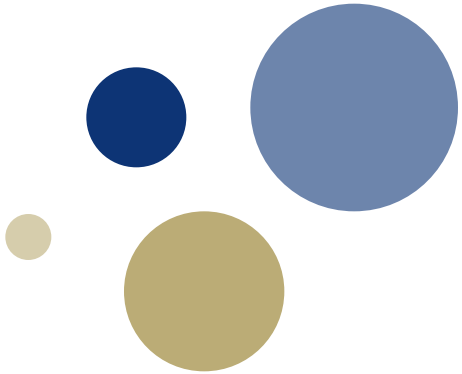
NTNU

Kunnskap for en bedre verden



Python: Løkker

TDT4110 IT Grunnkurs

Professor Guttorm Sindre

Læringsmål og pensum

- Mål
  - Forstå hvorfor vi trenger løkker i programmering
  - Ha kjennskap to ulike typer løkker (while-løkke, for-løkke)
    - Og vite når det er best å bruke hvilken
  - Praktiske ferdigheter i bruk av while-løkke
  - Praktiske ferdigheter i bruk av for-løkke

- Pensum
  - Starting out with Python:

  Chapter 4 Repetition Structures

Intro løkker

Kapittel 5.1

Løkker - gjenta kodelinjer

- HVORFOR: repetere like eller nesten like handlinger
  - Det datamaskiner er best på: samme operasjon om og om igjen
  - Vi ønsker IKKE å skrive samme kode om og om igjen
- Repetisjoner i programmering generelt:
  - To mulige teknikker: Løkker eller rekursjon
- Løkker i Python: to typer
  - while-løkke: ukjent antall repetisjoner
  - for-løkke: kjent antall repetisjoner

Ordet *løkke* brukes fordi vi skal gjenta noe flere ganger (rundt og rundt)

