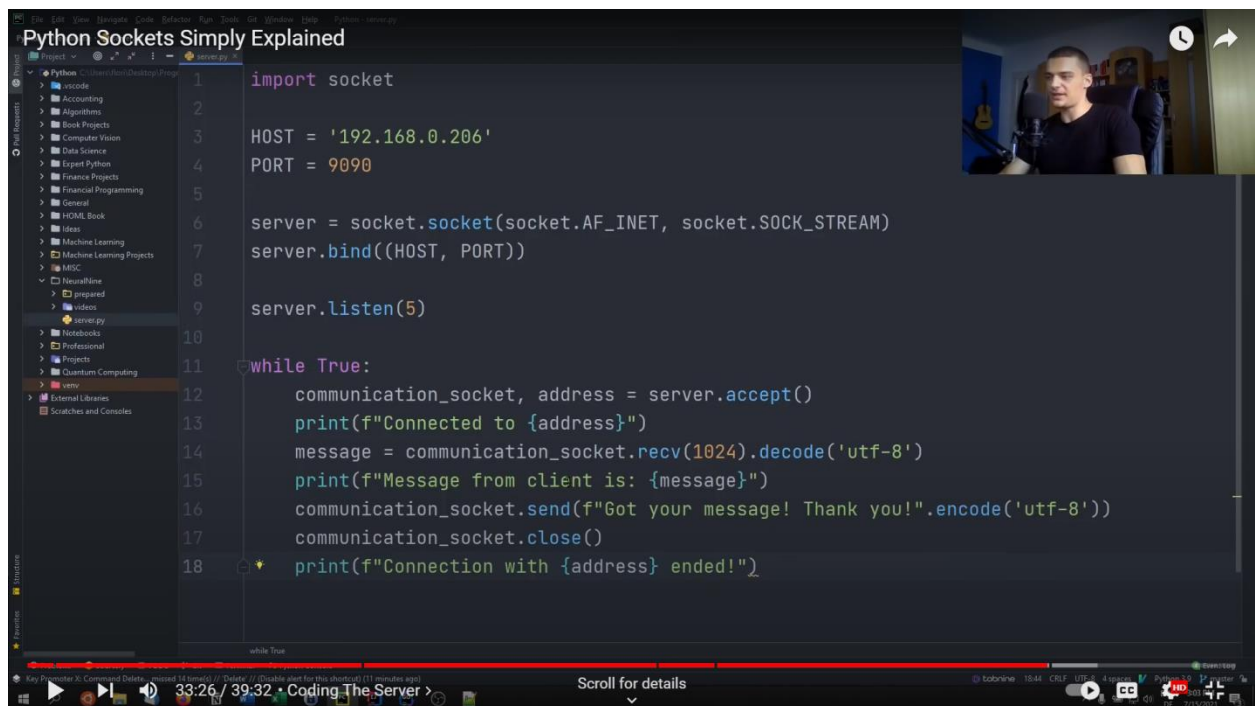


Credits - <https://www.youtube.com/watch?v=YwWfKitB8aA>

Socket Communication in Python

Server.py



```
1 import socket
2
3 HOST = '192.168.0.206'
4 PORT = 9090
5
6 server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7 server.bind((HOST, PORT))
8
9 server.listen(5)
10
11 while True:
12     communication_socket, address = server.accept()
13     print(f"Connected to {address}")
14     message = communication_socket.recv(1024).decode('utf-8')
15     print(f"Message from client is: {message}")
16     communication_socket.send(f"Got your message! Thank you!".encode('utf-8'))
17     communication_socket.close()
18     print(f"Connection with {address} ended!")
```

Client.py

Python Sockets Simply Explained

```
1 import socket
2
3 HOST = '192.168.0.206'
4 PORT = 9090
5
6 socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7 socket.connect((HOST, PORT))
8
9 socket.send("Hello World!".encode('utf-8'))
10 print(socket.recv(1024))
11
```

