

SEMINAR 4 – IN1000

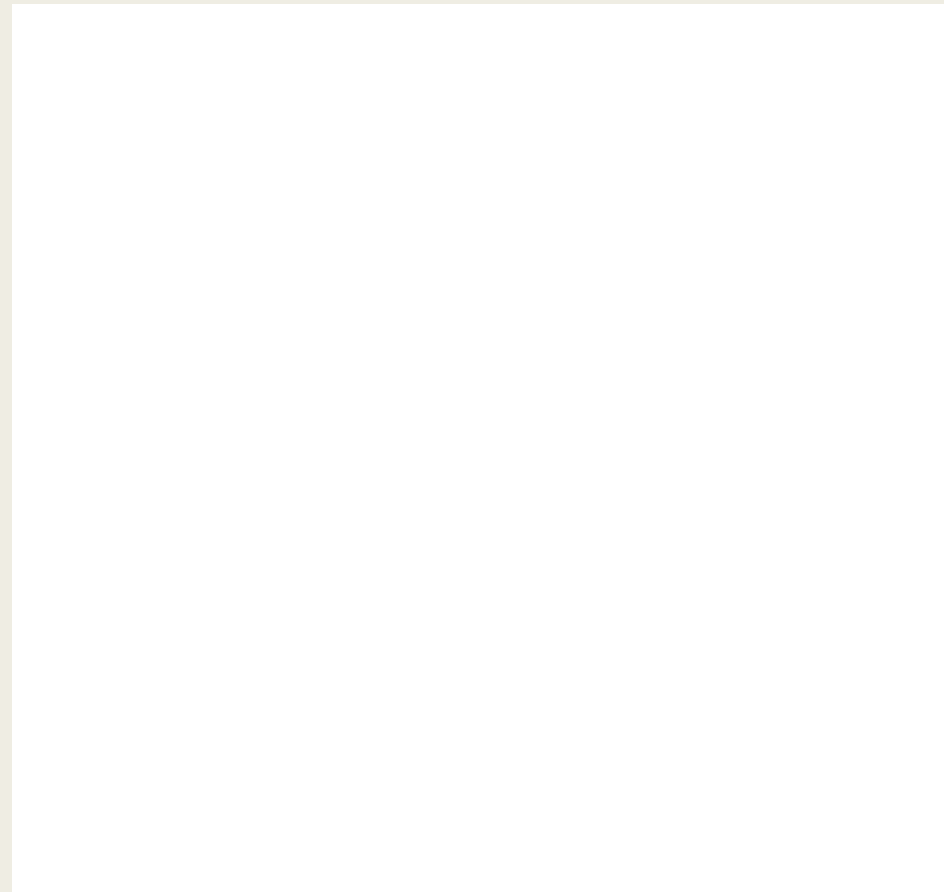
- While-løkker
- For-løkker med teller og på lister
 - Prosedyrer med parametre
 - Funksjoner

While-løkke:

- En løkke inneholder en kodeblokk
- Som gjentas helt til uttrykket evalueres om usant
- Det boolske uttrykket må være modifiserbart (kan evalueres til false på tidspunkt)

Oppgave 1

Hva skrives ut her:



Oppgave 2

Hva tror du skrives ut på skjermen av programmet under? Diskuter med gruppen



For-løkker

- `for i in range(tall)`
 - *Går fra 0 til tall.*
 - *F.eks: `for i in range(4)`: # vil gå fra 0 til 4 (NB! IKKE TIL OG MED) fra 0 tom 3*
- `for i in range(start, slutt)`
 - *I vil ha startsverdi start og slutte på slutt*
 - *F.eks. `for i in range(2, 5)`: # vil starte som 2 og gå til 5/tom. 4*
- `for i in range(start, slutt, økning)`
 - *I vil ha startsverdi start og slutte på slutt, for hver iterasjon (runde) vil i øke med økning*
 - *F.eks `for i in range(2, 15, 3)`: #i vil starte som 2 øke med 3 for hver runde helt til i er større enn 15 (eller lik 15), da slutter den*

for-loop VS. while-loop

- for-loop brukes ofte når man vet hvor mange ganger noe skal skje
 - While kan gjerne brukes når man ikke vet hvor mange ganger noe skal skje, men f.eks. hvis man vil lagre et ukjent antall verdier en bruker taster helt til brukeren taster 0.
-
- PS: ofte kan man likevel velge hva man ønsker

Oppgave 3

Skriv et program med en **prosedyre** printHei, prosedyren skal skrive ut teksten “Hei” til skjermen. Definer en variabel a med en verdi 5 og lag en while-løkke, løkken skal kalle prosedyren printHei. Bruk variabelen a slik at løkken er ferdig etter du har kalt print_hei 5 ganger.

Skriv **prosedyren** printHei. Deretter løs oppgaven både ved hjelp av while-løkke og for-løkke.

For-løkker for lister/ for each

Funksjon

- En funksjon som defineres med def. Funksjoner har ALLTID en return verdi!
- Når man returnerer noe avsluttes funksjonen (Videre linjer kjøres IKKE)



Parametere

- Til prosedyrer/funksjoner kan man også sende med parametre, som er en type variabler. Variablene man sender med “blir” de som er med som argument i prosedyren.



Funksjon VS Prosedyre Vs Metode

Funksjon	<p>En funksjon som defineres med <i>def</i>, som ikke er del av en klasse (ordet <i>self</i> brukes ikke). Funksjoner har alltid en returverdi.</p> <p>Eks:</p> <pre>def sum(a, b): c = a + b return c</pre> <p>På engelsk kalles dette <i>function</i>.</p>
Prosedyre	<p>Tilsvarende funksjon, men uten returverdi. Bruker aldri ordet <i>self</i>, og er ikke del av en klasse.</p> <p>Eks:</p> <pre>def superprint(ord): print("ordet er", ord)</pre> <p>På engelsk kalles dette <i>procedure</i>.</p>
Metode	<p>Tilsvarende funksjon, men som del av en klasse. Har alltid <i>self</i> som første parameter. Kan, men må ikke, ha en returverdi.</p> <p>Eks:</p> <pre>def areal(self): firkant_areal = self.lengde * self.bredde return firkant_areal</pre> <p>På engelsk kalles dette <i>method</i>.</p>

Oppgave 4

Skriv en prosedyre “`storst_av_to`” som tar imot to tall som parametre og skriver ut verdien til det største tallet.

Oppgave 5

(Du skal nå endre på oppgave 5): Skriv en funksjon “`storst_av_to`” som tar imot to tall som parametre og returnerer det største tallet. Deretter skrives tallet ut etter funksjonskallet.