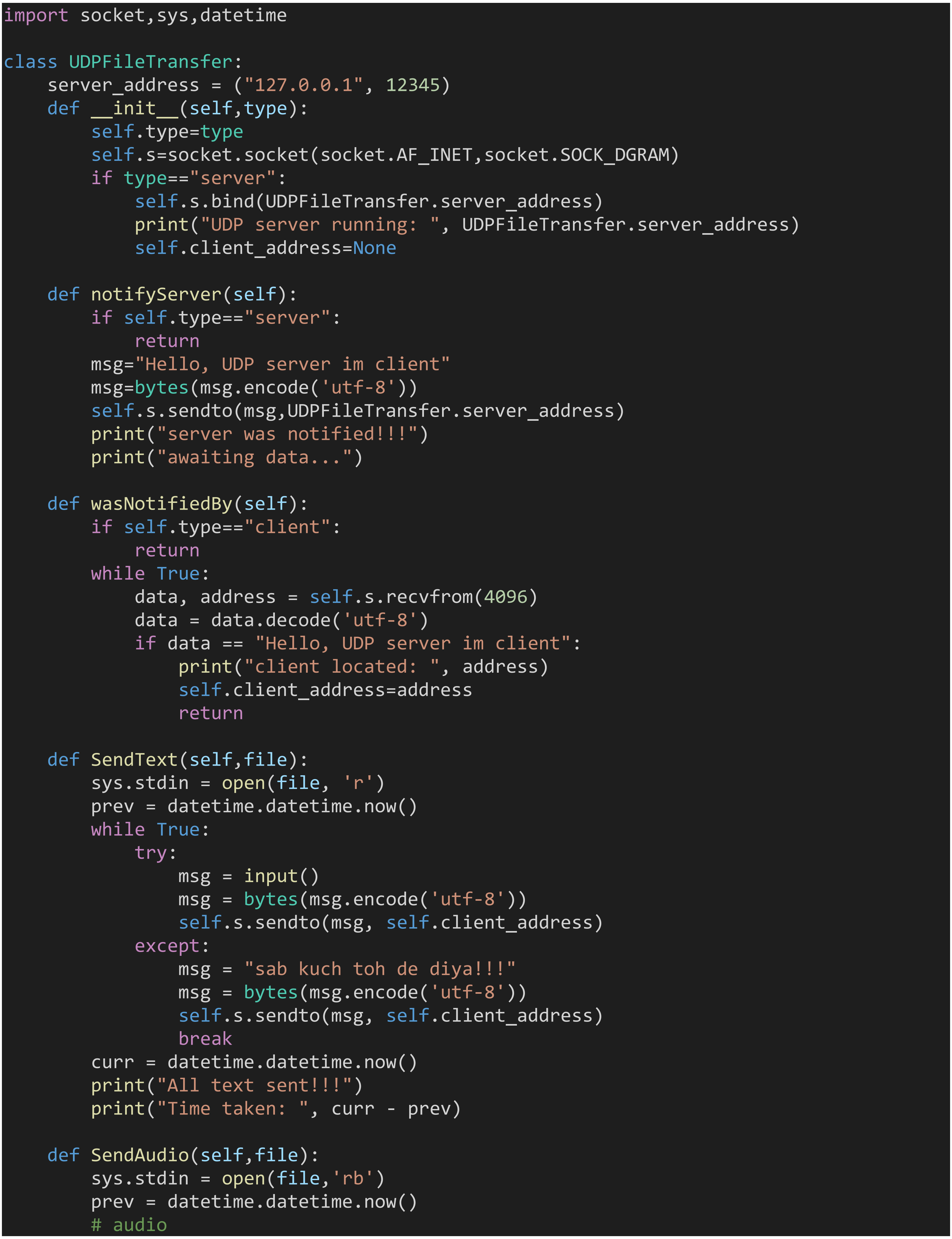
