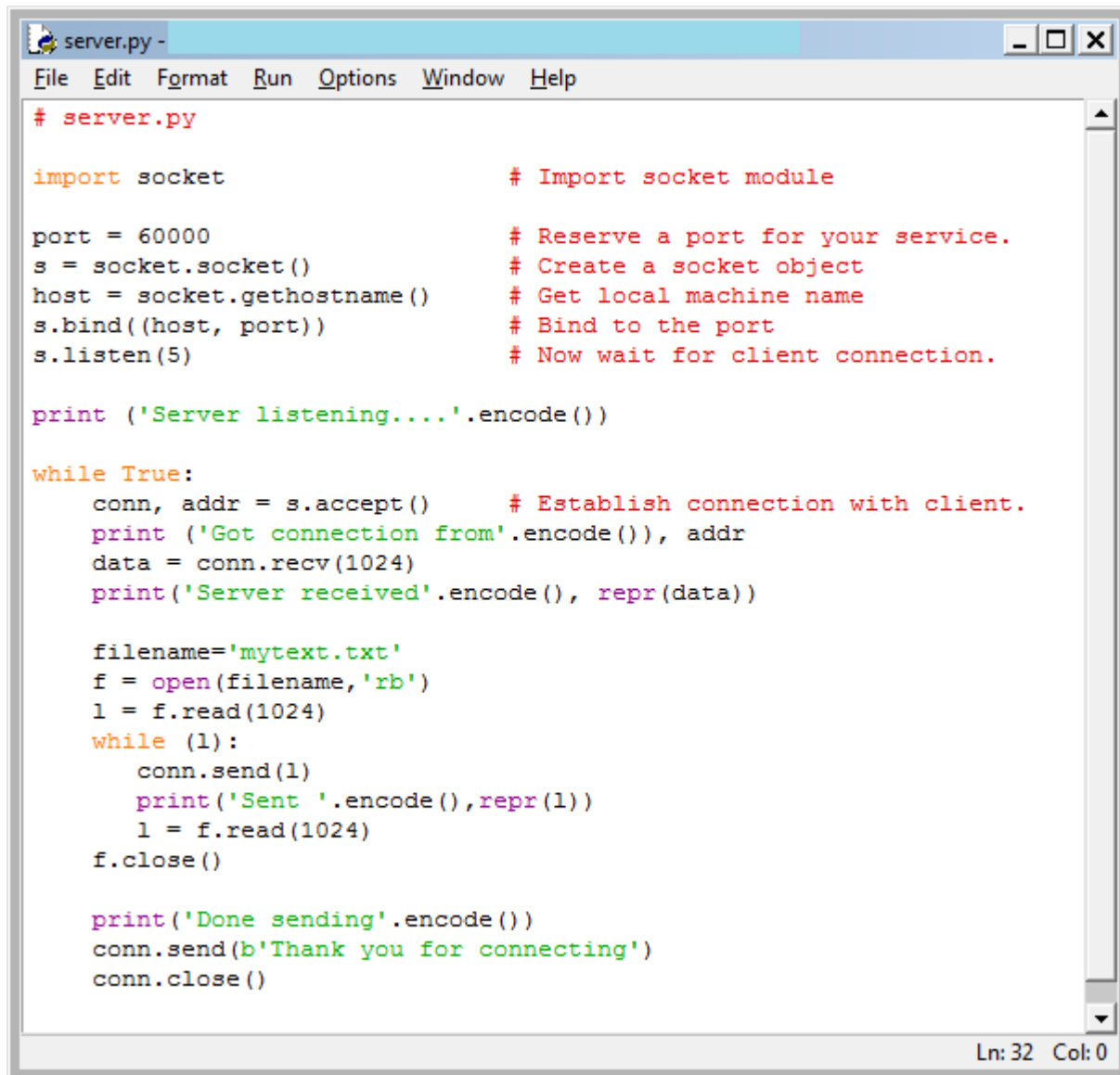


Local file transfer

Here is the code to send a file from a local server to a local client.

