

PYTHON BASIC PROJECTS WITH SOLUTIONS

1. Password Generator

Steps:

1. Import required libraries: random and string.
2. Define password length.
3. Create a character pool (uppercase, lowercase, digits, symbols).
4. Randomly select characters from the pool.
5. Generate and display the password.

Code:

```
main.py
1 import random
2 import string
3
4 def generate_password(length=12):
5     characters = string.ascii_letters + string.digits + string.punctuation
6     password = ''.join(random.choice(characters) for _ in range(length))
7     return password
8
9 print("Generated Password:", generate_password(12))
10
```

Output:

```
Generated Password: q["HFBy|}0<1
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```

2. To-Do List (CLI)

Steps:

1. Create a list to store tasks.
2. Provide options: Add, View, Remove, Exit.
3. Loop until the user exits.

Code:

```
main.py
1 tasks = []
2 while True:
3     print("\n1. Add Task\n2. View Tasks\n3. Remove Task\n4. Exit")
4     choice = input("Enter choice: ")
5     if choice == "1":
6         task = input("Enter task: ")
7         tasks.append(task)
8         print("Task added!")
9     elif choice == "2":
10        print("\nTo-Do List:")
11        for idx, task in enumerate(tasks, 1):
12            print(f"{idx}. {task}")
13    elif choice == "3":
14        task_num = int(input("Enter task number to remove: "))
15        if 0 < task_num <= len(tasks):
16            tasks.pop(task_num - 1)
17            print("Task removed!")
18    elif choice == "4":
19        break
20    else:
21        print("Invalid choice. Try again.")
22
```

Output:

```
input
+ 1. Add Task
  2. View Tasks
e. 3. Remove Task
   4. Exit
Enter choice: 1
Enter task: eating
Task added!

1. Add Task
2. View Tasks
3. Remove Task
4. Exit
Enter choice: 2

To-Do List:
< 1. eating

1. Add Task
2. View Tasks
3. Remove Task
4. Exit
Enter choice: 3
Enter task number to remove: 1
Task removed!

1. Add Task
2. View Tasks
3. Remove Task
4. Exit
Enter choice: 4
```

3. Weather App (API-based)

Steps:

1. Sign up for OpenWeatherMap API and get an API key.
2. Use requests to fetch weather data.
3. Display temperature, weather condition, and city name.

Code:

```
weather.py > ...
1  import requests
2
3  API_KEY = "192f1fbb096f61772b2aff3b80da26c9"
4  city = input("Enter city name: ")
5  url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={API_KEY}&units=metric"
6  response = requests.get(url).json()
7
8  if response["cod"] == 200:
9      print(f"City: {response['name']}")
10     print(f"Temperature: {response['main']['temp']}°C")
11     print(f"Weather: {response['weather'][0]['description']}")
12 else:
13     print("City not found!")
```

Output:

```
PS C:\Users\289250\Documents\python project> python weather.py
Enter city name: tenkasi
City: Tenkasi
Temperature: 27.41°C
Weather: broken clouds
PS C:\Users\289250\Documents\python project> |
```

4. Number Guessing Game

Steps:

1. Generate a random number between 1-100.
2. Ask the user to guess.
3. Give hints if the guess is too high/low.
4. Continue until guessed correctly.

Code:

```
numbergame.py > ...
1  import random
2
3  number = random.randint(1, 100)
4
5  while True:
6      guess = int(input("Guess the number (1-100): "))
7
8      if guess < number:
9          print("Too low! Try again.")
10     elif guess > number:
11         print("Too high! Try again.")
12     else:
13         print("Congratulations! You guessed it right.")
14         break
15
```

Output:

```
PS C:\Users\289250\Documents\python project> & C:/Users/289250/AppData/Local/Microsoft/WindowsApps/python3.11
.exe "c:/Users/289250/Documents/python project/numbergame.py"
Guess the number (1-100): 77
Too high! Try again.
Guess the number (1-100): 60
Congratulations! You guessed it right.
PS C:\Users\289250\Documents\python project>
```

5. QR Code Generator

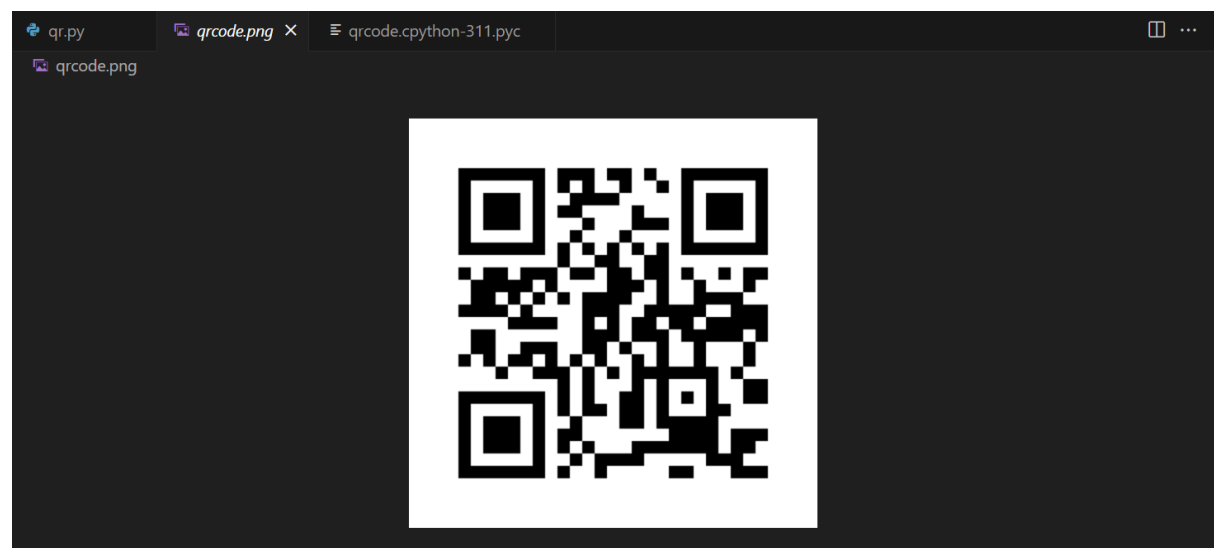
Steps:

1. Install qrcode library (pip install qrcode).
2. Take user input (text/link) to convert.
3. Generate and save the QR code.

Code:

```
qr.py > ...  
1  import qrcode  
2  
3  data = input("Enter text or URL: ")  
4  qr = qrcode.make(data)  
5  qr.save("qrcode.png")  
6  print("QR Code generated and saved as 'qrcode.png'!")  
7
```

Output:



```
PS C:\Users\289250\Documents\python project> python qr.py  
Enter text or URL: www.instagram.com  
Enter text or URL: www.instagram.com  
QR Code generated and saved as 'qrcode.png'!  
PS C:\Users\289250\Documents\python project>
```

Conclusion:

These basic Python projects help in understanding fundamental programming concepts such as loops, conditional statements, APIs, and external library usage. Each project provides hands-on experience with problem-solving using Python.