36:26 / 39:32 • Coding The Client >

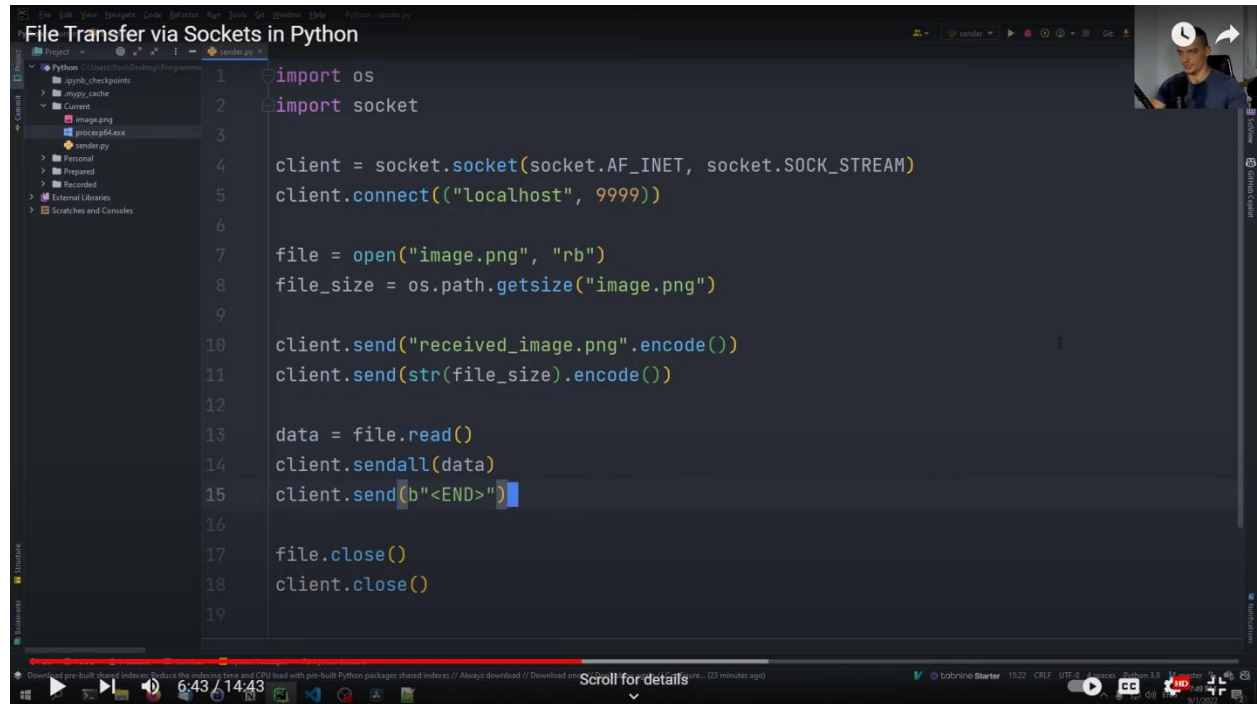
Scroll for details

tabnine 10:22 CRLF UTF-8 4 spaces Python 3.9 master

7/15/2021

File Transfer via Sockets in Python

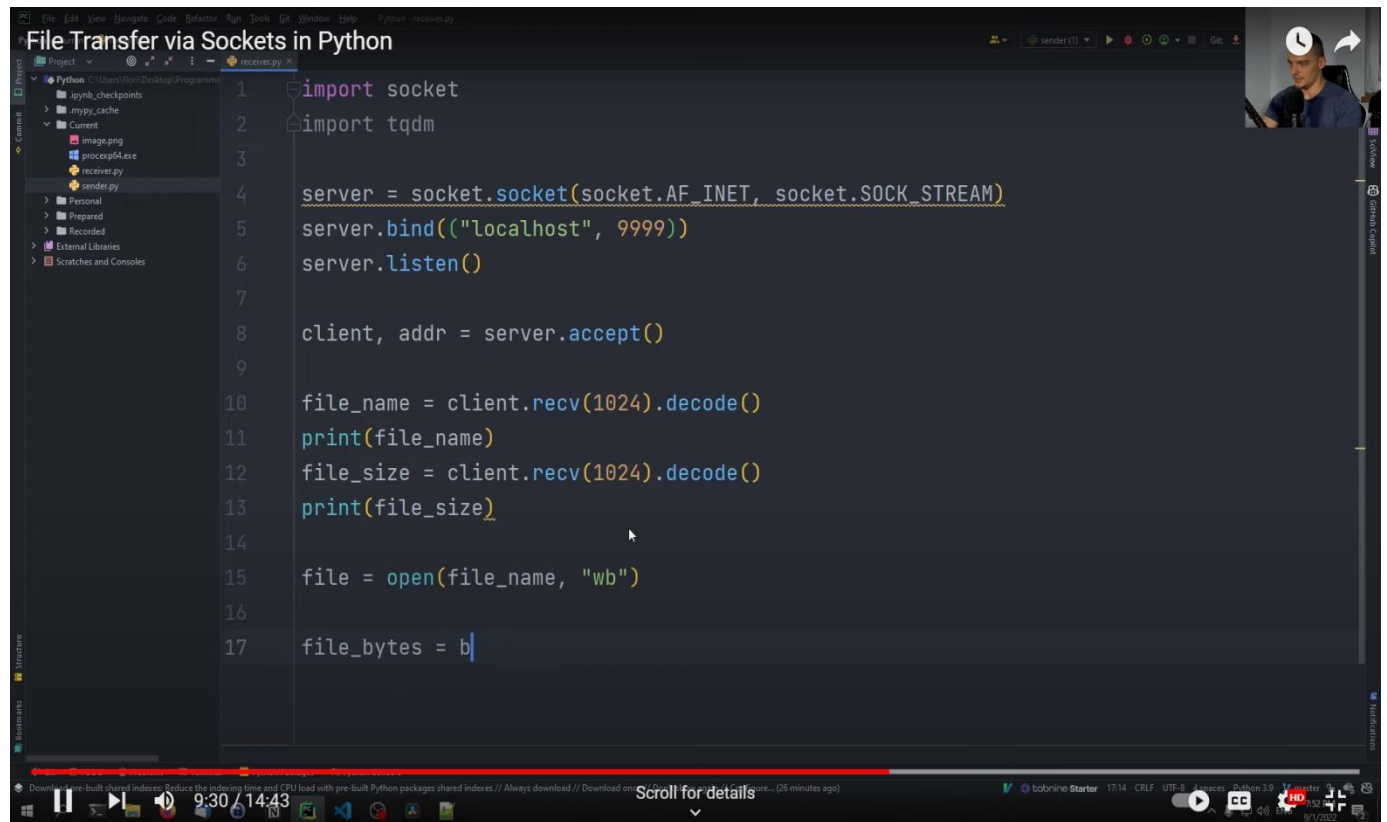
Sender.py

The image is a screenshot of a video player displaying a Python script titled "File Transfer via Sockets in Python". The script is named "sender.py" and is located in a project folder. The code is as follows:

```
1 import os
2 import socket
3
4 client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5 client.connect(("localhost", 9999))
6
7 file = open("image.png", "rb")
8 file_size = os.path.getsize("image.png")
9
10 client.send("received_image.png".encode())
11 client.send(str(file_size).encode())
12
13 data = file.read()
14 client.sendall(data)
15 client.send(b"<END>")
16
17 file.close()
18 client.close()
19
```

The video player interface includes a progress bar at the bottom showing 6:43 / 14:43, a "Scroll for details" button, and a video player control bar with play, pause, and volume icons. A small inset video of the presenter is visible in the top right corner.

Reciever.py



The screenshot shows a video player interface. The main content is a code editor window titled "File Transfer via Sockets in Python" with a dark theme. The code is a Python script for a server that receives files over a socket. The script includes imports for 'socket' and 'tqdm', sets up a server on 'localhost' port 9999, accepts a client connection, receives