|  |  |
| --- | --- |
| Ex.No:1 | SIMPLE WEBSERVER |
| Date: |

AIM:

To develop a simple webserver to serve HTML programming pages.

ALGORITHM:

Step 1: Install Visual Studio Code if you haven't already.

Step 2: Install the Python extension for VS Code.

Step 3: Open VS Code and create a new folder for your project.

Step 4: Inside the project folder, create a new file named app.py.

Step 5: Install Flask by opening a terminal in VS Code and running pip

Install flask.

Step 6: Write the following code in app.py:

PROGRAM:

HTML:

<!DOCTYPE html>

<html>

<head>

  <title>Using Python's SimpleHTTPServer Module</title>

  <style>

    #rectangle {

      height: 50px;

      width: 100px;

      background-color: #00f28f;

    }

  </style>

</head>

<body>

  <h2>Rectangle served by SimpleHTTPServer</h2>

  <div id="rectangle"></div>

</body>

</html>

We can use our custom handler to serve it on any path we want. In this example

we'll just serve it on the root path, /:

import http.server

import socketserver

class MyHttpRequestHandler(http.server.SimpleHTTPRequestHandler):

def do\_GET(self):

if self.path == '/':

self.path = 'mywebpage.html'

return http.server.SimpleHTTPRequestHandler.do\_GET(self)

# Create an object of the above class

handler\_object = MyHttpRequestHandler

PORT = 8000

my\_server = socketserver.TCPServer(("", PORT), handler\_object)

print(f"Serving on port {PORT}..")

# Start the server

my\_server.serve\_forever()

OUTPUT:



## RESULT:

Thus the simple webserver was created successfully.