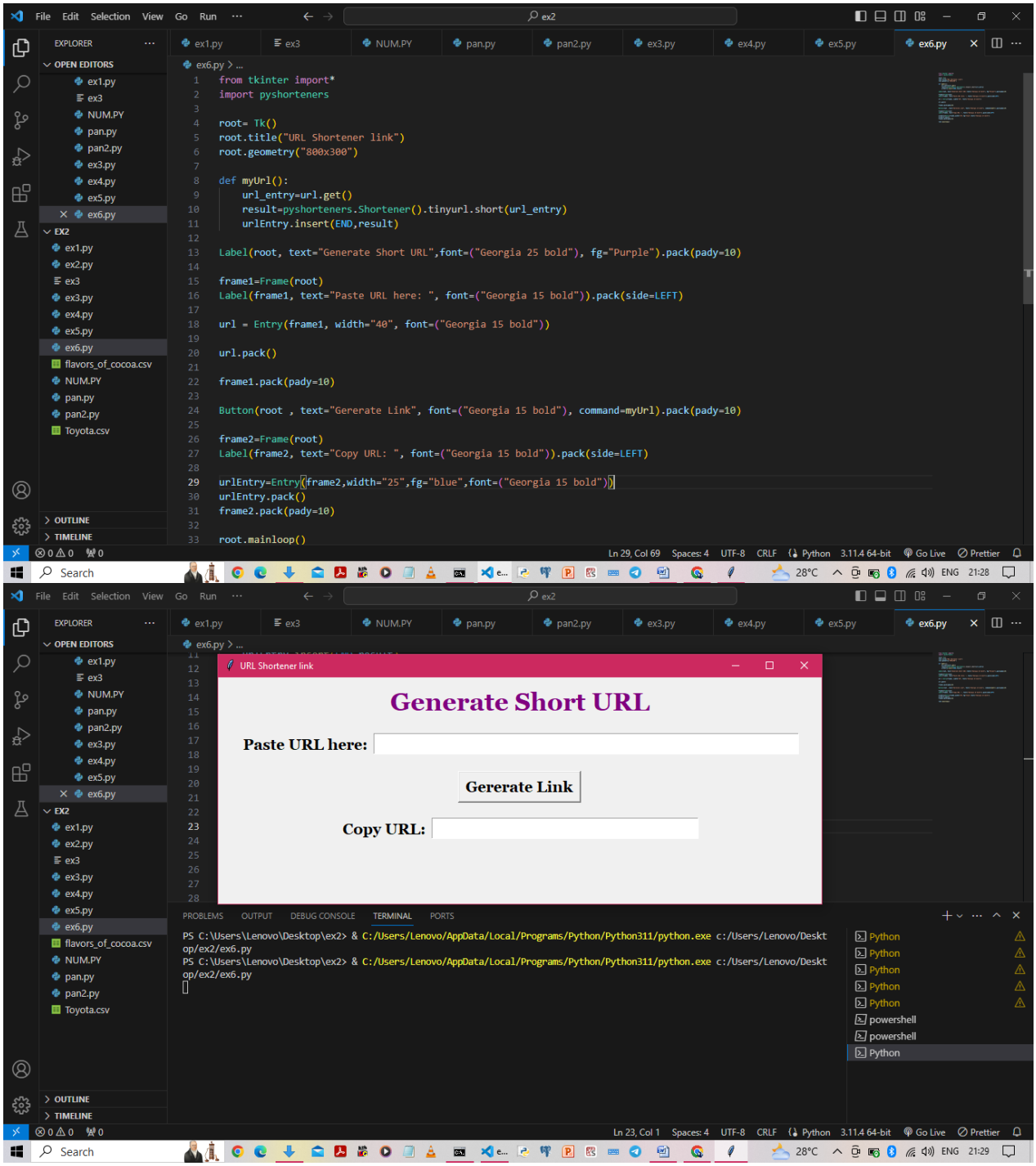
4 PROJECT IMPLEMENTATION :

```
#pip install pyshorteners

from  tkinter import*
import  pyshorteners

root= Tk()
root.title("URL Shortener link")
root.geometry("800x300")

def myUrl():
   url_entry=url.get()
   result=pyshorteners.Shortener().tinyurl.short(url_entry)
   urlEntry.insert(END,result)

Label(root, text="Generate Short URL",font=("Georgia 25 bold"), fg="Purple").pack(pady=10)

frame1=Frame(root)
Label(frame1, text="Paste URL here: ", font=("Georgia 15 bold")).pack(side=LEFT)

url = Entry(frame1, width="40", font=("Georgia 15 bold"))

url.pack()
```

