

## IN1000 - Opplegg seminartimer uke 6, Løsningsforslag

### Oppgave 1

Skriv en funksjon som tar imot to tall og returnerer det største av de to tallene (ikke skrive det ut).

#### Løsningsforslag

##### Alternativ 1:

```
def storst_av_to(tall1, tall2):  
    if tall1 > tall2:  
        return tall1  
    else:  
        return tall2
```

##### Alternativ 2

```
def storst_av_to(tall1, tall2):  
    if tall1 > tall2:  
        return tall1  
    return tall2
```

Alternativ 2 gir same resultat, fordi når programmet kommer til *return*, så går det ut av funksjonen. Hvis programmet er ferdig med if-testen i alternativ 2, uten å ha kommet til *return*, betyr det at if-testen ikke var sann.

### Oppgave 2

Anta at du har filen "historie.txt". Les inn alle linjene og lagre dem i en liste. Første linje skal ligge først i listen, osv. (Tips: husk å åpne og lukke filen..)

#### Løsningsforslag 1

```
innfil = open("historie.txt", "r")  
historie_liste = []  
linje = innfil.readline()  
  
while linje != "" :  
    historie_liste.append(linje)  
    linje = innfil.readline()  
  
innfil.close()
```

#### Løsningsforslag 2

```
innfil = open("historie.txt", "r")  
historieListe = []  
  
for linje in innfil:  
    historieListe.append(linje)  
  
innfil.close()
```

"r" er valgfritt, men det kan være fint å ha det med. Hvis man ikke skriver noe i tillegg til navna på fila, så forstår Python det som at det er "r" vi mener.

### Oppgave 3

Skriv en funksjon som tar i mot to sannhetsverdier (True/False) og returnerer 1 hvis begge verdier er sanne og 0 hvis begge verdier ikke er sanne.

Som vi snakket om i timen vil ikke denne formuleringen dekke tilfellene True/False False/True. En løsning er å bruke en annen verdi for å indikere de andre tilfellene. Da får vi noe slikt:

```
def is_true(bool1, bool2):
    if bool1 and bool2:
        return 1
    if not bool1 and not bool2:
        return 0
    return -1
```

I klassen snakket vi om tilfellet der vi vil ha 0 hvis ikke begge verdiene er sanne, da får vi:

```
def is_true(bool1, bool2):
    if bool1 and bool2:
        return 1
    else:
        return 0
```

Denne kan forenkles som i oppgave 1.

### Oppgave 4

Skriv et program som bruker for-løkker til å ...

- Skriv ut alle partall mellom 0 og 10.
- Finne summen til alle partall mellom 0 og 10.
- Finne summen av alle oddetall mellom 10 og 0.

Løsningsforslag:

```
#a)
for i in range(0, 10, 2) :
    print(i)

#b)
sum_partall = 0
for i in range(0, 10, 2) :
    sum_partall += i

print("sum av alle partall: ", sum_partall)

#c)
sum_oddetall = 0
for i in range(1, 10, 2) :
    sum_oddetall += i

print("sum av alle oddetall: ", sum_oddetall)
```

Dette er altså tallene fra og med 0, til, men ikke inkludert 10. Dette er det samme som `range(10)`, `range(0,10)`, og med matematisk notasjon  $[0,10)$ . Det vi gjorde i timen var  $[0,10]$ , altså `range(0,11)`, for å få med tallet 10 i beregningene.

Merk at her har `range()` tre parametre. Hvis vi bruker færre parametre er det alltid underforstått hva de to andre er:

```
range(10)      #Underforstått: range(0,10,1)
                #Representerer mengden [0,1,2,3,4,5,6,7,8,9]

range(0,10)   #Underforstått: range(0,10,1)
                #Representerer mengden [0,1,2,3,4,5,6,7,8,9]
                #(samme som over)

range(0,10,2) #Her har vi alle tre.
                #Representerer mengden [0,2,4,6,8]
```

## Oppgave 5

Skriv et program som definerer en liste `[2, 3, 6, 8]`. Bruk en for-løkke til å skrive ut alle verdiene i listen.

*Utfordringer:*

a)

Skriv om prosedyren over/skriv ny prosedyre, som tar en liste som input/parameter og som skriver ut alle verdiene i listen.

b)

Endre prosedyren til en funksjon slik at den returnerer den laveste verdien i listen i stede for å skrive dem ut. Hint: her kan det være lurt å ha en variabel som mellomlagring av den minste verdien. Den største verdien i listen er 8.

```
talliste = [2, 3, 6, 8]

for elem in talliste :
    print(elem)

#a)
def skriv_alle(liste):
    for elem in liste :
        print(elem)

#Eksempelkjøring
talliste = [2, 3, 6, 8]
skriv_alle(talliste)

#b)
def finn_minst(liste) :
    minst = liste[0]
    for elem in liste:
        if elem < minst:
            minst = elem
    return minst
```

## Oppgave 6

Hva skrives ut i de to forskjellige løkkene?

```
for i in range(5):  
    j = 0  
    j = j+1  
    print("j = ", j)  
  
k = 0  
for i in range(5):  
    k = k+1  
    print("k = ", k)
```

**Løsningsforslag:**

**Løsning**

```
j = 1  
j = 1  
j = 1  
j = 1  
j = 1
```

```
k = 1  
k = 2  
k = 3  
k = 4  
k = 5
```

Grunnen til at det blir slik er fordi i den første løkka finnes j bare inni løkka, den startes på nytt hver gang løkka tar en runde. I den andre løkka derimot er k definert i det globale skopet, og derfor kan vi ta vare på endringene.