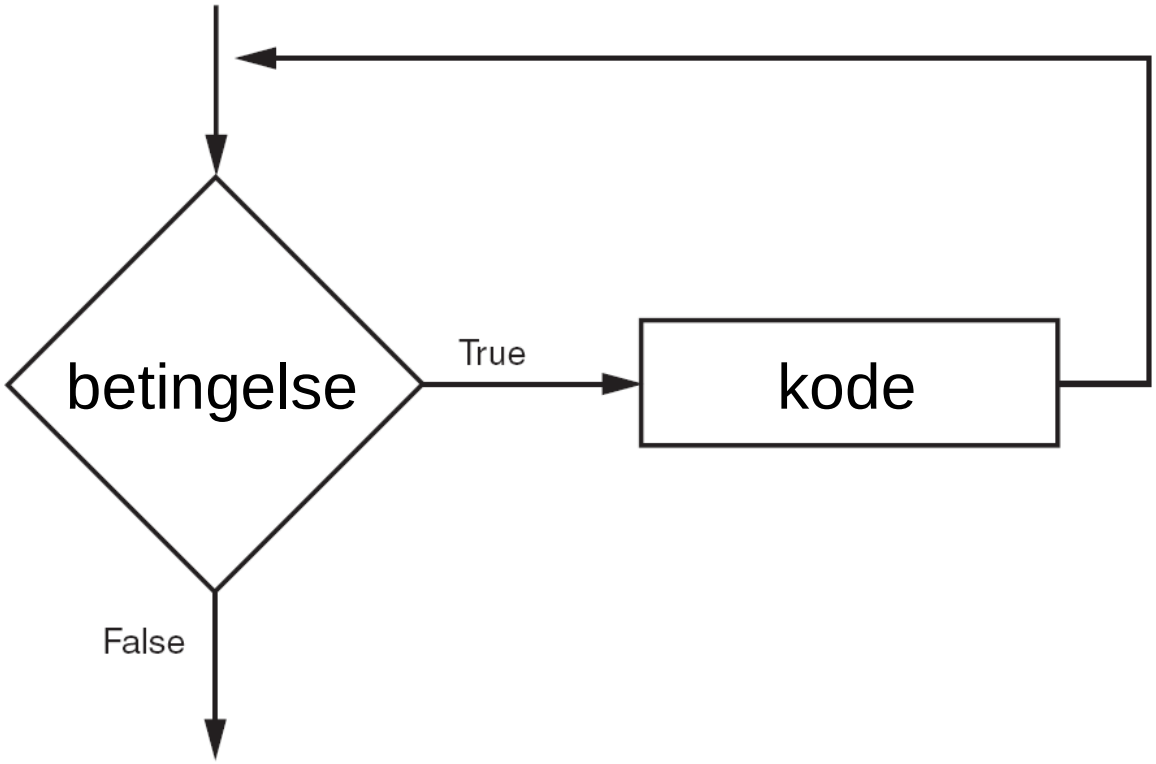
Typisk bruk av ulike løkker

- While-løkke: ukjent antall repetisjoner
  - Fortsette med en handling inntil brukeren ønsker å avslutte
  - Fortsette en beregning inntil et mål er nådd:
    - Matematisk beregning: til man finner nøyaktig nok svar
    - Søking: til ønskede data er funnet
    - Spill: til noen har vunnet
  - Typisk kjennetegn: vet ikke hvor mange ganger som trengs
- For-løkke: kjent antall repetisjoner
  - Enten fast antall, eller
  - Variabelt antall men kjent når løkka starter
  - Samme operasjon for
    - alle elementer i en tabell eller liste
    - alle tall i et intervall

while-løkker

Kapittel 5.2

Flytskjema while-løkke



Skrive en while-løkke

- Generell syntaks for while-løkke i Python:

```
while betingelse:
    kodelinje1
    kodelinje2
kodelinje3
```

- Minner om if:
  - Kun kode med innrykk hører til while-løkka
    - altså ikke kodelinje3
  - hvis betingelse er True blir linje1 og 2 utført
  - ellers blir de ikke utført
- Men viktig forskjell: repetisjon!
  - Kodelinje1 og 2 gjøres om og om igjen…
  - …helt til betingelsen blir False
  - Da går man videre i programmet

Oppgave: while-løkke

- Programmet skal
  - La brukeren skrive inn ett og ett tall ved hjelp av `input()`
  - Avslutte hvis tallet 0 blir gitt inn
  - Da skal vi skrive ut på skjermen:
    - Summen av tallene
  - Mulige utvidelser: også skrive ut
    - Produktet av tallene
    - Det største tallet, og hvilket nummer i rekken dette var

Hent fram koden: while1.py

Evig løkke

- While-løkker bør programmeres slik at de kan avsluttes
  - Noe inni løkka må føre til at betingelsen etter hvert blir False
- Evig løkke:
  - En løkke som ikke har kode som gjør at den stopper
    - For eksempel `while True:`
    - Eller annen betingelse som blir værende True
  - Kan være aktuelt for noen anvendelser
    - For eksempel kontrollsystemer som skal kjøre hele tiden
    - For oss: vanligvis pga. tabber
      - …glemt å inkludere kode som avslutter løkka
      - …feil i koden som skal endre en variabel
      - …feil i testbetingelsen så den aldri blir False
    - Men vi kommer oss ut med **Ctrl-C**