the file name and size, and opens a file to write the received bytes. The video player has a progress bar at the bottom showing 9:30 / 14:43. In the top right corner of the video, there is a small inset image of a person and a clock icon.

```
1 import socket
2 import tqdm
3
4 server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5 server.bind(("localhost", 9999))
6 server.listen()
7
8 client, addr = server.accept()
9
10 file_name = client.recv(1024).decode()
11 print(file_name)
12 file_size = client.recv(1024).decode()
13 print(file_size)
14
15 file = open(file_name, "wb")
16
17 file_bytes = b|
```

File Transfer via Sockets in Python

```
10 file_name = client.recv(1024).decode()
11 print(file_name)
12 file_size = client.recv(1024).decode()
13 print(file_size)
14
15 file = open(file_name, "wb")
16
17 file_bytes = b""
18
19 done = False
20
21 progress = tqdm.tqdm(unit="B", unit_scale=True, unit_divisor=1000,
22                     total=int(file_size))
23
24 while not done:
25     data = client.recv(1024)
26     |
27
```

10:55 / 14:43

File Transfer via Sockets in Python

```
20
21 progress = tqdm.tqdm(unit="B", unit_scale=True, unit_divisor=1000,
22                     total=int(file_size))
23
24 while not done:
25     data = client.recv(1024)
26     if file_bytes[-5:] == b"<END>":
27         done = True
28     else:
29         file_bytes += data
30     progress.update(1024)
31
32 file.write(file_bytes)
33
34 file.close()
35 client.close()
36 server.close()
```

12:40 / 14:43