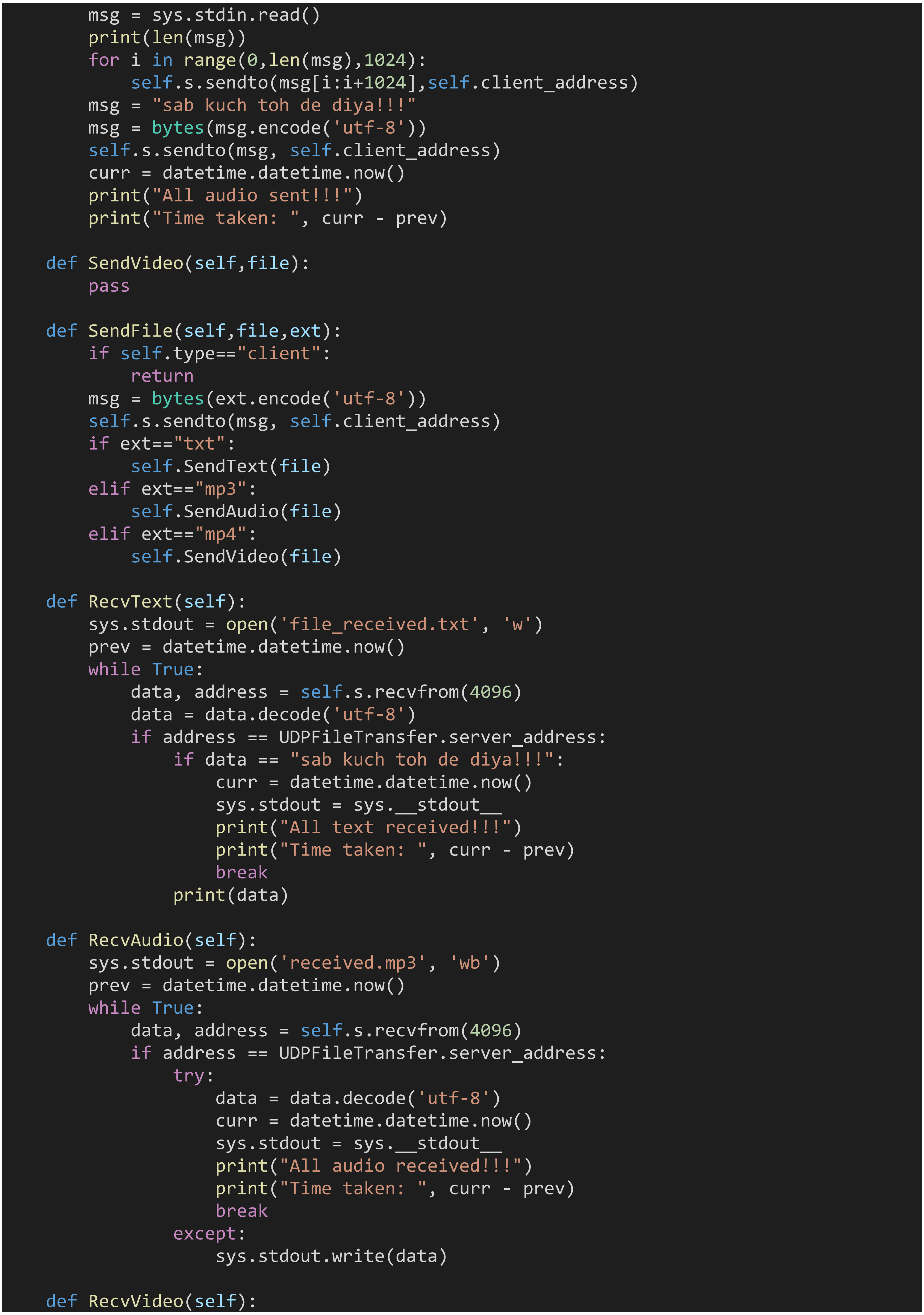
CODE :-