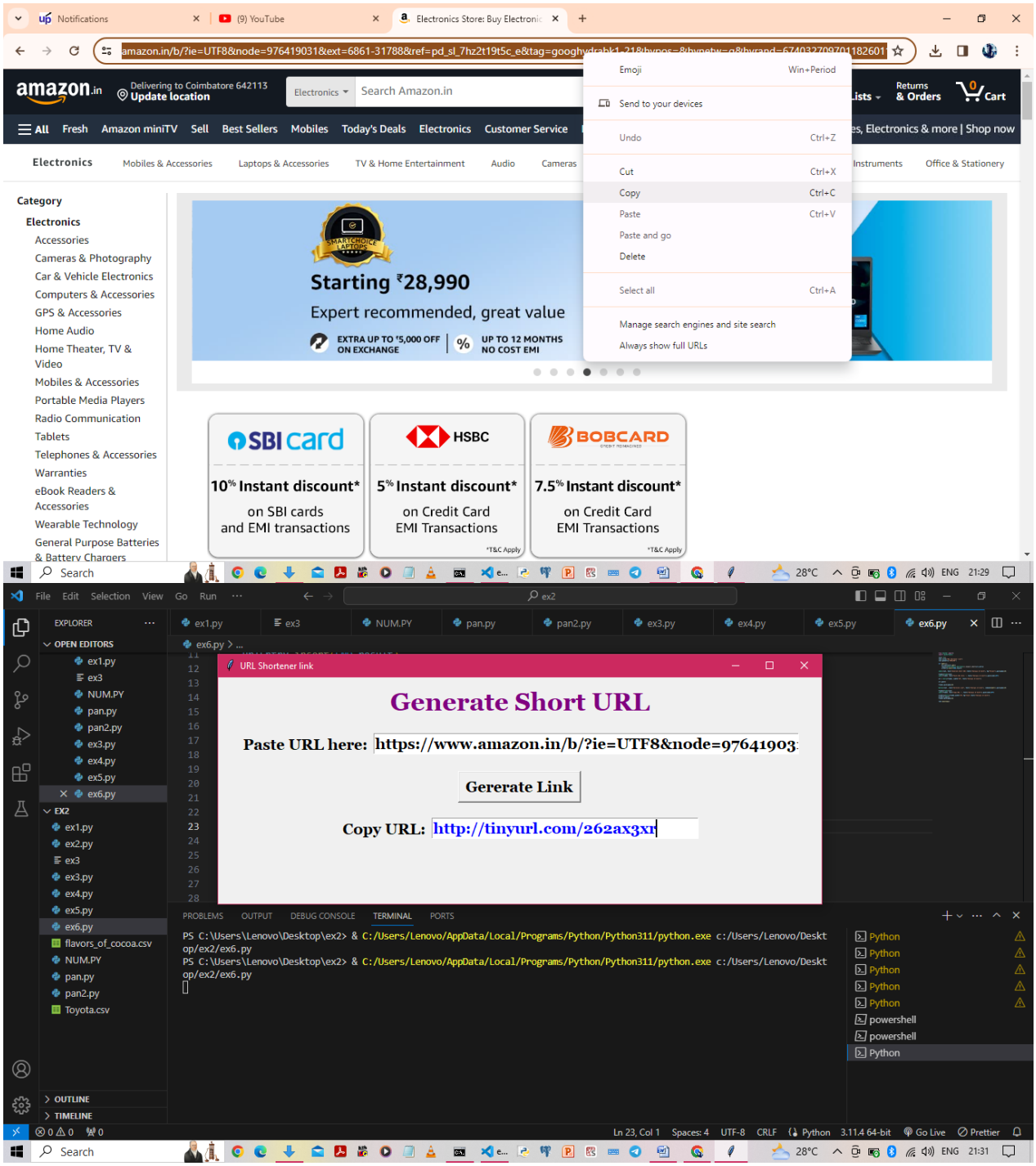
[

```
frame1.pack(pady=10)

Button(root , text="Gererate Link", font=("Georgia 15 bold"), command=myUrl).pack(pady=10)

frame2=Frame(root)
Label(frame2, text="Copy URL: ", font=("Georgia 15 bold")).pack(side=LEFT)

urlEntry=Entry(frame2,width="25",fg="blue",font=("Georgia 15 bold"))
urlEntry.pack()
frame2.pack(pady=10)

root.mainloop()
```

