Name : Husnain

Intern ID : TN/IN01/PY/009

Email ID : husnain45605@gmail.com

Internship Domain : python

Week 6 task

Instructor Name : Hassan ALI

- **Task 1:**

Use math & statistics libraries to get square roots and average.

Description:

This Python script uses the built-in `math` and `statistics` libraries to:

- Calculate the square roots of a list of numbers
- Find the average (mean) of those numbers

It prints the original numbers, their square roots, and the average to the screen.

Output

```python
import math

numbers = [4, 9, 16, 25]

# Calculate square roots
print("Square roots:")
for num in numbers:
    print(f"√{num} = {math.sqrt(num)}")

# Calculate average
average = sum(numbers) / len(numbers)
print(f"\nAverage: {average}")
```

```
PS F:\python> & F:/python/Chtbot/myenv/Scripts/Activate.ps1
(myenv) PS F:\python> & F:/python/Chtbot/myenv/Scripts/python.exe f:/python/week6/task1.py
Square roots:
√4 = 2.0
√9 = 3.0
√16 = 4.0
√25 = 5.0

Average: 13.5
(myenv) PS F:\python>
```

- **Task 2:**

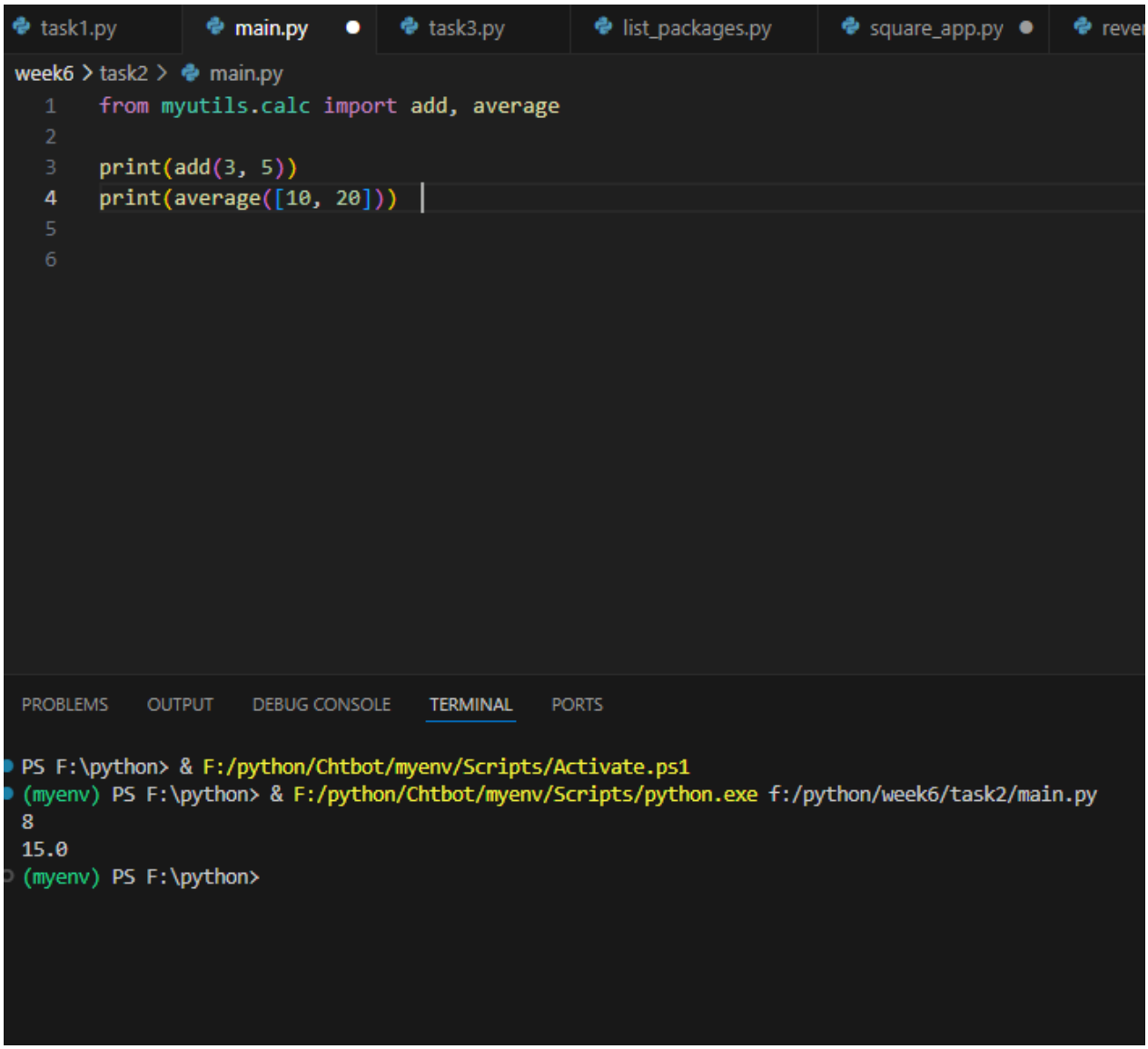
Create a custom package and import it in another script.