Server.py



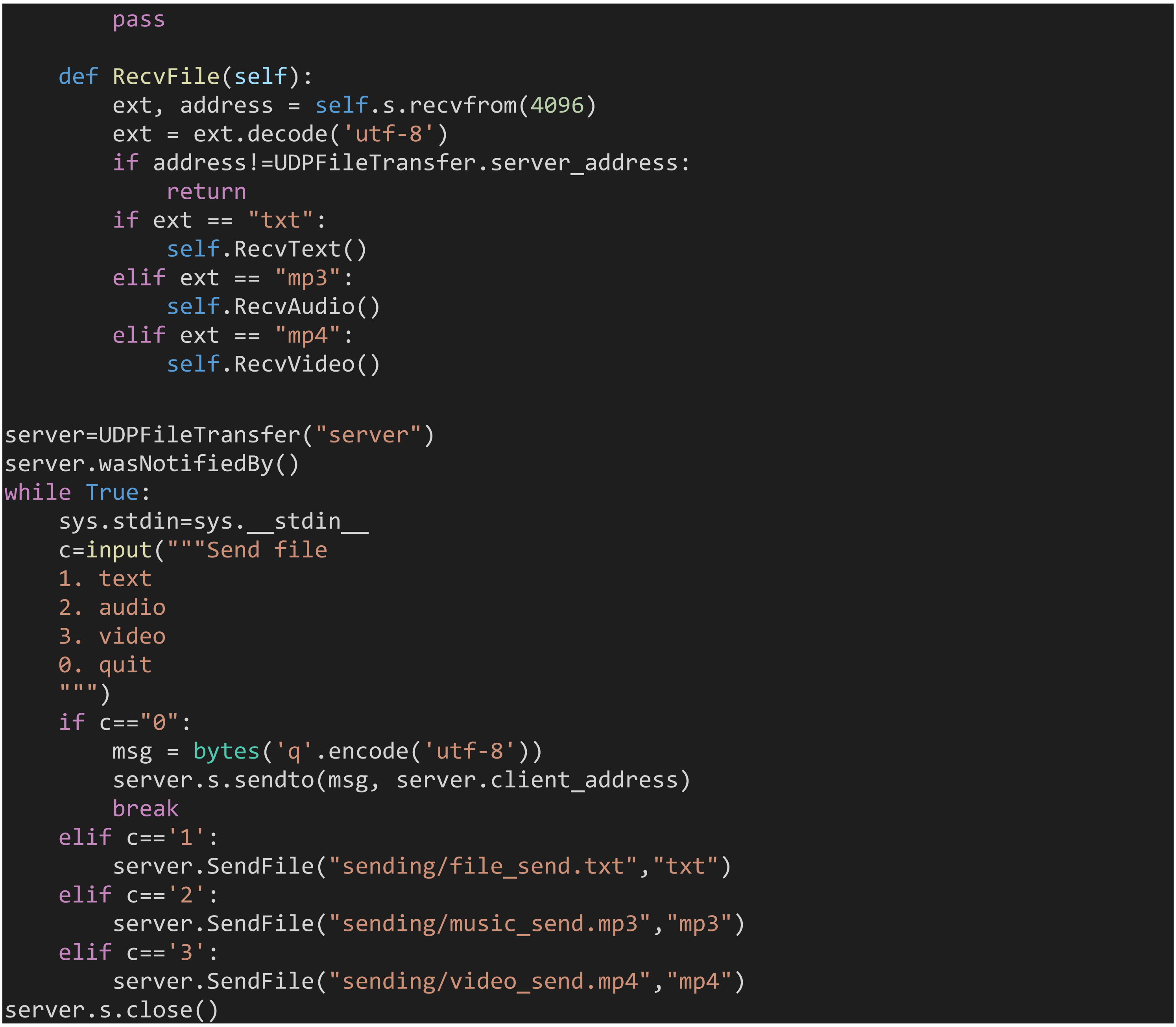
```
import socket,sys,datetime

class UDPFileTransfer:
    server_address = ("127.0.0.1", 12345)
    def __init__(self,type):
        self.type=type
        self.s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
        if type=="server":
            self.s.bind(UDPFileTransfer.server_address)
            print("UDP server running: ", UDPFileTransfer.server_address)
            self.client_address=None

    def notifyServer(self):
        if self.type=="server":
            return
        msg="Hello, UDP server im client"
        msg=bytes(msg.encode('utf-8'))
        self.s.sendto(msg,UDPFileTransfer.server_address)
        print("server was notified!!!")
        print("awaiting data...")

    def wasNotifiedBy(self):
        if self.type=="client":
            return
        while True:
            data, address = self.s.recvfrom(4096)
            data = data.decode('utf-8')
            if data == "Hello, UDP server im client":
                print("client located: ", address)
                self.client_address=address
                return

    def SendText(self,file):
        sys.stdin = open(file, 'r')
        prev = datetime.datetime.now()
        while True:
            try:
                msg = input()
                msg = bytes(msg.encode('utf-8'))
                self.s.sendto(msg, self.client_address)
            except:
                msg = "sab kuch toh de diya!!!"
                msg = bytes(msg.encode('utf-8'))
                self.s.sendto(msg, self.client_address)
                break
        curr = datetime.datetime.now()
        print("All text sent!!!")
        print("Time taken: ", curr - prev)

    def SendAudio(self,file):
        sys.stdin = open(file,'rb')
        prev = datetime.datetime.now()
        # audio
```