[

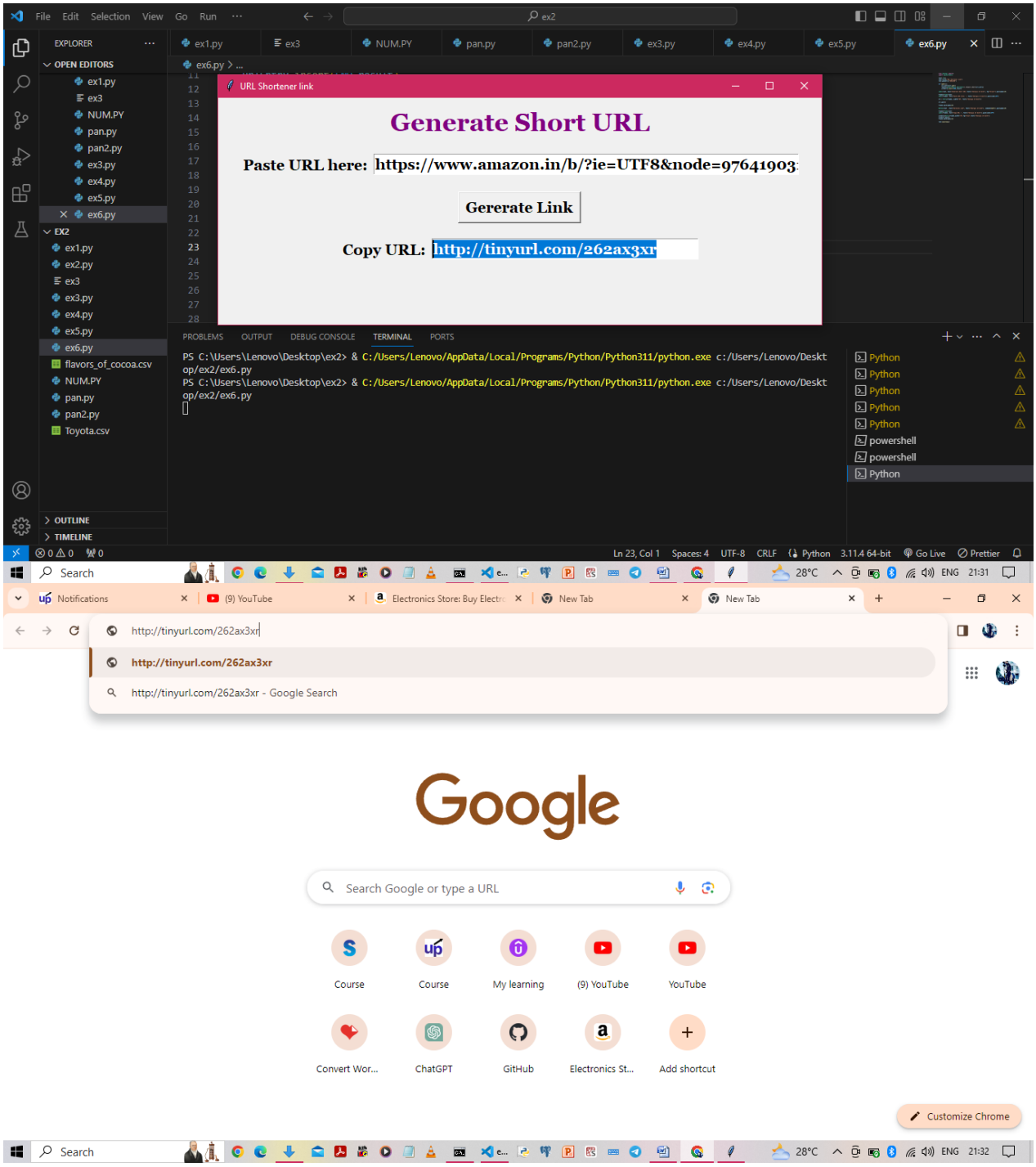

File Edit Selection View Go Run
ex2
EXPLORER
OPEN EDITORS
ex1.py
ex3
NUM.PY
pan.py
pan2.py
ex3.py
ex4.py
ex5.py
ex6.py
EX2
ex1.py
ex2.py
ex3
ex3.py
ex4.py
ex5.py
ex6.py
flavors_of_cocoa.csv
NUM.PY
pan.py
pan2.py
Toyota.csv
OUTLINE
TIMELINE
from tkinter import*
import pyshorteners
root= Tk()
root.title("URL Shortener link")
root.geometry("800x300")
def myUrl():
url_entry=url.get()
result=pyshorteners.Shortener().tinyurl.short(url_entry)
urlEntry.insert(END,result)
Label(root, text="Generate Short URL",font=("Georgia 25 bold"), fg="Purple").pack(pady=10)
frame1=Frame(root)
Label(frame1, text="Paste URL here: ", font=("Georgia 15 bold")).pack(side=LEFT)
url = Entry(frame1, width="40", font=("Georgia 15 bold"))
url.pack()
frame1.pack(pady=10)
Button(root , text="Gererate Link", font=("Georgia 15 bold"), command=myUrl).pack(pady=10)
frame2=Frame(root)
Label(frame2, text="Copy URL: ", font=("Georgia 15 bold")).pack(side=LEFT)
urlEntry=Entry(frame2,width="25",fg="blue",font=("Georgia 15 bold"))
urlEntry.pack()
frame2.pack(pady=10)
root.mainloop()
Ln 29, Col 69 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier
Search
28°C ENG 21:28
URL Shortener link
Generate Short URL
Paste URL here:
Gererate Link
Copy URL:
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Lenovo\Desktop\ex2> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python311/python.exe c:/Users/Lenovo/Desktop/ex2/ex6.py
PS C:\Users\Lenovo\Desktop\ex2> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python311/python.exe c:/Users/Lenovo/Desktop/ex2/ex6.py
Python
Python
Python
Python
Python
powershell
powershell
Python
Ln 23, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier
Search
28°C ENG 21:29

