Gruppetime 4

Løkker



Hva er løkker?

- For eksempel: Hvis vi vil skrive ut print(“Hei”) fem ganger går det an å bruke fem print-setninger
- Men hva hvis vi vil skrive ut print(“Hei”) ti tusen ganger?
- Løkker lar oss repetere kode så mange ganger vi ønsker…
- Uten at vi trenger å kopiere koden!

While-løkke

Løkker som kjører til en betingelse er oppfylt/et spesifisert uttrykk er sant

Syntaks:

```
while <uttrykk er sant> :
    <gjør dette>
```

- NB! Evige løkker
- En tommelfingerregel: Det boolske uttrykket må kunne forandre sannhetsverdi
- Nyttig bruk: kommandoløkke

While-løkke - Eksempel

```
x = 5  # kan også endre slik at x er input fra bruker
i = 0  # i er telleren vår, veldig vanlig å bruke variabelnavnet i for teller i løkker.

while i < x :     # kan leses som “så lenge i er mindre enn 5”
 print(i)         # skriver ut telleren vår
 i = i + 1        # deretter øker vi variabelen i med 1
                  #  kan alternativt skrives som i += 1
# så gjentas løkken
```

http://www.pythontutor.com/visualize.html#mode=edit

Oppgave 1: while-løkker

Forklar hva som skjer i denne koden:

```
x = 3

while x < 10:
 x += 1

print(x)
```

Oppgave 2

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

```
tekst = [“hadet”, “på”, “badet”, “din”, “gamle”,
“sjokolade”]

indeks = 0


while indeks < len(tekst):
    print(tekst[indeks])
    indeks += 2
```

Oppgave 2 - Løsning

> hadet

> badet

> gamle

NB! Indekseringen øker med 2 for hver runde.

For-løkker

For-løkker er løkker på samlinger. Vi “går gjennom”/itererer over elementene i en samling.

For-løkker in range

For-løkker kan brukes for å gjøre et visst antall ganger, f.eks:

```
for i in range(tall):                       # gjør noe tall ganger
        …
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 startverdi start og slutte på tallet før slutt
        …
for i in range(2, 5):                       # i vil starte som 2 og gå tom. 4
        …
for i in range(start, slutt, økning):       # i vil ha startverdi start, slutte på slutt og øke med økning
        …
for i in range(2, 15, 3):                   # i vil starte som 2, øke med 3 for hver runde helt frem til i er større enn 15
        …                                   # (eller lik 15), da slutter den. Dvs. 2, 5, 8, 11, 14 > løkken kjøres ikke mer.
```

Hvordan fungerer range?

- Range lager en slags samling (range) av verdier i rekkefølge
- Starter på 0 og slutter på tallet under argumentet
- Eksempel:
  - range(1) -> [0]
  - range(8) -> [0, 1, 2, 3, 4, 5, 6, 7]
- Vi kan også velge et annet tall å begynne på:
  - range(1, 8) -> [1, 2, 3, 4, 5, 6, 7]
- For-løkken itererer da over disse verdiene
- Hvis vi skriver for i in range(9) vil altså i ha en ny verdi fra range-samlingen hver gang

Range lager IKKE en liste!

- Det er lett å forvirre output fra range(n) med en liste, men det er en egen datatype
- Du kan konvertere den til en liste med list(range(n))

Oppgave 3 - for-løkker

Skriv et program med en prosedyre print_hei(), 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 print_hei().

Bruk variabelen a slik at løkken er ferdig etter du har kalt print_hei() 5 ganger.

Løs oppgaven både ved hjelp av while-løkke og for-løkke.

Oppgave 3 løsningsforslag

```
def print_hei() :
 print("Hei")

i = 0
a = 5

while i < a :
 print_hei()
 i += 1

for i in range(a) :
 print_hei()
```

Når bruker man hva?

- For brukes gjerne når man vet hvor mange ganger noe skal skje.
- While kan gjerne brukes når man ikke vet hvor mange ganger noe skal skje, f.eks. hvis man vil lagre et ukjent antall verdier en bruker taster helt til brukeren taster 0, da er det fordelaktig å bruke while.
- Ofte kan man likevel velge hva man ønsker.

For-each-løkker

- Vi kan bruke for-løkker direkte på lister dersom vi ønsker å bruke hvert element til noe
- Syntaks:
  - for <element> in <liste>:
    gjør noe med <element>
- Her er element en “felles variabel” for alle elementene i listen, men peker bare på én om gangen.