Scanned by TapScanner



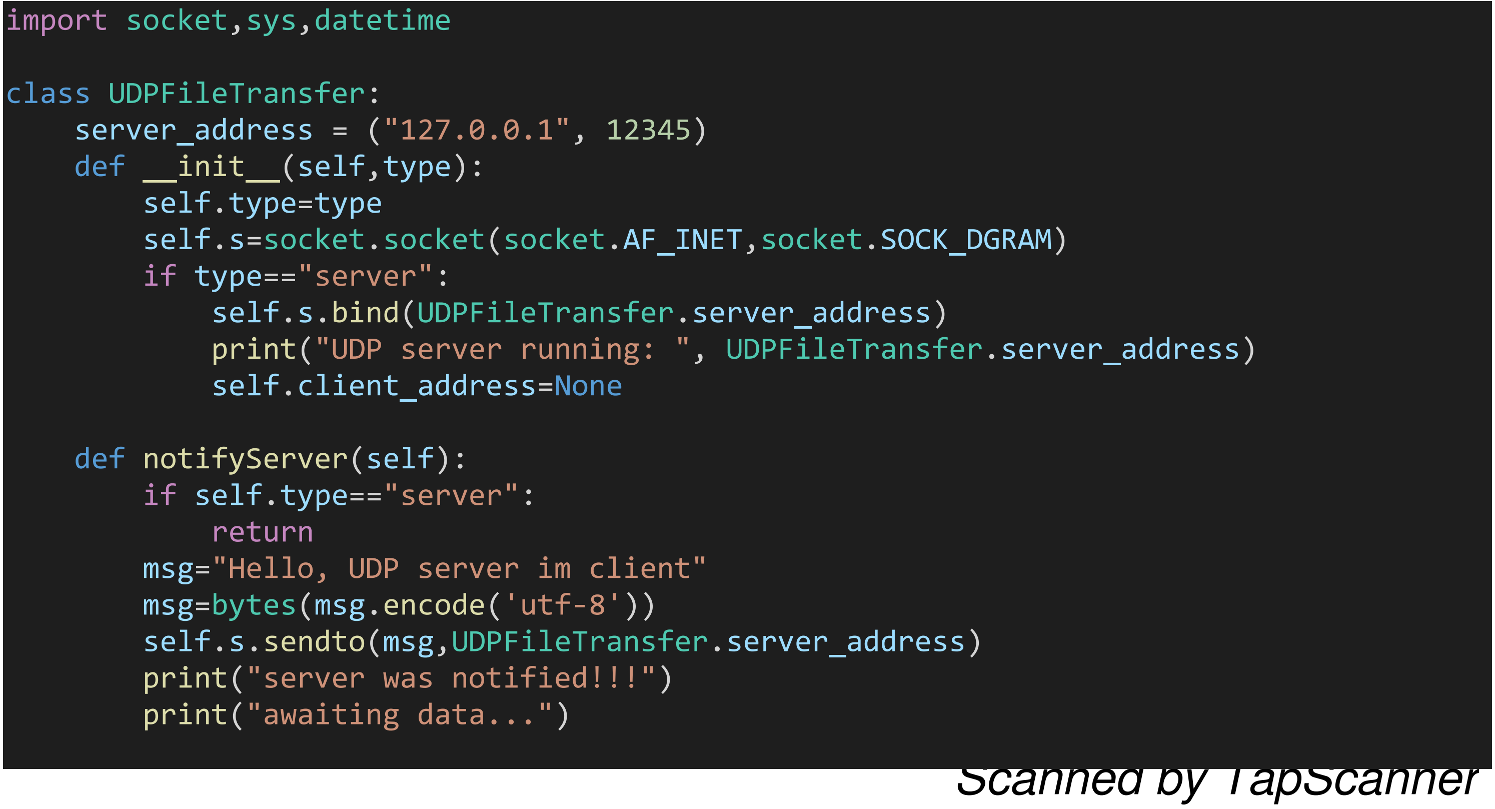
```
        msg = sys.stdin.read()
        print(len(msg))
        for i in range(0,len(msg),1024):
            self.s.sendto(msg[i:i+1024],self.client_address)
        msg = "sab kuch toh de diya!!!"
        msg = bytes(msg.encode('utf-8'))
        self.s.sendto(msg, self.client_address)
        curr = datetime.datetime.now()
        print("All audio sent!!!")
        print("Time taken: ", curr - prev)

    def SendVideo(self,file):
        pass

    def SendFile(self,file,ext):
        if self.type=="client":
            return
        msg = bytes(ext.encode('utf-8'))
        self.s.sendto(msg, self.client_address)
        if ext=="txt":
            self.SendText(file)
        elif ext=="mp3":
            self.SendAudio(file)
        elif ext=="mp4":
            self.SendVideo(file)

    def RecvText(self):
        sys.stdout = open('file_received.txt', 'w')
        prev = datetime.datetime.now()
        while True:
            data, address = self.s.recvfrom(4096)
            data = data.decode('utf-8')
            if address == UDPFileTransfer.server_address:
                if data == "sab kuch toh de diya!!!":
                    curr = datetime.datetime.now()
                    sys.stdout = sys.__stdout__
                    print("All text received!!!")
                    print("Time taken: ", curr - prev)
                    break
                print(data)

    def RecvAudio(self):
        sys.stdout = open('received.mp3', 'wb')
        prev = datetime.datetime.now()
        while True:
            data, address = self.s.recvfrom(4096)
            if address == UDPFileTransfer.server_address:
                try:
                    data = data.decode('utf-8')
                    curr = datetime.datetime.now()
                    sys.stdout = sys.__stdout__
                    print("All audio received!!!")
                    print("Time taken: ", curr - prev)
                    break
                except:
                    sys.stdout.write(data)

    def RecvVideo(self):
```

