

This file transfer project – done in Python – was done to as a way to begin developing my skills in Python to develop and broaden my knowledge in both networking and programming.

As stated in the project name, it simply allows for file transfers from one client to another (in this case, it just copies the file and puts it in the same directory with whatever name I put for the copy since the recipient port is literally the same).

Anyways, to the point; this documentation aims to serve as a reflection for this project and understand exactly what it does, per each notable command and their purpose. And hopefully down the line, I can *eventually* develop the ability to critique the code and projects I encounter and come up with better ways to approach certain problems and/or projects.

*sender.py*

**import os** – part of the standard library of Python and provides a portable way of using operating system dependant functionality, such as returning the filesize (`os.path.getsize`) of the file to be sent.

**import socket** -- provides a way to work with sockets within our program. These sockets provide the ability to create network connections through a myriad of network protocols, including, but not limited to, TCP and UDP to send and receive data.

receiver.py

using `bytearray()` instead of `b""` on line 17 instantly sped up the process; `b""` tends to go all the way back to the previous chunk to “copy” it onto the new chunk. `Bytearray` simply appends the new chunk onto the already existing chunk, speeding up the process

The 'tqdm' package was used to visualise progress via a bar for receiver.py to let the user (or receiver) see, well, progress. It does so via percentage and file size.

