Day 13

Name: Siddarth
Date : 12-8-2024

Questions:
1.Reverse a string "WorldWord". Hint: :: or join
2.Remove duplicates in ['dog','cat','tiger','dog', 'tiger']
Hint: use set
3. Perform union and intersection using Set
4. Create virtual environment and show installation of package matplotlib and import of modules for visualization.
5. Create a range to display players list within Players class

Answers

1.Reverse a string "WorldWord". Hint: :: or join

Solution:

Description:
This program demonstrates two ways to reverse the string "WorldWord." The first approach uses Python's slicing syntax `[::-1]`, which reverses the string by stepping through it backward. The second approach uses the `join` function in combination with `reversed()`, which creates an iterator that returns the characters of the string in reverse order.

Code and Output:

```
1    # Reversing the string "WorldWord"
2    string = "WorldWord"
3    reversed_string = string[::-1]
4    print("Reversed String:", reversed_string)
5
6    # Alternatively, using join:
7    reversed_string_join = ''.join(reversed(string))
8    print("Reversed String using join:", reversed_string_join)
9
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   PORTS   TERMINAL

```
PS D:\training python> & C:/Users/siddarth.s/AppData/Local/Programs/Python/
"
Reversed String: droWdlroW
Reversed String using join: droWdlroW
PS D:\training python> █
```

2.Remove duplicates in ['dog','cat','tiger','dog', 'tiger']
Hint: use set

Description:

To remove duplicates, we convert the list into a set using `set()`, which inherently removes any duplicate elements since sets cannot have duplicate values. The set is then converted back into a list to preserve the original list format.

Code and Output:

```
 assignment.py > ...
   1     # List with duplicates
   2     animals = ['dog', 'cat', 'tiger', 'dog', 'tiger']
   3
   4     # Removing duplicates using set
   5     unique_animals = list(set(animals))
   6     print("Unique Animals:", unique_animals)
   7     |
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    PORTS    TERMINAL

PS D:\training python> & C:/Users/siddarth.s/AppData/Local/Pr
"
Unique Animals: ['tiger', 'dog', 'cat']
PS D:\training python>
```

3. Perform union and intersection using Set

Description:
This program demonstrates set operations in Python using two sets of names.
The union operation combines the elements of both sets, removing any
duplicates, to create a set that contains all unique elements. The intersection
operation finds common elements between the two sets.

Code and Output:

```
 assignment.py > ...
   1     # Example sets
   2     set1 = {'Siddarth', 'Dhanapal', 'Sownthari'}
   3     set2 = {'Vijay', 'Sanjit', 'Siddarth'}
   4
   5     # Union of sets
   6     union_set = set1.union(set2)
   7     print("Union of sets:", union_set)
   8
   9     # Intersection of sets
  10     intersection_set = set1.intersection(set2)
  11     print("Intersection of sets:", intersection_set)
  12
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    PORTS    TERMINAL

PS D:\training python> & C:/Users/siddarth.s/AppData/Local/Programs/Pyth
"
Union of sets: {'Dhanapal', 'Vijay', 'Sanjit', 'Sownthari', 'Siddarth'}
Intersection of sets: {'Siddarth'}
PS D:\training python>
```

4. Create virtual environment and show installation of package matplotlib and import of modules for visualization.

Description:

This part explains how to create a Python virtual environment, which is an isolated environment that allows you to manage dependencies for your projects separately. After creating and activating the environment, the guide walks through installing the `matplotlib` library, which is commonly used for creating visualizations. The example code then demonstrates how to import `matplotlib.pyplot` and create a simple plot.

Code and Output:

```
C:\Users\siddarth.s\Documents\Training>mkdir python

C:\Users\siddarth.s\Documents\Training>cd python

C:\Users\siddarth.s\Documents\Training\python>python -m venv venv

C:\Users\siddarth.s\Documents\Training\python>cd venv

C:\Users\siddarth.s\Documents\Training\python\venv>.\Scripts\activate

(venv) C:\Users\siddarth.s\Documents\Training\python\venv>pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.9.1.post1-cp312-cp312-win_amd64.whl.metadata (11 kB)
Collecting contourpy>=1.0.1 (from matplotlib)
  Downloading contourpy-1.2.1-cp312-cp312-win_amd64.whl.metadata (5.8 kB)
Collecting cycler>=0.10 (from matplotlib)
  Downloading cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Downloading fonttools-4.53.1-cp312-cp312-win_amd64.whl.metadata (165 kB)
     ---------------------------------------- 165.9/165.9 kB 766.9 kB/s eta 0:00:00
Collecting kiwisolver>=1.3.1 (from matplotlib)
  Downloading kiwisolver-1.4.5-cp312-cp312-win_amd64.whl.metadata (6.5 kB)
Collecting numpy>=1.23 (from matplotlib)
  Downloading numpy-2.0.1-cp312-cp312-win_amd64.whl.metadata (60 kB)
     ---------------------------------------- 60.9/60.9 kB 1.1 MB/s eta 0:00:00
Collecting packaging>=20.0 (from matplotlib)
  Downloading packaging-24.1-py3-none-any.whl.metadata (3.2 kB)
Collecting pillow>=8 (from matplotlib)
  Downloading pillow-10.4.0-cp312-cp312-win_amd64.whl.metadata (9.3 kB)
Collecting pyparsing>=2.3.1 (from matplotlib)
  Downloading pyparsing-3.1.2-py3-none-any.whl.metadata (5.1 kB)
Collecting python-dateutil>=2.7 (from matplotlib)
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting six>=1.5 (from python-dateutil>=2.7->matplotlib)
  Downloading six-1.16.0-py2.py3-none-any.whl.metadata (1.8 kB)
Downloading matplotlib-3.9.1.post1-cp312-cp312-win_amd64.whl (8.0 MB)
     ---------------------------------------- 8.0/8.0 MB 1.9 MB/s eta 0:00:00
```

```python
import matplotlib.pyplot as plt
import numpy as np

def simple_bar_chart():
    categories = ['A', 'B', 'C', 'D']
    values = [10, 17, 25, 4]
    colors = ['#ff9999','#66b3ff','#99ff99','#ffcc99']
    bars = plt.bar(categories, values, color=colors, edgecolor='black', linewidth=2)
    plt.title('Bar Chart', fontsize=16, fontweight='bold')
    plt.xlabel('Categories', fontsize=14)
    plt.ylabel('Values', fontsize=14)
    plt.grid(axis='y', linestyle='--', alpha=0.7)
    for bar in bars:
        yval = bar.get_height()
        plt.text(bar.get_x() + bar.get_width()/2, yval + 0.5, yval, ha='center', fontsize=12, fontweight='bold')
    plt.xticks(fontsize=12, fontweight='bold')
    plt.yticks(np.arange(0, 31, 5), fontsize=12, fontweight='bold')
    plt.savefig('wonderful_bar_chart.png', dpi=300, bbox_inches='tight')
    plt.show()

