

Sockets

Blendor Uruci

Inhaltsverzeichnis

- / Was sind Sockets
- / Funktionsweise von Sockets
- / Aufsetzen der Modelle

Socket

- / Ermöglicht die Netzwerkkommunikation zwischen verschiedenen Computern.
- / Stellt die grundlegenden Funktionen zur Verfügung, um sowohl serverseitige als auch clientseitige Anwendungen zu erstellen.
- / Ein Socket ist ein Endpunkt einer bidirektionalen Kommunikationsverbindung zwischen zwei Programmen, die über ein Netzwerk ausgeführt werden.
- / Sockets können sowohl über TCP (Transmission Control Protocol) als auch über UDP (User Datagram Protocol) kommunizieren.

Grundlegende Begriffe

- / **Host:** Computer im Netzwerk
- / **Port:** Eine nummerierte Tür auf einem Host
- / **Server:** Ein Programm, das auf die Verbindung von Clients wartet
- / **Client:** Ein Programm, das eine Verbindung zu einem Server herstellt.

Einzelauftrag

- / Ihr findet die beiden Python files «server_tcp.py» und «client_tcp.py» im Microsoft Teams
- / Erstelle einen einfachen Chat-Server, der Nachrichten von mehreren Clients empfangen und an alle verbundenen Clients senden kann.

Modelle

- / Ihr findet die Modelle im Microsoft Teams
- / Bitte ladet diese herunter und speichert sie ab
- / Startet eines der Modelle

Pythonanbindung

/ Neues Pythonfile erstellen und erste Zeilen ausfüllen

```
import socket # Import von Scket modul
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # Internet, UDP
sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1624) # reuse already used port address 1600

# deblock if former programm run got crashed
sock.setblocking(0) # setting to non-blocking socket mode-> socket.SOCK_NONBLOCK not supported by
Windows
sock.bind(('localhost', 1624)) # Socket stays open + keeps used Portnummer... if not properly closed at
the end
# double parenthesis since 1 parameter(== union) expected
```

Pythonanbindung

/ Verbindungsaufbau + schreiben zum Modell:

```
while True: # Schlaufe für das Programm im Runing-Betrieb
    lichtschalter= 0
    ##### S11 Port 1611
    try:
        msg, addr= sock.recvfrom(1024)
        print("Got message from %s: %s" % (addr, msg))
        lichtschalter= int(msg)
    except socket.errorase: # supress BlockingIOError from non-blockingsocket access if noData received
        #print ("Error creating socket: %s" % e)
        pass
    print(lichtschalter)
    sock.sendto(str.encode(str(lichtschalter)), ('localhost', 1711))
```