Python Socket

👨‍🦱 **Me Pro:** To have socket working, as a newbie what do you think you need first?

👨‍🎓 **Me Newbie:** So for socket, we need a server and a client. The server should always be accepting connections, and when a client connects, the client sends data so the server sees it. So here is the list of things I think I might need:

- **i.** Server listens
- **ii.** Check for new connection and, if a new connection arrives (client), receive the client address (to reply later) and data client sent.

👨‍🦱 **Me Pro:** Nice! So for:

- **"i"** we could do:

  ```
  server.listen()
  ```

- **"ii"** we do:

  ```
  data, addr = socket.accept()
  ```

  We receive data and their address.

👨‍🎓 **Me Newbie:** Easy!

👨‍🦱 **Me Pro:** But you never said you need to create the socket connection, right?

👨‍🎓 **Me Newbie:** Oh yea, so I create a socket connection and open the port and give the port and IP info.

👨‍🦱 **Me Pro:** Yup, you got it.

```
socket = socket(AF_INET, SOCK_STREAM)
socket.bind((SERVER, PORT))
```

✅ Summary Steps

1. **Create socket, open port**

```
socket_server.bind((SERVER, PORT))
```

2. **Start listening**

```
socket_server.listen()
```

3. **Wait for connection and receive data**

```
conn, addr = socket_server.accept()
```

4. **Close the connection**

```
conn.close()
```

Final Code - Server

```
import socket

PORT = 5052
SERVER = socket.gethostbyname(socket.gethostname())

socket_server = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
socket_server.bind((SERVER,PORT)) # 1

socket_server.listen() # 2

while True:
    data, addr = socket_server.accept()
    print(data.recv(5).decode()) # 3
```

👨‍🏫 **Me Pro:** Now what do you think the client need to do.
👨‍🎓 **Me Newbie:** Connect to server and send data?
👨‍🏫 **Me Pro:** Pretty Much!
So, to connect :

```
client.connect(ADDR)
```

And to send data :

```
client.send(message)
```

Final Client

```
import sys
import socket


PORT = 5052
SERVER = "127.0.0.1"

client = socket.socket(socket.AF_INET,socket.SOCK_STREAM
# No need for Binding; The os deals with that

client.connect((SERVER,PORT))

def send(msg):
    message = msg.encode()
    client.send(message) # 2. cliend.send(encoded_message)


send(sys.argv[1])
```

So ,

1. **Create socket, open port**
2. **Start listening**
3. **Wait for connection and receive data**
4. Client connect
5. Client send message