**Topics**

1. **Conditional Statements**:
   * Use if, elif, and else to make decisions in a program.
   * Syntax:

python

Copy code

if condition:

# Code block

elif another\_condition:

# Code block

else:

# Code block

1. **Loops**:
   * **For Loop**: Iterates over a sequence (e.g., list, dictionary).

python

Copy code

for item in sequence:

# Code block

* + **While Loop**: Repeats a block of code as long as a condition is True.

python

Copy code

while condition:

# Code block

1. **range() Function**:
   * Generates a sequence of numbers.
   * Example:

python

Copy code

for i in range(5):

print(i) # Outputs 0, 1, 2, 3, 4

**Practical Exercises**

**1. Categorize Servers Using Conditional Statements**

* Objective: Determine server type based on user input.
* Script:

python

Copy code

server\_name = input("Enter the server name: ")

if server\_name.startswith("web"):

print(f"{server\_name} is a Web Server.")

elif server\_name.startswith("db"):

print(f"{server\_name} is a Database Server.")

else:

print(f"{server\_name} is an Unknown Server Type.")

* Test with inputs like web\_server1, db\_server1, or others.

**2. Loop Through a List of Servers**

* Objective: Iterate over a list and display server names.
* Script:

python

Copy code

servers = ["web\_server1", "db\_server1", "cache\_server1"]

for server in servers:

print(f"Checking status for {server}...")

* Output:

lua

Copy code

Checking status for web\_server1...

Checking status for db\_server1...

Checking status for cache\_server1...

**3. Retry Connection Using a While Loop**

* Objective: Simulate retrying a connection until successful.
* Script:

python

Copy code

import random

max\_retries = 5

retries = 0

while retries < max\_retries:

print("Trying to connect...")

success = random.choice([True, False]) # Simulate connection success

if success:

print("Connection Successful!")

break

else:

print("Connection Failed. Retrying...")

retries += 1

else:

print("Max retries reached. Exiting.")

* Run the script multiple times to observe different outcomes.

**4. Generate a List of IP Addresses**

* Objective: Use range() to generate and display a list of sequential IP addresses.
* Script:

python

Copy code

base\_ip = "192.168.1."

for i in range(1, 11):

print(f"{base\_ip}{i}")

* Output:

python

Copy code

192.168.1.1

192.168.1.2

...

192.168.1.10

**5. Advanced Challenge: Monitor Multiple Servers**

* Objective: Create a script that monitors multiple servers and identifies their status.
* Script:

python

Copy code

servers = {

"web\_server1": "up",

"db\_server1": "down",

"cache\_server1": "up"

}

print("Server Status:")

for server, status in servers.items():

if status == "up":

print(f"{server} is operational.")

else:

print(f"{server} is not responding. Investigate!")

* Output:

csharp

Copy code

Server Status:

web\_server1 is operational.

db\_server1 is not responding. Investigate!

cache\_server1 is operational.

**6. Bonus: Interactive Server Query**

* Script:

python

Copy code

servers = ["web\_server1", "db\_server1", "cache\_server1"]

active = True

while active:

print("\nAvailable Servers:", servers)

query = input("Enter a server name to check (or type 'exit' to quit): ")

if query in servers:

print(f"{query} is active and responding.")

elif query == "exit":

print("Exiting server monitor.")

active = False

else:

print("Server not found.")

* Test with server names and the exit command.