for-løkker

Kapittel 4.3-4.4

for-løkker

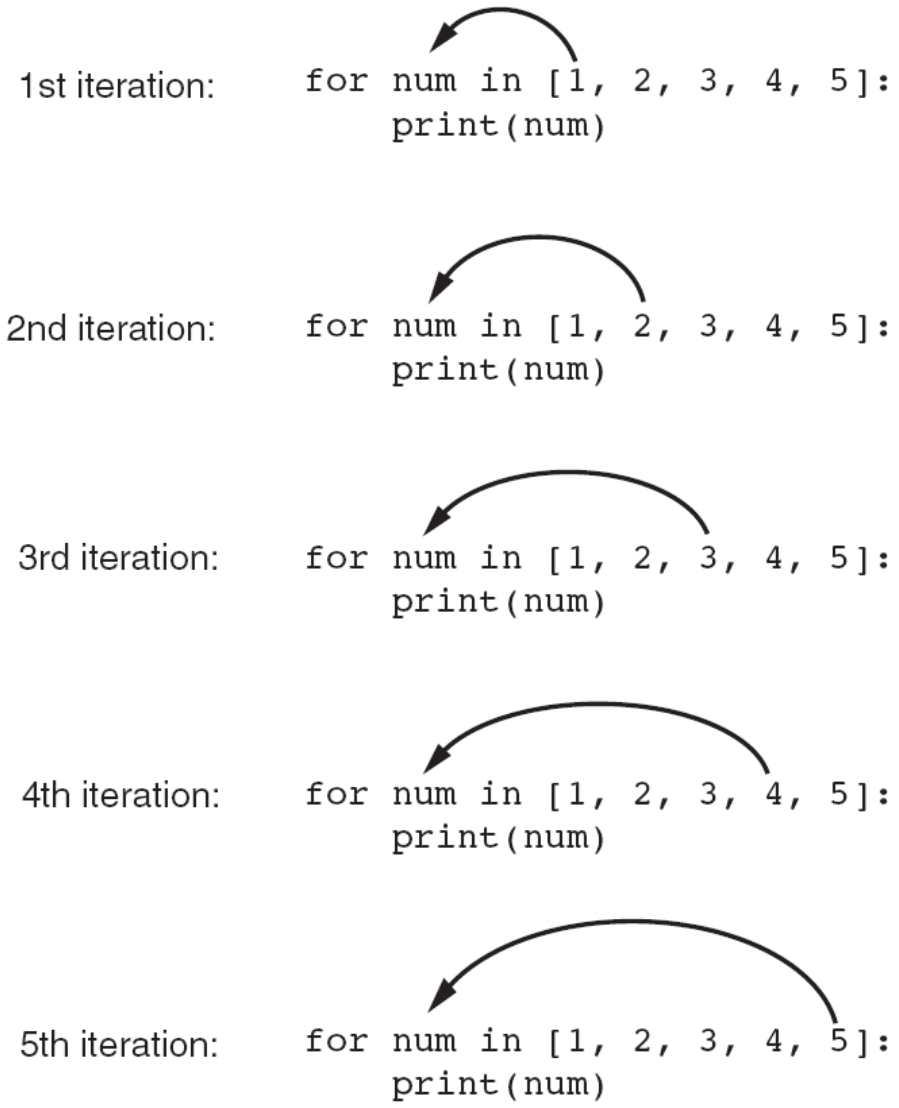
- Gjenta ei kodeblokk *et bestemt antall ganger*
  - Gå igjennom en sekvens (liste), element for element
- Generelt format:

  for variabel **in** liste:

    kodelinjer
- liste kan være
  - En variabel som inneholder ei liste
  - Liste gitt på direkten, f.eks. [‘Arne’ , ’Nina’ , ’Lea’] eller [2, 3, 5, 7, 11, 13]
  - Liste av tall i et intervall, generert ved range( )-funksjonen
- variabel får verdier…
  - Første runde: variabel = første element i lista
  - Andre runde: variabel = andre element i lista
  - …
  - Siste runde: variabel = siste element i lista
- Koden i for-løkka må ha innrykk
  - Når løkka er ferdig, går vi videre med koden nedenfor
  - Hvis lista er tom, hopper vi over løkkekoden og går med en gang videre

Hva skjer når man kjører?

- Kode som skriver ut tallene 1 til 5.
- Variabelen num får verdier fra lista, element for element, for hver runde løkka kjører



for-løkke for variabel som inneholder ei liste

- En for-løkke kan også kjøres på en variabel som inneholder en liste av verdier:

```
navneliste=['Ole', 'Per', 'Oline', 'Anna', 'Frida']
for navn in navneliste:
    print(navn)
```

- Skriver ut navnene til skjerm i rekkefølge
- Ei liste defineres ved
  - [ ] rundt verdier
  - komma mellom elementer
  - kan ha ulike typer verdier
    - strenger, heltall, flyttall, sannhetsverdier osv.

Bruk av range-funksjonen

- range er en funksjon som gjør det enklere å skrive for-løkker
- range-funksjonen lager et objekt av typen *iterable*.
  - *iterable* er et objekt som ligner på ei liste
  - inneholder verdier ei liste kan iterere over, typisk en tallserie
- range kan ha tre varianter:

```
range(til)          # tilsvarer [0,1,2…,til-1]
range(fra, til)     # tilsvarer [fra, fra+1, fra+…n, til-1]
range(fra, til, steg)  # tilsvarer [fra, fra+steg, fra+2*steg, fra+3*steg…]
                            #t.o.m. størst mulige fra+N*steg <= til-1
                            #eller minst mulige fra+N*steg >= til+1 hvis steg < 0
```

Eksempler på range (og liste)

- Range: == Liste:

```
for n in range(5):
    print(n)
```

```
for n in [0,1,2,3,4]:
    print(n)
```

```
for n in range(1,5):
    print(n)
```

```
for num in [1,2,3,4]:
    print(n)
```

```
for n in range(1,6,2):
    print(n)
```

```
for num in [1,3,5]:
    print(n)
```

```
for n in range(10,1,-3):
    print(n)
```

```
for num in [10, 7, 4]:
    print(n)
```