Scanned by TapScanner



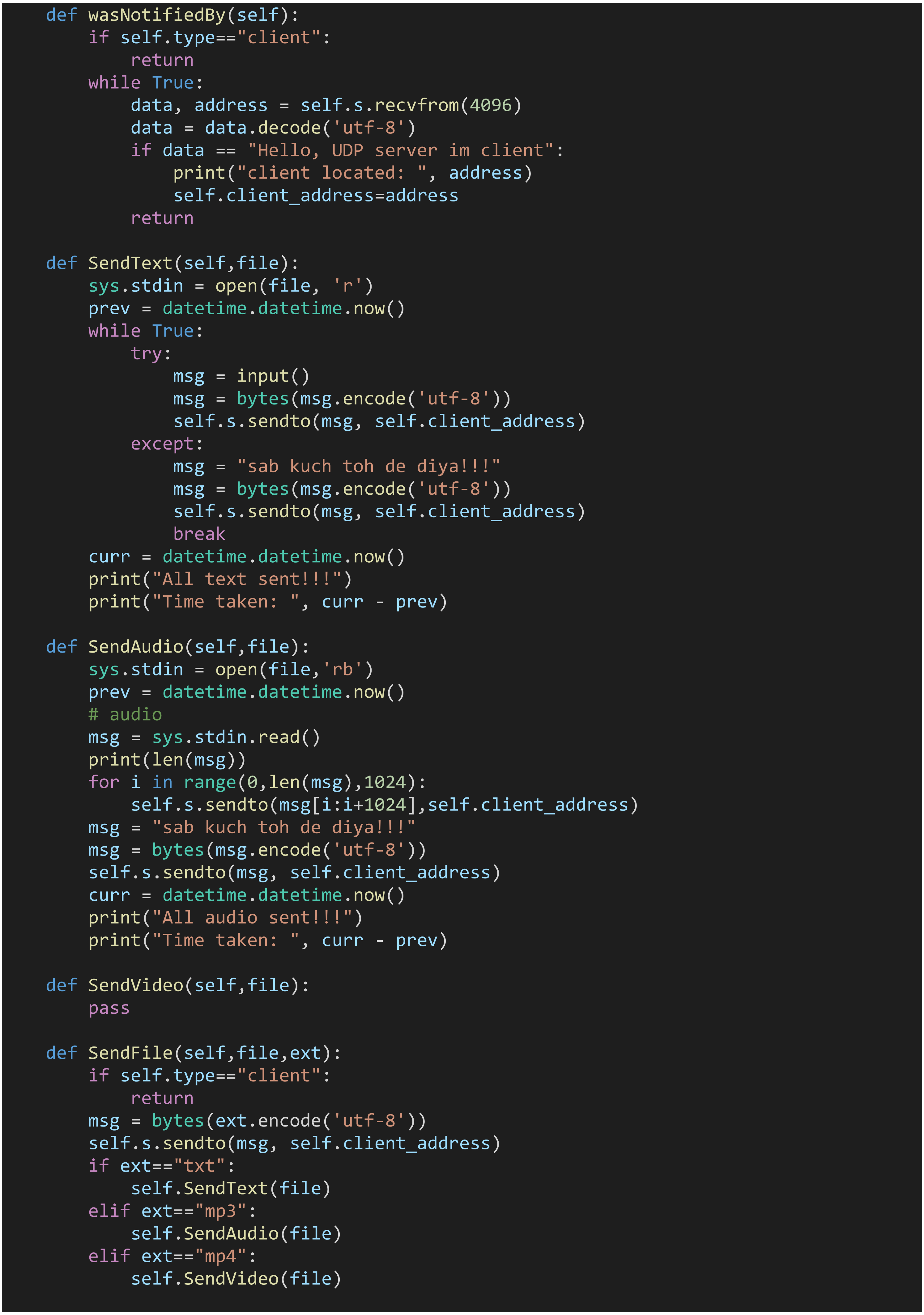
```
        pass

    def RecvFile(self):
        ext, address = self.s.recvfrom(4096)
        ext = ext.decode('utf-8')
        if address!=UDPFileTransfer.server_address:
            return
        if ext == "txt":
            self.RecvText()
        elif ext == "mp3":
            self.RecvAudio()
        elif ext == "mp4":
            self.RecvVideo()


server=UDPFileTransfer("server")
server.wasNotifiedBy()
while True:
    sys.stdin=sys.__stdin__
    c=input("""Send file
    1. text
    2. audio
    3. video
    0. quit
    """)
    if c=="0":
        msg = bytes('q'.encode('utf-8'))
        server.s.sendto(msg, server.client_address)
        break
    elif c=='1':
        server.SendFile("sending/file_send.txt","txt")
    elif c=='2':
        server.SendFile("sending/music_send.mp3","mp3")
    elif c=='3':
        server.SendFile("sending/video_send.mp4","mp4")
server.s.close()
```

