



# Basic Knowledge

- Python provides two levels of access to network services.
- At a low level, you can access the basic socket support in the underlying operating system, which allows you to implement clients and servers for both connection-oriented and connectionless protocols.
- Python also has libraries that provide higher-level access to specific application-level network protocols, such as FTP, HTTP, and so on.



# server-tcp.py

```
import socket

s = socket.socket()
print("Socket successfully created")

port = 8888

s.bind(('', port))
print("socket binded to " + str(port))

s.listen(5)
print("socket is listening")

while True:
    c, addr = s.accept()
    print("Got connection from" + str(addr))

    c.send(b'Thank you for connecting')
c.close()
```

# client-tcp.py

```
import socket

s = socket.socket()

port = 8888

s.connect(('192.168.114.6', port))

data = s.recv(1024)

print (data)

s.close()
```

# Exercises

By using Python programming please create:

1. UDP time server and client that connect through port 22000.
2. A program using port 27679 TCP that passing multiple random numbers.
3. A UDP client-server port 11235 that passing Fibonacci series based on input from client where follow the following criteria:
  - $x_n$  is term number "n"
  - The client must provide n.
  - The server must send term and onwards of five sequence.
  - The client can choose either want another five sequence or quit the program.
  - The server must continue send another five sequence onward of last term provided if ask to do so by the client.

# References

- Google
- Youtube
- eBook