

Running The UDP Server & Client Together

We've spent the last few lessons writing code for a very basic client and a server. Let's see these in action in this lesson!

We'll cover the following



- Connecting the Two

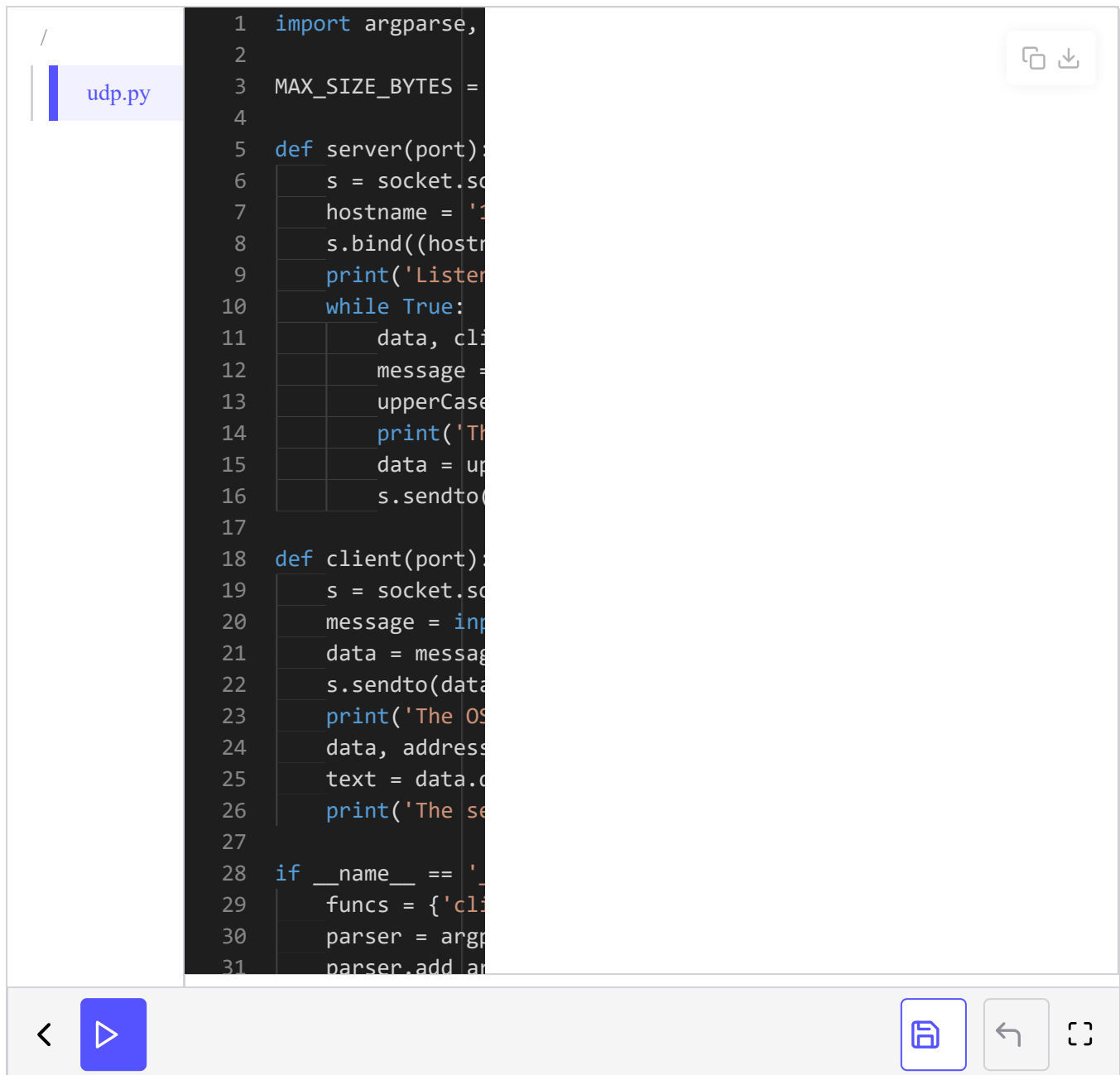
Connecting the Two

We'll run both together in one file called `udp.py` instead of running them separately. We've written some python code in the main function that allows you to specify which function you want the code to run, the `server` or the `client`.

To run the code, you would need to follow these steps:

1. Type your code and when you are ready to run the program, click on **Run**. The server code should start up automatically.
2. Open another terminal by clicking on +
3. Type the command `python3 /usercode/udp.py client` Note that it can be `server` in place of `client`.
4. Enter the text in the client window and see the effect.
5. If the program is not running to your satisfaction:
 1. Kill the running server program by typing the break sequence `ctrl+c` or `command+c` in both of the terminal windows.
 2. Change the code
 3. Click on **Run**.
 4. Type command `python3 /usercode/udp.py server` and `python3 /usercode/udp.py client` in the first and second terminal window, respectively. Go back to step 4.

Every time you make a change to the code you must click **run** for the changes to take effect.



The screenshot shows a code editor with a file named `udp.py`. The code is a Python program that implements a simple UDP server and client. The server function (`server`) listens on a specified port, receives data from a client, and prints it. The client function (`client`) sends a message to the server and prints the received data. The main function (`if __name__ == '__main__':`) sets up the command-line arguments and calls the server or client function based on the command-line options.

```
1 import argparse,
2
3 MAX_SIZE_BYTES =
4
5 def server(port):
6     s = socket.so
7     hostname = '1
8     s.bind((hostn
9     print('Liste
10    while True:
11        data, cli
12        message =
13        upperCase
14        print('Th
15        data = up
16        s.sendto(
17
18 def client(port):
19     s = socket.so
20     message = inp
21     data = messag
22     s.sendto(data
23     print('The OS
24     data, address
25     text = data.d
26     print('The se
27
28 if __name__ == '
29     funcs = {'cli
30     parser = argp
31     parser.add ar
```

The editor interface includes a file explorer on the left showing the file `udp.py`. The code is displayed in a dark-themed editor with line numbers. A toolbar at the bottom contains icons for navigation (back, forward), saving, undo, and a full-screen view.

In the next lesson, we're going to look at some possible improvements to our current UDP program.