Client.py



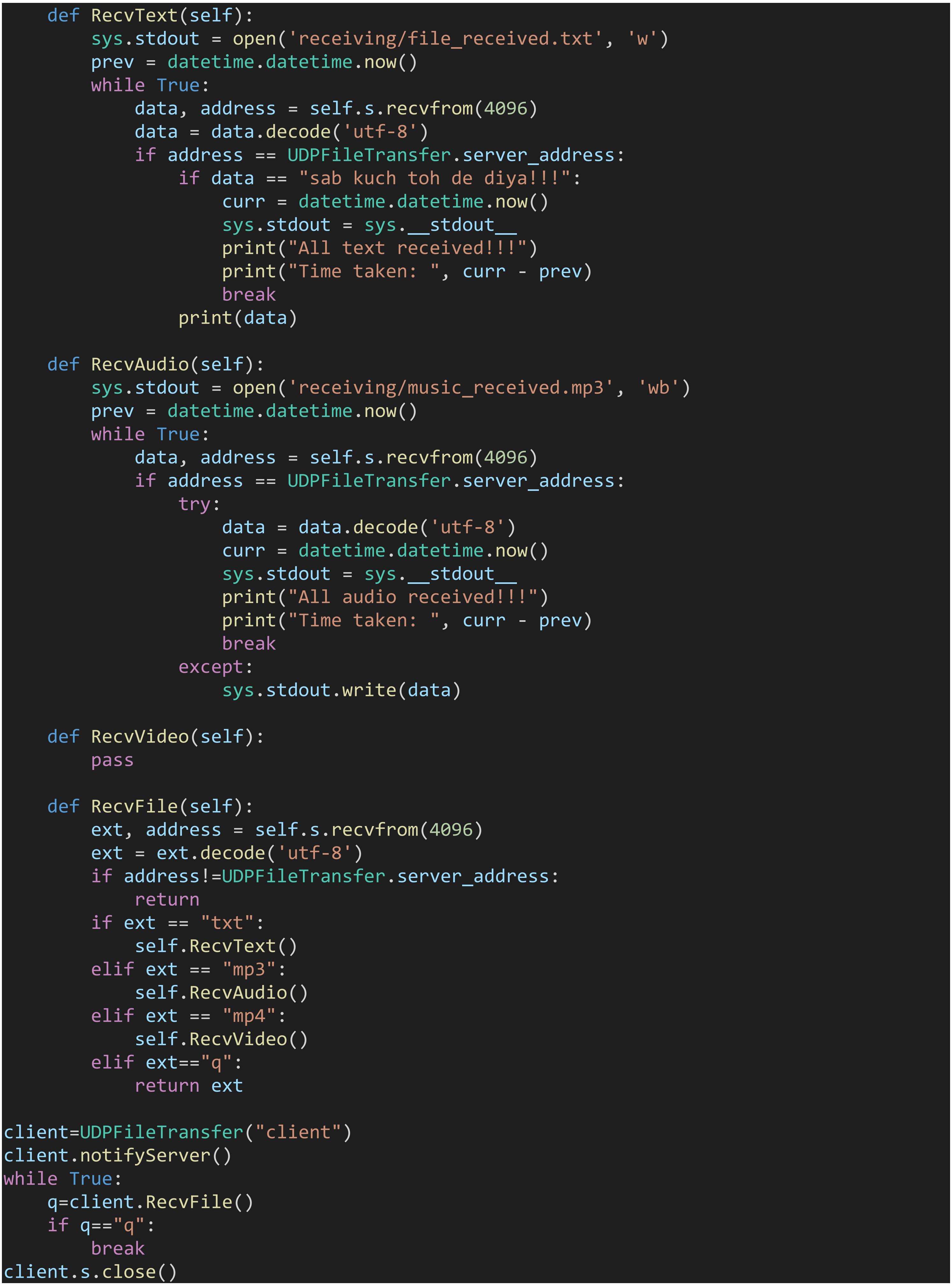
```
import socket,sys,datetime

class UDPFileTransfer:
    server_address = ("127.0.0.1", 12345)
    def __init__(self,type):
        self.type=type
        self.s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
        if type=="server":
            self.s.bind(UDPFileTransfer.server_address)
            print("UDP server running: ", UDPFileTransfer.server_address)
            self.client_address=None

    def notifyServer(self):
        if self.type=="server":
            return
        msg="Hello, UDP server im client"
        msg=bytes(msg.encode('utf-8'))
        self.s.sendto(msg,UDPFileTransfer.server_address)
        print("server was notified!!!")
        print("awaiting data...")
```

