## Simple Web Server

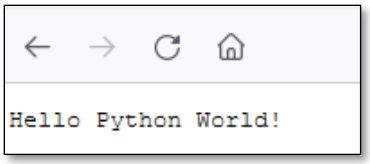
Time required: 30 minutes

- Comment each line of code as shown in the tutorials and other code examples.
- Follow all directions carefully and accurately.
- Think of the directions as minimum requirements.

## Tutorial

```
"""
    Name: simple_web_server.py
    Author:
    Created:
    Purpose: Simple web server
"""

import socket

HOST = "127.0.0.1"
PORT = 80

# Create a socket object
web_socket = socket.socket()
# bind associate the socket with the HOST and PORT
web_socket.bind((HOST, PORT))

# The server is listening on port 80 for TPv4 TCP connections
web_socket.listen()

while True:
    print("Waiting for connection")

    # accept waits for an incoming connection
    # accept creates a socket connection object
    # connection is the client TCP connection
    # address is the incoming client IP address
    connection, address = web_socket.accept()
    print(f"Connection from: {address}")

    # Receive data 1024 bytes at a time
    data = connection.recv(1024)
    # data is in bytes, decode to a string
    data_string = data.decode("utf-8")
    print(data_string)

    # Serve web page, b converts the string into bytes
    connection.send(b"HTTP/1.1 200 OK\r\n\r\nHello Python World!")
    connection.close()
```

Example run:




```
Waiting for connection
Connection from: ('127.0.0.1', 18508)
GET /favicon.ico HTTP/1.1
Accept: */*
UA-CPU: AMD64
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
Trident/7.0; rv:11.0) like Gecko
Host: 127.0.0.1
Connection: Keep-Alive


Waiting for connection
```

Notice that for each request your computer has a different client port.

## Assignment Submission

Attach the pseudocode and program to the assignment. Submit in Blackboard.