Oppgave: for-løkke

- Skriv et program som
  - Leser inn et positivt heltall
  - Regner ut **n!**
    - n! er 1*2*3*…*n
    - 0! er 0
  - Skriver ut svaret

løsn: fakultet.py

Avslutningsverdi

Kapittel 4.5

Endring av verdier i variable (gi avslutningsverdi i løkker)

Lang-form	Kompakt-form	Hva gjøres
`x = x + 4`	`x += 4`	Øker verdien av x med 4
`x = x - 3`	`x -= 3`	Minsker verdien av x med 3
`x = x * 10`	`x *= 10`	Multipliserer verdien av x med 10
`x = x / 2`	`x /= 2`	Dividerer verdien av x med 2
`x = x % 4`	`x %= 4`	Resten av x delt på 4

Validering av input ved hjelp av løkker

Kapittel 4.6

Validering av input vha løkker

- Brukere kan gi feil input
  - Med vilje
  - Tastefeil
  - Misforståelser
- Ofte viktig å sjekke input før programmet går videre
  - Evt. også tvinge brukeren til korrekt input
  - Dette kan man gjøre vha while-løkke:

```
alder = int(input("Hvor gammel er du? "))
while (alder < 0 or alder > 150):
    print("Feil: Alder må være mellom 0 og 150!")
    alder = int(input("Hvor gammel er du? "))
print("Takk for den du!!!")
```

