Subdomain Status Checker Documentation

Overview:

This Python script checks the status of user-defined subdomains (or URLs) and reports whether they are "UP and Running" or
"Down and Unreachable." Status checks are performed at user-defined intervals, and results are displayed in a tabular format
along with a timestamp indicating when the checks were carried out.

Dependencies:

- requests: Sends HTTP GET requests to check the status of subdomains.
- tabulate: Formats and displays results in a table.
- datetime: Captures the timestamp of each status check.
- time: Manages sleep intervals between status checks.

To install the necessary dependencies, run the following command:

pip install requests tabulate

How the Script Works:

1. User Input:
   - The user specifies the number of subdomains they wish to check.
   - For each subdomain, the full URL is entered (e.g., https://subdomain.example.com).
   - The user defines the time interval (in seconds) between status checks.

2. Status Checking:

- An HTTP GET request is made for each subdomain using requests.get().
- Based on the response:
  - If the status code is 200, the subdomain is considered "UP and Running."
  - Otherwise, it's marked as "Down and Unreachable."
  - If any connection errors occur, the subdomain is also marked as "Down and Unreachable."

3. Displaying Results:

- The script displays the status of each subdomain with the timestamp of the check in a table format using the tabulate library.
- Status checks repeat indefinitely based on the user-specified time interval.

4. Looping:

- The script continuously checks subdomains and displays updated statuses at regular intervals as defined by the user.

Code Explanation:

```
import requests
from tabulate import tabulate
from datetime import datetime
import time
```

requests: Sends HTTP requests.

tabulate: Formats results into a readable table.

datetime: Captures the current time for status logging.

time: Controls the time interval between status checks.

Function Definitions:

subdomains_status():

This function checks the status of each subdomain and displays the results in a table format.

- A list, subdomains_status_list, holds the status and timestamp of each subdomain.
- The current timestamp is captured using datetime.now().strftime('%Y-%m-%d %H:%M:%S').
- An HTTP GET request is made for each subdomain. Based on the response, the subdomain is marked either "UP and Running"
or "Down and Unreachable."
- The results are displayed in a formatted table using the tabulate library.

```
def subdomains_status():
    subdomains_status_list = []
    current_time = datetime.now().strftime('%Y-%m-%d %H:%M:%S')

    for subdomain in subdomains:
        try:
            response = requests.get(subdomain, timeout=5)
            if response.status_code == 200:
                subdomains_status_list.append([subdomain, "UP and Running", current_time])
            else:
                subdomains_status_list.append([subdomain, "Down and Unreachable", current_time])
```

```
        except requests.RequestException:
            subdomains_status_list.append([subdomain, "Down and Unreachable", current_time])

            print(tabulate(subdomains_status_list, headers=["Subdomain", "Status", "Timestamp"],
tablefmt="grid", numalign="center"))
```

Main Program Flow:

1. Input Collection:
   - The user specifies the number of subdomains and enters each one. The subdomains are stored in a list.

```
if __name__ == '__main__':
    No_of_domains = int(input('Enter the number of domains: '))
    for i in range(No_of_domains):
        usersubdomain = input("Enter the main domain (e.g., https://subdomain.example.com): ")
        subdomains.append(usersubdomain)
```

2. Time Interval Input:
   - The user inputs the interval in seconds between status checks.

```
time_int = int(input("Please enter the time interval in seconds: "))
```

3. Continuous Status Checking:
   - The script enters a loop where it checks each subdomain and waits for the specified time interval before checking again.

```
while True:
    print("\nChecking subdomains status...\n")
    subdomains_status()
    print(f"\nNext check in {time_int} seconds...")
    time.sleep(time_int)
```

Sample Execution:

1. User Input:

```
Enter the number of domains: 2
Enter the main domain (e.g., https://subdomain.example.com): https://www.example.com
Enter the main domain (e.g., https://subdomain.example.com): https://api.example.com
Please enter the time interval in seconds: 60
```

2. Output:

```
Checking subdomains status...

+------------------------+----------------------+--------------------+
|       Subdomain        |        Status        |      Timestamp     |
+------------------------+----------------------+--------------------+
| https://www.example.com  | UP and Running       | 2024-10-02 12:45:30  |
| https://api.example.com  | Down and Unreachable  | 2024-10-02 12:45:30  |
+------------------------+----------------------+--------------------+

Next check in 60 seconds...
```