1. Server



```
# server.py

import socket                                # Import socket module

port = 60000                                # Reserve a port for your service.
s = socket.socket()                          # Create a socket object
host = socket.gethostname()                 # Get local machine name
s.bind((host, port))                        # Bind to the port
s.listen(5)                                  # Now wait for client connection.

print ('Server listening....'.encode())

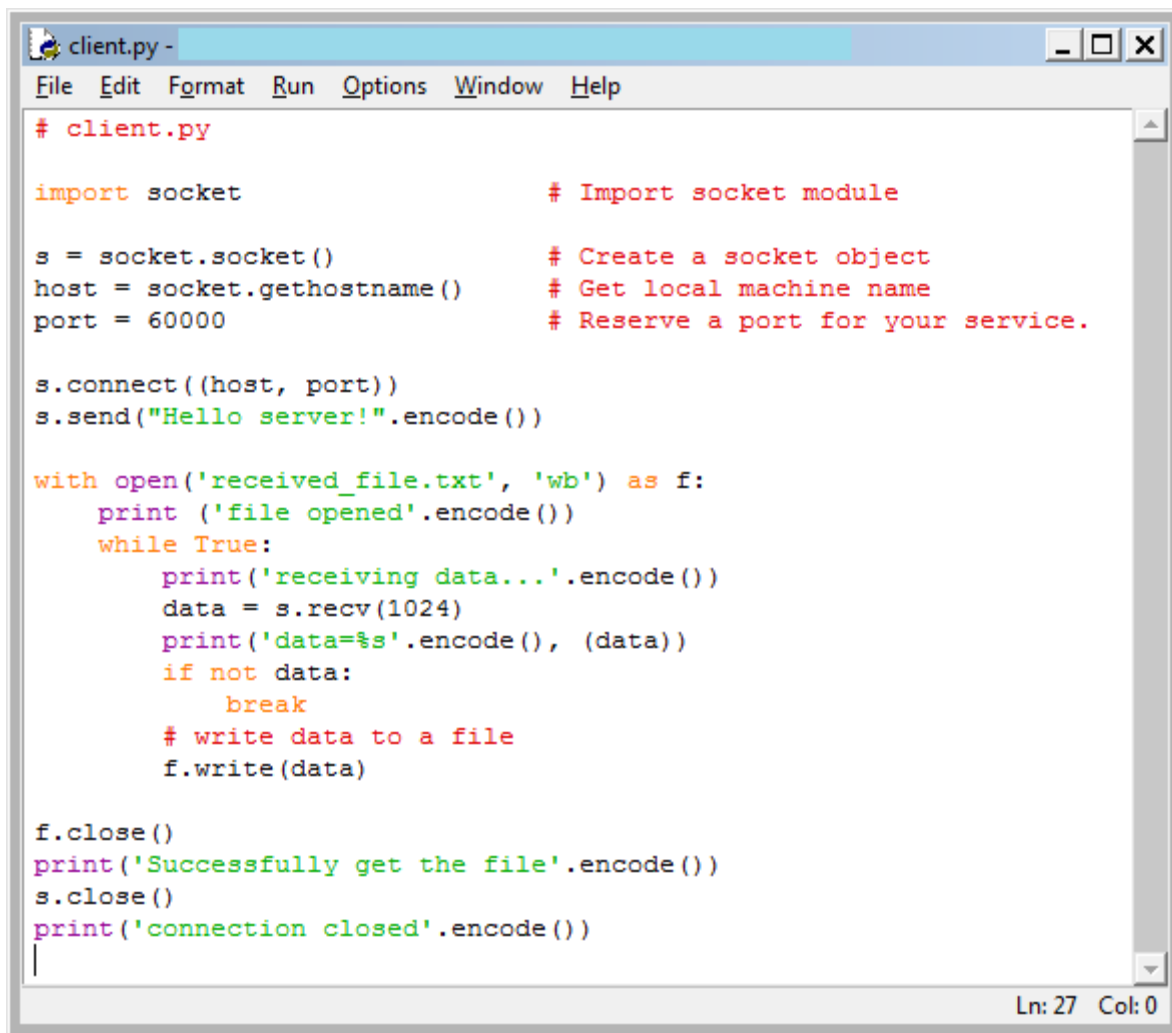
while True:
    conn, addr = s.accept()                 # Establish connection with client.
    print ('Got connection from'.encode()), addr
    data = conn.recv(1024)
    print('Server received'.encode(), repr(data))

    filename='mytext.txt'
    f = open(filename,'rb')
    l = f.read(1024)
    while (l):
        conn.send(l)
        print('Sent '.encode(),repr(l))
        l = f.read(1024)
    f.close()

    print('Done sending'.encode())
    conn.send(b'Thank you for connecting')
    conn.close()
```