kode: alder.py

Nøstede løkker

Kapittel 5.7

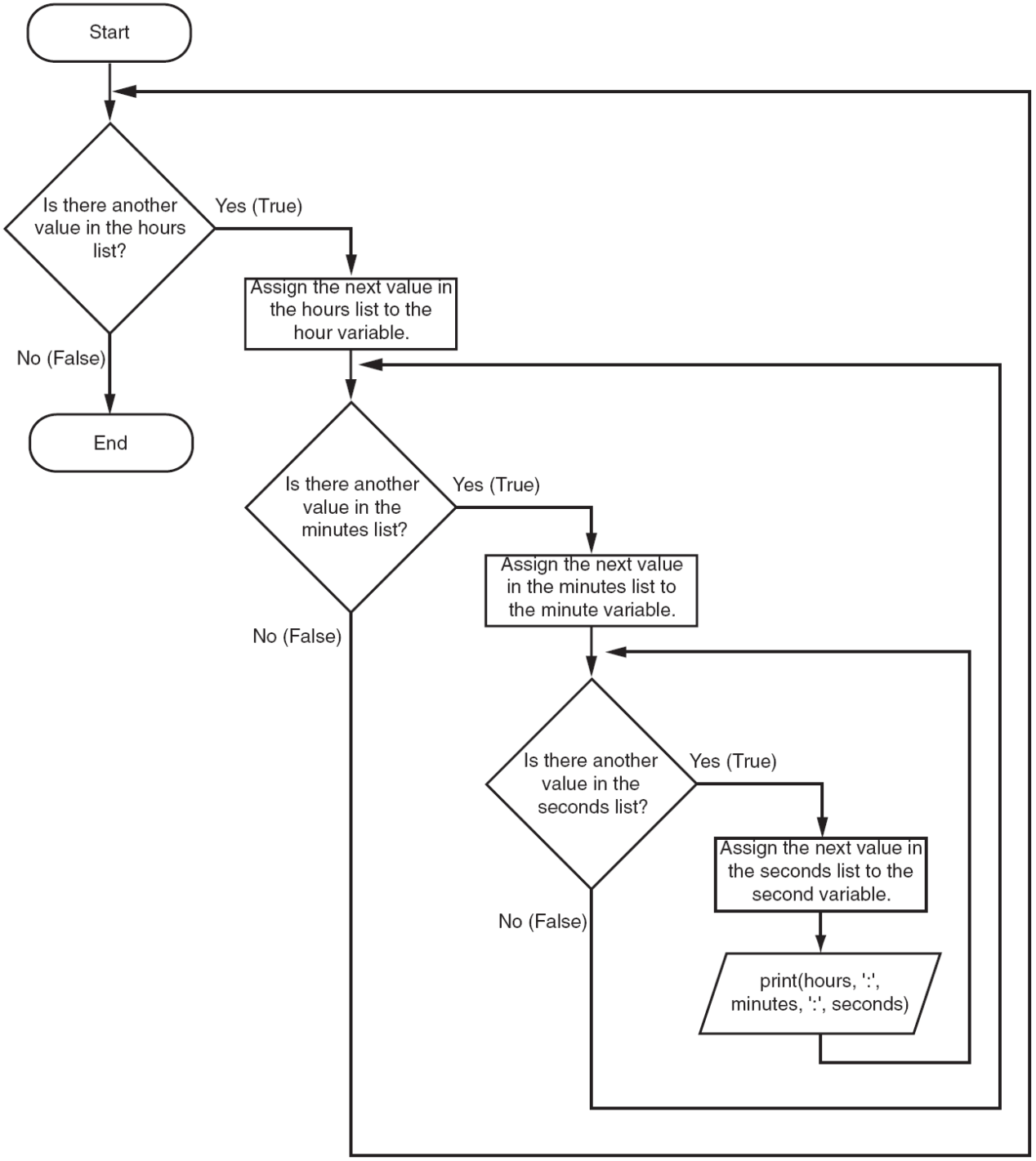
Nøstede løkker

- Løkke inni annen løkke kalles nøstede løkker.
- Nyttig i mange situasjoner
  - Hierarkiske strukturer
    - F.eks. konserner som har datterselskap som har avdelinger som har utført kjøp og salg av ulike typer produkter…
  - Tabeller, matriser
- Tid er et godt eksempel
  - teller først 60 sekunder,
  - øker minutt med 1 osv…
- Utskrift av tid som nøstede løkker:

```
for t in range(24):
    for m in range(60):
        for s in range(60):
            print(t,":",m,":",s)
```

kode: tid.py

Flytskjema for utskrift av tid



Oppgave: nøstet for-løkke

- Lag et program som skriver ut
  - (Med dobbel for-løkke, ikke hardkode hele teksten)

Rutene på sjakkbrettet heter:

A8 B8 C8 D8 E8 F8 G8 H8
A7 B7 C7 D7 E7 F7 G7 H7
A6 B6 C6 D6 E6 F6 G6 H6
A5 B5 C5 D5 E5 F5 G5 H5
A4 B4 C4 D4 E4 F4 G4 H4
A3 B3 C3 D3 E3 F3 G3 H3
A2 B2 C2 D2 E2 F2 G2 H2
A1 B1 C1 D1 E1 F1 G1 H1

løsn: sjakkbrett.py

Oppsummering

- while-løkke: en betingelse avgjør antall iterasjoner:
  - while(betingelse):
    setning(er)

  Setningene utføres så lenge betingelse er True
- for-løkke brukes for et bestemt antall iterasjoner
  - F.eks. gjøre noe for alle elementer i ei liste eller intervall

```
for x in [1, 2, 3, 4]:
for y in [’test’, 3.14, True, 9]:
for i in range(1, 5):
for z in range(1, 5, 2):
```

- Nøstede løkker er løkker inne i andre løkker:

```
for x in [1, 3, 5]:
    for y in range [5, 7, 12]:
        ...
```

Neste uke: Funksjoner (kap.5.1-5.7)

- Aktuelle spørsmål for quiz:
  - «…what happens when the end … is reached» (Checkpoint 5.8)
  - «What is a variable's scope?» (Checkpoint 5.11)
  - «What are the pieces / variables… called?» (Checkpoint 5.13, 5.14)
  - «What is a parameter variable's scope?» (Checkpoint 5.15)
  - «When a parameter is changed…?» (Checkpoint 5.16)
  - «What is the scope of a global variable?» (Checkpoint 5.18)
  - «A variable created inside a function block is known as a…» (Review, Multiple Choice 4)
  - «When possible you should avoid using ____ …» (Review, Multiple Choice 12)
  - «A global variable whose value cannot be changed…» (Review, Multiple Choice 14)
  - « Void functions do not return any value …» (Review, True or False 2)
  - «A statement in a function can access a local variable…» (Review, True or False 7)
  - «Name and describe two parts of a function definition.» (Review, Short Answer 2)
  - «…what happens when the end of the function…» (Review, Short Answer 3)
  - «What will the following program display?» (Algorithm Workbench 4)
- Noen av disse gis i «kahootisert form»
- Pluss 1-2 helt uannonserte spørsmål, men også om løkker