Hello, HTTP Server 🌐

Goal

In this exercise, we'll create a simple HTTP server using Python that responds to GET requests with a basic "Hello, World!" message.

Background

HTTP (Hypertext Transfer Protocol) is the foundation of communication on the World Wide Web. Servers listen for HTTP requests from clients and send back responses. In this exercise, we'll build a basic HTTP server that listens for incoming GET requests and sends a simple response.

Instructions

1. Create a new Python file called `simple_http_server.py`.
2. Import the necessary module:

```
import socket
```

3. Define a function called `handle_request` that takes a client socket as a parameter. This function will handle incoming client requests.

```
def handle_request(client_socket):
    # Your code here
```

4. Inside the `handle_request` function, receive the client request data using `client_socket.recv(1024).decode()`.
5. Parse the request to get the requested path (e.g., "/", "/hello", etc.).

6. If the requested path is "/", send a response with a "200 OK" status and a simple HTML message:

```
<html>
<body>
    <h1>Welcome to my simple HTTP server!</h1>
</body>
</html>
```

7. If the requested path is anything else, send a response with a "404 Not Found" status and a simple error message.

8. Close the client socket after sending the response.

9. Define a `start_server` function that creates a server socket, binds it to a host and port, and starts listening for incoming connections.

```
def start_server():
    # Your code here
```

10. Inside the `start_server` function, create a loop that accepts client connections, calls the `handle_request` function to handle each request, and then closes the client socket.

11. Call the `start_server` function to start your HTTP server.

Hint: You may use the python file attached "simple*http*server.py" to kick start the exercise

Note

you are **NOT allowed** to use the builtin http.server module in this exercise.

Example Usage and Output

Example of how the HTTP server script should run:

```
python simple_http_server.py
```

Example output of the HTTP server script:

```
Server is running on http://localhost:8000
New connection from ('127.0.0.1', 46353)
.
.
.
```

Once the HTTP server is running, you can visit http://localhost:8000 on your browser to see the Welcome message.

- Try visiting other paths of the website like http://localhost:8000/hello to see the 404 error message

To submit

Submit your code in a python script named "simple*http*server.py"