Ln: 32 Col: 0

2. Client

A screenshot of a Python IDE window titled 'client.py'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The code is as follows:

```
# client.py

import socket                                # Import socket module

s = socket.socket()                          # Create a socket object
host = socket.gethostname()                  # Get local machine name
port = 60000                                # Reserve a port for your service.

s.connect((host, port))
s.send("Hello server!".encode())

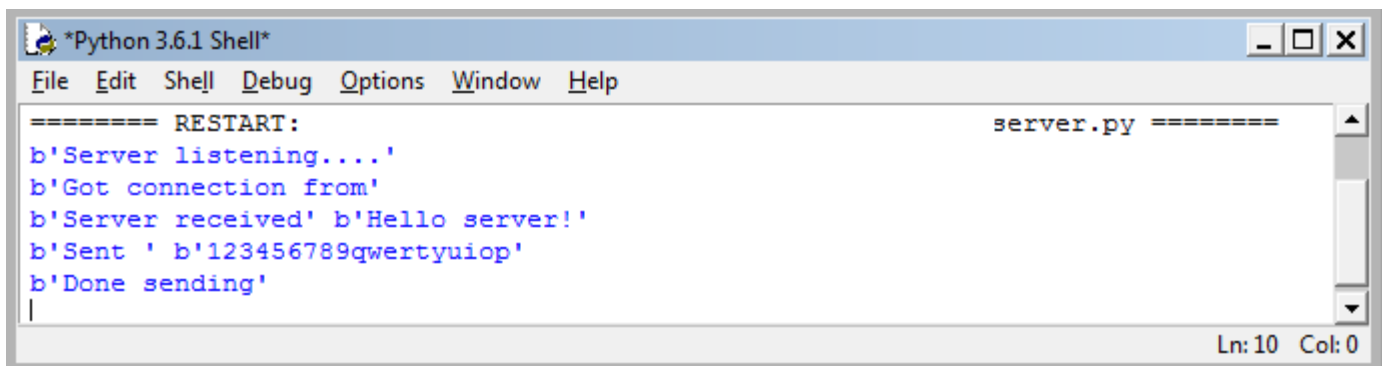
with open('received_file.txt', 'wb') as f:
    print('file opened'.encode())
    while True:
        print('receiving data...'.encode())
        data = s.recv(1024)
        print('data=%s'.encode(), (data))
        if not data:
            break
        # write data to a file
        f.write(data)

f.close()
print('Successfully get the file'.encode())
s.close()
print('connection closed'.encode())
```

The status bar at the bottom right shows 'Ln: 27 Col: 0'.

