

# Lab 5

## [Obligatorisk] Socket: Sten Sax Påse

Syftet med denna laboration är att öva på s.k. socketprogrammering med python.

Som läsanvisning ges dels föreläsningen (den om TCP/IP:s transportlager) och dels referenserna

*[docs.python.org/3.6/library/socket.html](https://docs.python.org/3.6/library/socket.html) och*

*[docs.python.org/3.6/howto/sockets.html#socket-howto](https://docs.python.org/3.6/howto/sockets.html#socket-howto)*

### 5.1 Sten Sax Påse

Sten sax påse är ett klassiskt spel! Eftersom ett 'O' ser ut som en sten, 'X' som en sax, och 'U' som en påse, är det lämpligt att låta dessa tre tecken representera sten, sax och påse i denna datoriserade variant av spelet! Spelets regler är enkla. Två spelare. Båda knappar in var sitt tecken (O,X,U) , därefter får de reda på motspelarens tecken, och sedan får eventuellt en av spelarna poäng:

1. Om sten och sax möts vinner stenen (saxen går sönder då den klipper i sten)
2. Om sax och påse möts vinner saxen (saxen klipper sönder påsen)
3. Om sten och påse möts vinner påsen (påsen fångar stenen)

Så här skall spelet gå till:

1. Båda spelarna använder samma program och startar det på varsin dator.

### 5.1. STEN SAX PÅSE5. [OBLIGATORISK] SOCKET: STEN SAX PÅSE

2. Programmet frågar om man skall vara klient (K) eller server (S). En av spelarna måste svara S och den andra K. Den spelare som är S måste svara först!
  - (a) Om spelaren svarar S skall programmet skapa en server som lyssnar på port 60003 på den aktuella datorn<sup>1</sup>
  - (b) Om spelaren svarar K skall programmet fråga efter serverns namn eller ip-adress, och därefter göra connect till port 60003 på denna ip-adress.
3. Programmet skall nu skriva ut spelställningen och sedan fråga efter spelarens drag. Så här (den aktuella spelarens poängantal först i parentesen)

(0,0) Ditt drag:
4. Programmet skall sedan läsa in spelarens drag. Verifiera att det är giltigt (dvs O, X, eller U), och begära ny inmatning annars.
5. Så snart spelaren knappat in ett giltigt speldrag skall programmet skicka detta till motspelaren över socketförbindelsen.
6. Därefter skall programmet läsa motspelarens drag över socketförbindelsen. Detta kan innebära att programmet låser sig under tiden som det väntar på motspelarens drag, men det kan också hända att motspelaren redan har skickat sitt drag till oss, och i så fall är det istället motspelaren som fått vänta lite.
7. Så fort vi läst motspelarens drag skall vi skriva ut det på skärmen, så här:

(motspelarens drag: X)
8. Sedan avgör vi vem som vann rondens och uppdaterar spelställningen. Om ingen spelare fått 10 poäng fortsätter spelet, dvs vi hoppar tillbaka till punkt 3 ovan.
9. Om någon har fått 10 poäng skall man få reda på om man vann eller förlorade och med hur mycket. Så här:

Du vann med 10 mot 7

---

<sup>1</sup>Om detta portnummer inte fungerar kan du prova en annan port. Tänk också på följande: Om man nyligen har kört en server på en viss port måste man normalt vänta några sekunder innan man kan starta en ny server som lyssnar på samma port.

### 5.1. STEN SAX PÅSE5. [OBLIGATORISK] SOCKET: STEN SAX PÅSE

10. Därefter skall spelet avslutas och socketförbindelsen stängas.

Nedan ser vi ett litet körexempel:

```
nora$ python3 stenSax.py
Vill du kora som klient eller server (K/S): K
Ange servers namn eller ip:127.0.0.1
(0,0) Ditt drag:X
(motspelarens drag U)
(1,0) Ditt drag:0
(motspelarens drag X)
(2,0) Ditt drag:X
(motspelarens drag U)

...

(8,9) Ditt drag:0
(motspelarens drag U)
Du forlorade med 8 mot 10
```

#### 5.1.1 Implementeringstips

Här följer några tips som man får ignorera om man vill

1. Använd python version 3.x (inte 2.x)!
2. Provkör i det interaktiva fönstret innan du knappar in kod i programmet!
3. Svenska tecken kan ge problem. Undvik dem!
4. I python 3 kan man läsa data från tangentbordet med `input("...")`.  
Exempel:

```
ans = input("Vill du kora som klient eller server (K/S): ")
```

5. De första raderna i mitt eget huvudprogram ser ungefär ut så här

```
def serversideGetPlaySocket():
    bla bla bla....

def clientsideGetPlaySocket(host):
    bla bla..

ans = "?"
while ans not in {"K", "S"}:
    ans = input("Vill du kora som klient eller server (K/S): ")
```

### 5.1. STEN SAX PÅSE5. [OBLIGATORISK] SOCKET: STEN SAX PÅSE

```
if ans=="S":
    sock = serversideGetPlaySocket()
else:
    host = input("Ange servers namn eller ip:")
    sock = clientsideGetPlaySocket(host)
```

de kodrader som följer är desamma för klient och server!

6. En sträng bör göras om till en *byte*-array innan den kan skickas över en socketförbindelse. Nedanstående kod visar hur det kan gå till

```
barr = bytearray(str, "ASCII")
```

7. Har man tagit emot en *byte*-array och vill göra om den till sträng skriver man så här

```
str = barr.decode("ASCII")
```

#### 5.1.2 Redovisning

Ladda upp pythonfilen till pingpong (och redovisa också muntligt).