[

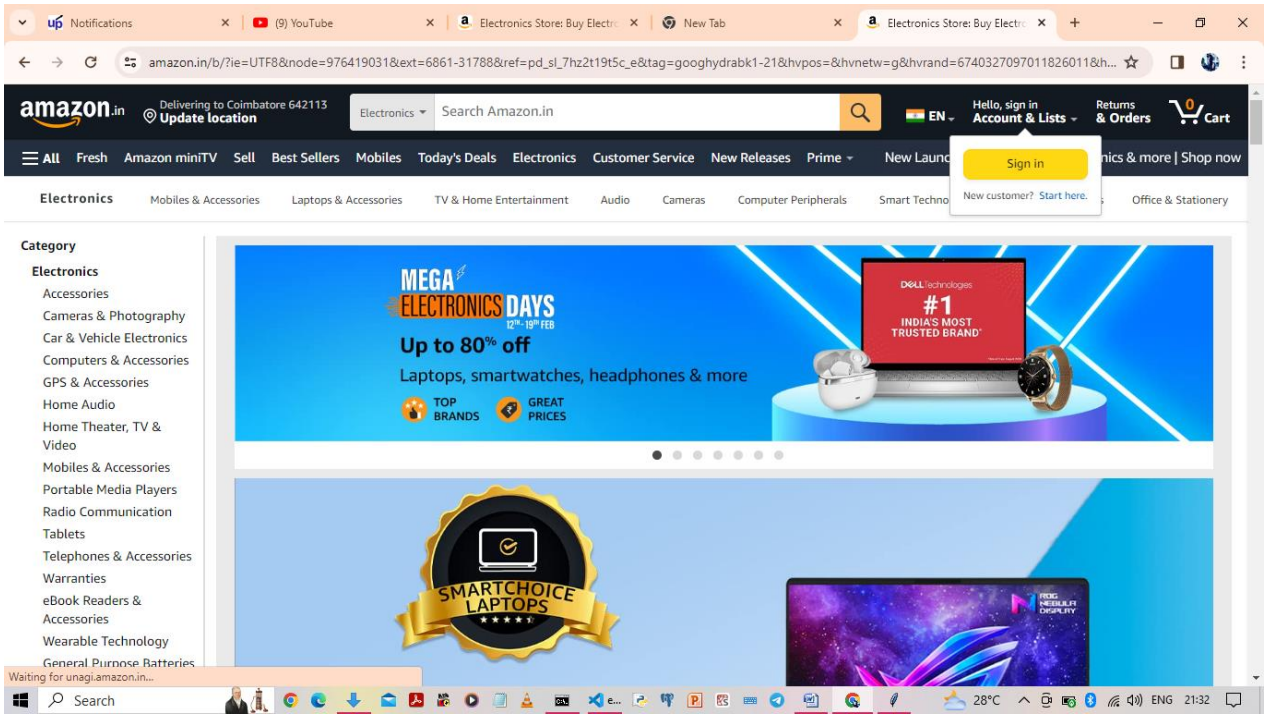

Generate Short URL
Paste URL here: https://www.amazon.in/b/?ie=UTF8&node=97641903
Gererate Link
Copy URL: http://tinyurl.com/262ax3xr

[

[


URL Shortener link
Generate Short URL
Paste URL here: https://www.amazon.in/b/?ie=UTF8&node=97641903
Gererate Link
Copy URL: http://tinyurl.com/262ax3xr
PS C:\Users\Lenovo\Desktop\ex2> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python311/python.exe c:/Users/Lenovo/Desktop/ex2/ex6.py
PS C:\Users\Lenovo\Desktop\ex2> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python311/python.exe c:/Users/Lenovo/Desktop/ex2/ex6.py
http://tinyurl.com/262ax3xr
http://tinyurl.com/262ax3xr - Google Search
Google
Search Google or type a URL
Course
Course
My learning
(9) YouTube
YouTube
Convert Wor...
ChatGPT
GitHub
Electronics St...
Add shortcut
Customize Chrome

[


Notifications
(9) YouTube
Electronics Store: Buy Electro
New Tab
Electronics Store: Buy Electro
amazon.in/b/?ie=UTF8&node=976419031&ext=6861-31788&ref=pd_sl_7hz2t19t5c_e&tag=googhydrabk1-21&hvpos=&hvnetw=g&hvrand=6740327097011826011&h...
amazon.in
Delivering to Coimbatore 642113
Update location
Electronics
Search Amazon.in
EN
Hello, sign in
Account & Lists
Returns
& Orders
Cart
All
Fresh
Amazon miniTV
Sell
Best Sellers
Mobiles
Today's Deals
Electronics
Customer Service
New Releases
Prime
Sign in
New customer? Start here.
Electronics
Mobiles & Accessories
Laptops & Accessories
