

Pythonprogramm das wie Server funktioniert: python udp_server.py

```
import socket
SERVER_IP = "0.0.0.0"
SERVER_PORT = 5005

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((SERVER_IP, SERVER_PORT))

print("UDP Server läuft...")

while True:
    data, addr = sock.recvfrom(1024)
    print(f"Empfangen von {addr}: {data.decode()}")

    if data.decode() == "ping":
        sock.sendto("pong".encode(), addr)
        print("→ pong gesendet")
```

Funktionalität, die Python braucht, um über das Netzwerk zu sprechen.

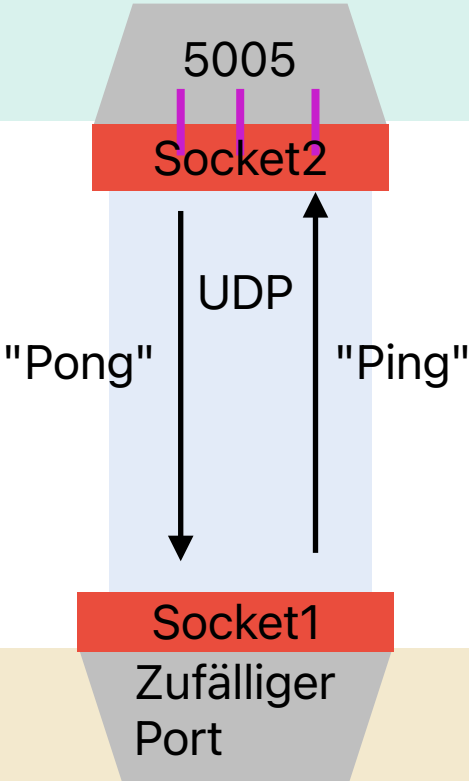
Anfrage kann von überall kommen

Port auf dem der Server lauscht

Erstellt ein neues Socket

Bindet das Socket an die angegebene IP und Port. (Nötig t das Socket an die angegebene IP und Port. (nötig weil fester prot gewählt wird, damit client weiss wohi packet schicken)

Wartet (blockiert) auf ein Nachricht vom Netzwerk:



```
import socket

SERVER_IP = "127.0.0.1"
SERVER_PORT = 5005

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

sock.sendto("ping".encode(), (SERVER_IP, SERVER_PORT))
print("→ ping gesendet")

data, addr = sock.recvfrom(1024)
print("Antwort:", data.decode())
```

(localhost).

An diesen Port wird gesendet

Erstellt ein neues Socket

Schickt die Nachricht:

Wartet (blockiert) auf eine Antwort vom Netzwerk:

Wandelt die empfangenen Bytes zurück in Text (decode()) und zeigt die Antwort an

Pythonprogramm das wie Client funktioniert:python udp_client.py

>_Terminal

```
python udp_server.py

python udp_client.py
```