3. Results:

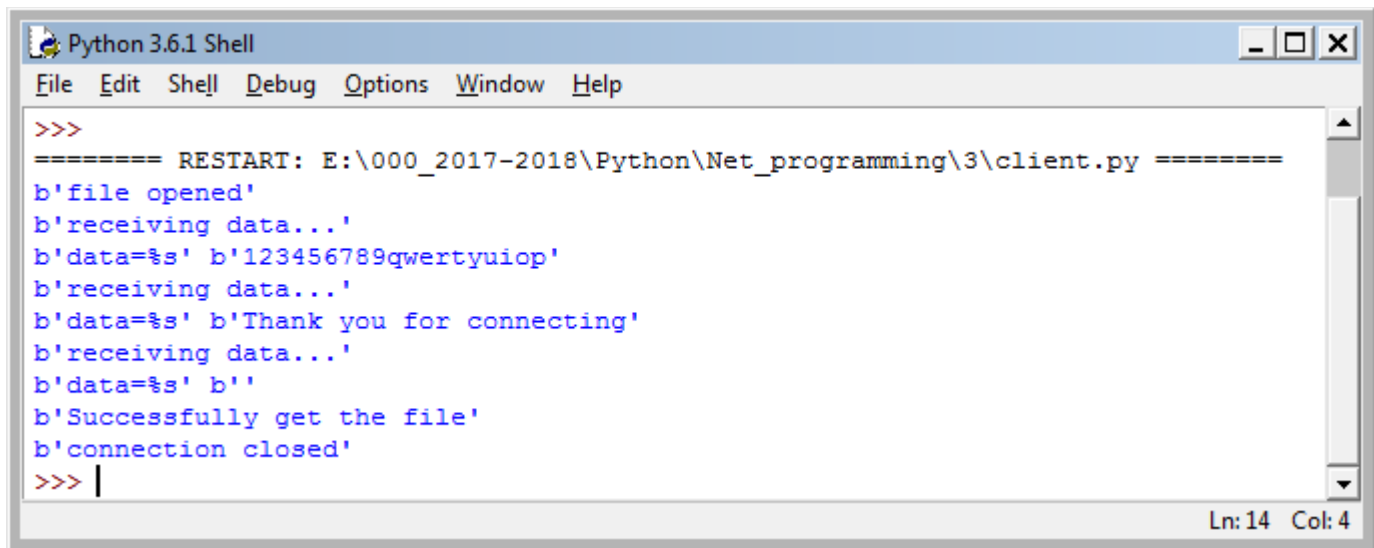
3.1. Output on a local server:

A screenshot of a Python 3.6.1 Shell window titled '*Python 3.6.1 Shell*'. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The output is as follows:

```
===== RESTART:                                server.py =====
b'Server listening....'
b'Got connection from'
b'Server received' b'Hello server!'
b'Sent ' b'123456789qwertyuiop'
b'Done sending'
```

The status bar at the bottom right shows 'Ln: 10 Col: 0'.

3.2. Output on a local client:



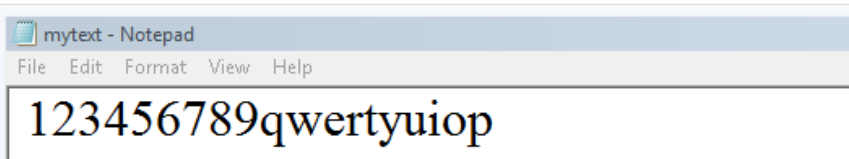
```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help

>>>
===== RESTART: E:\000_2017-2018\Python\Net_programming\3\client.py =====
b'file opened'
b'receiving data...'
b'data=%s' b'123456789qwertyuiop'
b'receiving data...'
b'data=%s' b'Thank you for connecting'
b'receiving data...'
b'data=%s' b''
b'Successfully get the file'
b'connection closed'
>>> |
```


Ln: 14 Col: 4

3.3. File Transfer

Name ^	Date modified	Type	Size
client			
mytext			
received_file			
server			



mytext - Notepad
File Edit Format View Help
123456789qwertyuiop



received_file - Notepad
File Edit Format View Help
123456789qwertyuiopThank you for connecting

Obs: Studentii sunt invitati sa peronalizeze solutia prezentata mai sus si sa includa in raportul laboratorului o prezentare similara sectiunii "Results" de mai sus