TV & Home Entertainment
Audio
Cameras
Computer Peripherals
Office & Stationery
Category
Electronics
Accessories
Cameras & Photography
Car & Vehicle Electronics
Computers & Accessories
GPS & Accessories
Home Audio
Home Theater, TV & Video
Mobiles & Accessories
Portable Media Players
Radio Communication
Tablets
Telephones & Accessories
Warranties
eBook Readers & Accessories
Wearable Technology
MEGA ELECTRONICS DAYS
Up to 80% off
Laptops, smartwatches, headphones & more
TOP BRANDS
GREAT PRICES
#1
INDIA'S MOST TRUSTED BRAND
SMARTCHOICE LAPTOPS
Waiting for unagi.amazon.in...
Search
28°C
ENG
21:32

[

5 MY Learnings and Outcomes :

Strengthened understanding of basic programming concepts like variables, data types, loops, and conditional statements.Learned how to structure and organize code for readability and maintainability.Improved proficiency in Python syntax and language features.Acquired knowledge about Python's data structures, including lists, tuples, dictionaries, and sets.Developed skills in debugging Python code and identifying common errors.Gained problem-solving techniques to overcome challenges in coding.Gained hands-on experience working on real-world projects, possibly contributing to the development or enhancement of existing applications.

5.1 Code submission (Github link)

https://github.com/Abdul1103/upskillcampus.git

5.2 Report submission (Github link) :

https://github.com/Abdul1103/upskillcampus.git

[

[

[

[