if __name__ == "__main__":
    simple_bar_chart()
```
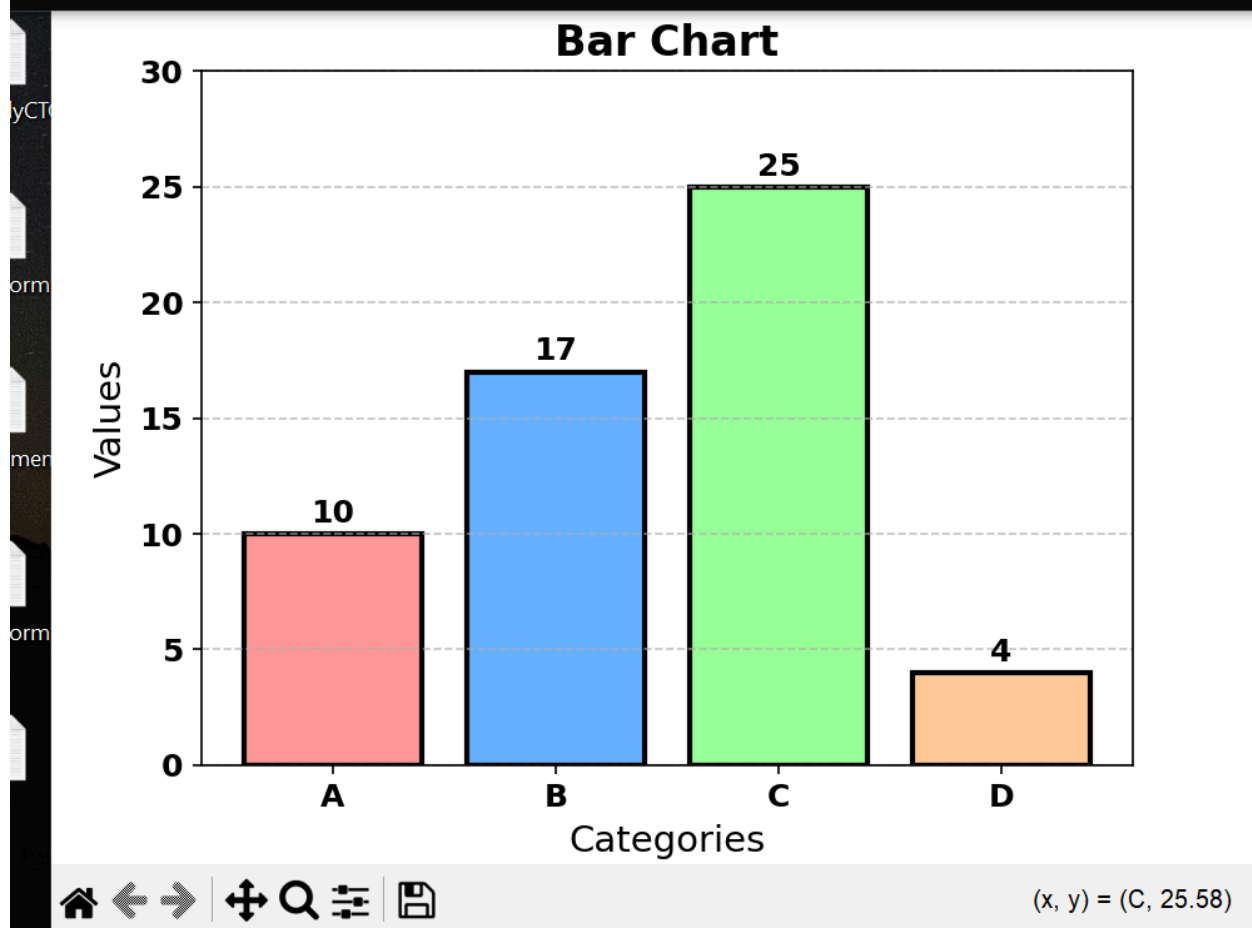
5. Create a range to display players list within Players class

Description:
This program defines a `Players` class that accepts a list of player names upon initialization. The `display_players` method iterates over the list and prints each player's name. The example creates an instance of the `Players` class with the list `["Siddarth", "Dhanapal", "Sownthari", "Sanjit", "Vijay"]` and calls the `display_players` method to display each player's name.

Code and Output:

```python
class Players:
    def __init__(self, players_list):
        self.players_list = players_list

    def display_players(self):
        for player in self.players_list:
            print(player)

# Creating an instance of Players class and displaying players
players = Players(["Siddarth", "Dhanapal", "Sownthari", "Sanjit", "Vijay"])
players.display_players()
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   PORTS   TERMINAL

```
PS D:\training python> & C:/Users/siddarth.s/AppData/Local/Programs/Python/Python312/pyt
ment.py"
Siddarth
Dhanapal
Sownthari
Sanjit
Vijay
PS D:\training python>
```