Description:

This task shows how to create a custom Python package named `mypackage` with a module `tools.py` that contains a `multiply()` function.

We then import this package in another script (`main.py`, placed outside the package folder) and use the function to perform multiplication and print the result.

Output



- **Task 3:**

Create a virtual environment, install requests & numpy, and print their versions.

description:

This task demonstrates how to set up a Python virtual environment, install external libraries (`requests` and `numpy`), and check their installed versions using a simple Python script.

Output

```
task1.py   main.py   task3.py   list_packages.py   square_app.py   revers.py
week6 > task3.py
import requests
import numpy as np

print(f"Requests version: {requests.__version__}")
print(f"Numpy version: {np.__version__}")
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS F:\python> & F:/python/Chtbot/myenv/Scripts/Activate.ps1
(myenv) PS F:\python> & F:/python/Chtbot/myenv/Scripts/python.exe f:/python/week6/task3.py
Requests version: 2.32.4
Numpy version: 2.3.1
(myenv) PS F:\python>
```

- **Task 4:**

Print list of all installed pip packages from Python code. description:

This script uses the module from `setuptools` to list all installed pip packages and their versions in the current Python environment.

Output

task1.py | main.py | task3.py | list_packages.py | squar

week6 > list_packages.py > ...

```python
import importlib.metadata

# Get all installed packages
installed_packages = importlib.metadata.distributions()

# Print name and version
for pkg in installed_packages:
    print(f"{pkg.metadata['Name']}=={pkg.version}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
tomlkit==0.13.3
tqdm==4.67.1
typer==0.16.0
typing_extensions==4.14.1
typing-inspection==0.4.1
tzdata==2025.2
urllib3==2.5.0
uvicorn==0.35.0
websockets==15.0.1
Werkzeug==3.1.3
zstandard==0.23.0
(myenv) PS F:\python>
```

- **Task 5:**

Create Gradio app that takes a number and returns its square.

description:

This is a simple Gradio web app that asks the user to enter a number, then calculates and displays its square when the **"Calculate"** button is clicked.

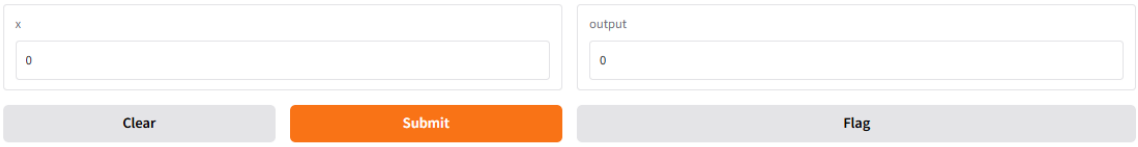
✓ Built with Gradio's `Blocks` layout for a clean and modern interface.
✓ Includes a title and instructions to guide the user.
✓ Shows the result neatly in a textbox below.
✓ Uses a soft theme to make the interface look smooth and user-friendly.

Output

square_app.py

```python
import gradio as gr
# Use one of these proper theme names:
gr.Interface(
    lambda x: x*x,
    gr.Number(),
    "number",

).launch()
```



- **Task 6:**

Create Gradio interface that takes a sentence and returns it reversed. description:

This Gradio app takes a sentence typed by the user and returns it reversed when the **"Reverse"** button is clicked.

✓ Built using Gradio's `Blocks` layout for a clean and modern look.

✓ Includes a textbox for input, a button to trigger the action, and an output box to display the reversed sentence.

✓ Uses a soft theme to keep the interface simple and user-friendly.

output