Scanned by TapScanner



```
    def wasNotifiedBy(self):
        if self.type=="client":
            return
        while True:
            data, address = self.s.recvfrom(4096)
            data = data.decode('utf-8')
            if data == "Hello, UDP server im client":
                print("client located: ", address)
                self.client_address=address
            return

    def SendText(self,file):
        sys.stdin = open(file, 'r')
        prev = datetime.datetime.now()
        while True:
            try:
                msg = input()
                msg = bytes(msg.encode('utf-8'))
                self.s.sendto(msg, self.client_address)
            except:
                msg = "sab kuch toh de diya!!!"
                msg = bytes(msg.encode('utf-8'))
                self.s.sendto(msg, self.client_address)
                break
        curr = datetime.datetime.now()
        print("All text sent!!!")
        print("Time taken: ", curr - prev)

    def SendAudio(self,file):
        sys.stdin = open(file,'rb')
        prev = datetime.datetime.now()
        # audio
        msg = sys.stdin.read()
        print(len(msg))
        for i in range(0,len(msg),1024):
            self.s.sendto(msg[i:i+1024],self.client_address)
        msg = "sab kuch toh de diya!!!"
        msg = bytes(msg.encode('utf-8'))
        self.s.sendto(msg, self.client_address)
        curr = datetime.datetime.now()
        print("All audio sent!!!")
        print("Time taken: ", curr - prev)

    def SendVideo(self,file):
        pass

    def SendFile(self,file,ext):
        if self.type=="client":
            return
        msg = bytes(ext.encode('utf-8'))
        self.s.sendto(msg, self.client_address)
        if ext=="txt":
            self.SendText(file)
        elif ext=="mp3":
            self.SendAudio(file)
        elif ext=="mp4":
            self.SendVideo(file)
```

Scanned by TapScanner



```python
    def RecvText(self):
        sys.stdout = open('receiving/file_received.txt', 'w')
        prev = datetime.datetime.now()
        while True:
            data, address = self.s.recvfrom(4096)
            data = data.decode('utf-8')
            if address == UDPFileTransfer.server_address:
                if data == "sab kuch toh de diya!!!":
                    curr = datetime.datetime.now()
                    sys.stdout = sys.__stdout__
                    print("All text received!!!")
                    print("Time taken: ", curr - prev)
                    break
                print(data)

    def RecvAudio(self):
        sys.stdout = open('receiving/music_received.mp3', 'wb')
        prev = datetime.datetime.now()
        while True:
            data, address = self.s.recvfrom(4096)
            if address == UDPFileTransfer.server_address:
                try:
                    data = data.decode('utf-8')
                    curr = datetime.datetime.now()
                    sys.stdout = sys.__stdout__
                    print("All audio received!!!")
                    print("Time taken: ", curr - prev)
                    break
                except:
                    sys.stdout.write(data)

    def RecvVideo(self):
        pass

    def RecvFile(self):
        ext, address = self.s.recvfrom(4096)
        ext = ext.decode('utf-8')
        if address!=UDPFileTransfer.server_address:
            return
        if ext == "txt":
            self.RecvText()
        elif ext == "mp3":
            self.RecvAudio()
        elif ext == "mp4":
            self.RecvVideo()
        elif ext=="q":
            return ext

client=UDPFileTransfer("client")
client.notifyServer()
while True:
    q=client.RecvFile()
    if q=="q":
        break
client.s.close()
```

OUTPUT :-

Scanned by TapScanner

Server

```
UDP server running:  ('127.0.0.1', 12345)
client located:  ('127.0.0.1', 55167)
Send file
    1. text
    2. audio
    3. video
    0. quit
1
All text sent!!!
Time taken:  0:00:00.001994
Send file
    1. text
    2. audio
    3. video
    0. quit
    2
2977540
All audio sent!!!
Time taken:  0:00:00.028003
Send file
    1. text
    2. audio
    3. video
    0. quit
    3
Send file
    1. text
    2. audio
    3. video
    0. quit
    0
```

Client

```
server was notified!!!
awaiting data...
All text received!!!
Time taken:  0:00:00.000995
All audio received!!!
Time taken:  0:00:00.027004
```

Scanned by TapScanner