TCP-programmering med Python del 2

Formål: At lave mere avancerede TCP programmering i Python

Baggrund:

- TCP opgave 1

Opgave 1: Refactor Serveren til at håndtere flere klienter samtidigt

Din server kan behandle en klient ad gangen og først derefter behandle den næste.
Du skal nu lave din server så den kan håndtere flere klienter samtidigt.

Step 1:

Flyt al kode, der har en én klient at gøre, over i en metode (kald den evt. handleClient).
den skal have to parametre en socketConnection og addr (de to variable der bliver initialiseret ved 'accept'-kaldet).
Hint: det er al kode efter accept-kaldet der flyttes til denne metode.

Din kode skulle gerne lige noget a la dette:

```
while True:
    connectionSocket, addr = serverSocket.accept()
    handleClient(connectionSocket, addr)
```

og din metode noget a la dette:
:

```
def handleClient(connectionSocket, address):
    sentence = connectionSocket.recv(1024).decode()
    print(sentence)
    capitalizedSentence = sentence.upper()
    connectionSocket.send(capitalizedSentence.encode())
    connectionSocket.close()
```

Din Refactoring er nu klaret :)

Prøv om dit program stadig virker.
Start din server og start to udgaver af SocketTest og send en 'message' fra den senest åbne SocketTest – hvad sker der?
Så send en fra den først åbne – hvad sker der nu?

Step 2:

Du skal nu I din server understøtte at der kan køre flere udgaver samtidigt – det er hvad vi i programering kalder Threading (tråde)

I python gøres dette ved:

1. import threading # gøre brug af threading modulet
2. Du starter en thread ved:

   threading.Thread(target = <function>,args = (<args>)).start()

   Udskift <function> med din metode og <args> med *connectionSocket, addr*
3. Implementer dette i din server

Prøv den same øvelse som før med at åbne to SocketTest og se hvad der skar nu.

Opgave 2: Prøv at sende til sidepersonen

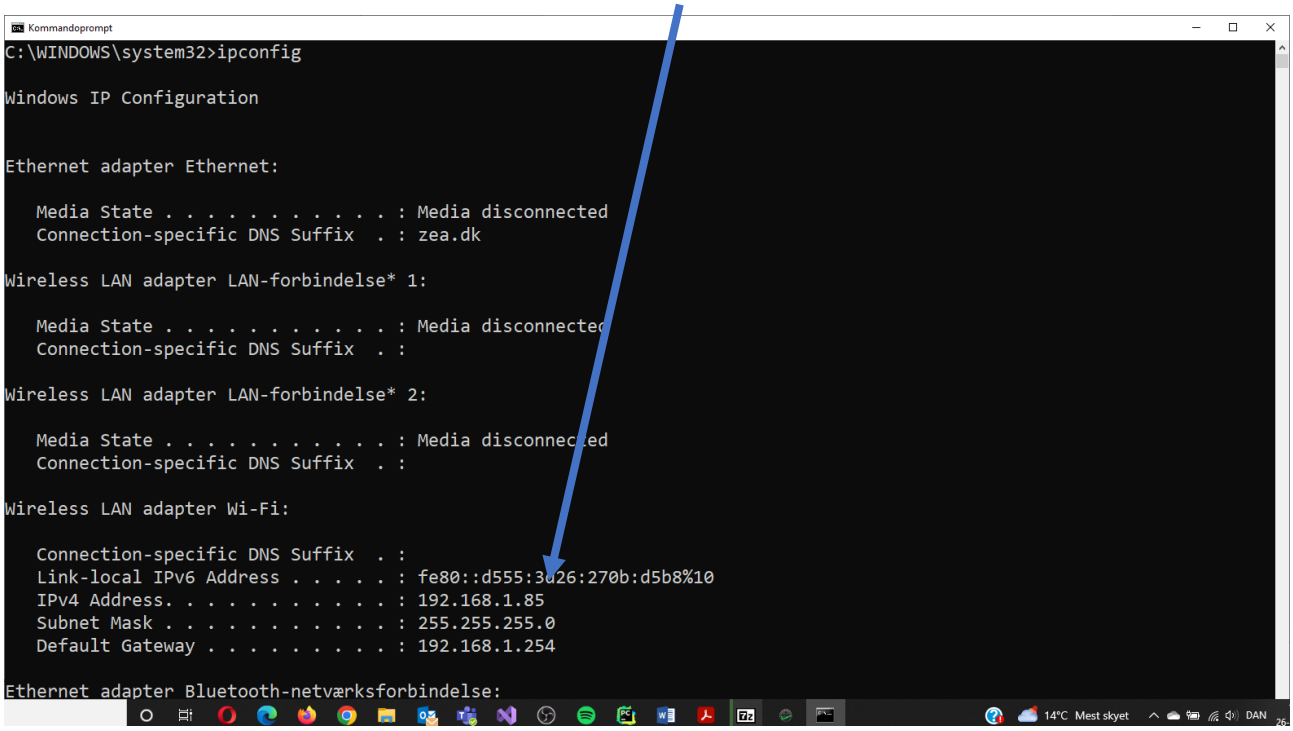
Du skal skifte over til **DMU – nettet** (dmu1,2,3 eller 4).

Sørg for at I begge skifter til det samme net.

Find din IP-adresse

1. Åben kommandoprompten (commandprompt)

2. Skriv ipconfig, hvorefter du kan aflæse din IP-adresse



3. Klienten skal nu benytte denne adresse i Connect-kaldet clientSocket.connect('<<IP-adresse>>', serverPort))

Ekstra A: Aftal en protokol med din sideperson

Du skal sammen med din sideperson aftale hvad en client skal sende til serverne og hvad serveren så skal svare.

f.eks:

Client ’peter’ -> Server ’PETER’ - til store bodstaver

Client ’peter’ -> server ’retep’ - baglæns

(hint https://www.w3schools.com/python/python_howto_reverse_string.asp)

Client ‘Add 4 5’ -> Server 9 - sum

(hint: du skal splitte teksten og derefter finde de to tal
https://www.w3schools.com/python/ref_string_split.asp)

Den ene koder Serveren – den anden koder klienten