Kap 2: Løkker og lister

Ole Christian Lingjærde, Inst for Informatikk, UiO

30 August - 5 September, 2021 (Del 1 av 2)

Tema denne uken

- Programmering med løkker
- Programmering med lister
- Mange eksempler på Python-programmer

Quiz-oppgaver

- Kommer til å dukke opp på de fleste forelesninger
- Laget for å teste forståelse
- Oftest trivielle å løse med datamaskin
- Stå i mot fristelsen til å bruke datamaskin når du løser dem!

Utskrift på skjerm

Det er tre metoder dere skal kjenne til:

- Uformatert utskrift
- Formatert utskrift med %-operator
- Formatert utskrift med f-strings

Merk: Oppgavene i den tykke læreboka ("A Primer on Scientific Programming in Python") bruker %-operator, mens den tynne læreboka ("Introduction to Scientific Programming with Python") bruker f-strings. Vi kommer primært til å bruke f-strings på forelesningene, men mange oppgaver bruker %-operator.

Uformatert utskrift

```
1 print("Dette er en tekst")
2 print("")
3 print(3)
4 print(3.1)
5 print(1/3)
```

```
Dette er en tekst

3
3.1
0.3333333333333333
```

Merk: vi kommer til å droppe linjenummerne i programmer i fortsettelsen

Formatert utskrift med %-operator

```
høyde = 180
vekt = 75.5  adresse = "Andeby"

print("Høyden er %d centimeter" % høyde)

print("Vekten er %f kilo" % vekt)

print("Vekten er %5.2f kilo" % vekt)

print("Hun bor i %s." % adresse)

print("Hun bor i %10s." % adresse)
```

```
Høyden er 180 centimeter
Vekten er 75.500000 kilo
Vekten er 75.50 kilo
Hun bor i Andeby.
Hun bor i     Andeby.
```

Formatert utskrift med f-strings

```
høyde = 180
vekt = 75.5
adresse = "Andeby"

print(f"Høyden er {høyde} centimeter")

print(f"Vekten er {vekt} kilo")

print(f"Vekten er {vekt:5.2f} kilo")

print(f"Hun bor i {adresse}.")

print(f"Hun bor i {adresse:>10s}.")

print(f"Hun bor i {adresse:<10s}.")
```

```
Høyden er 180 centimeter
Vekten er 75.5 kilo
Vekten er 75.50 kilo
Hun bor i Andeby.
Hun bor i     Andeby.
Hun bor i Andeby     .
```

Quiz

Anta at $x = 0.5$. Hva skriver hver av linjene ut?

```
print("Prisen er x kroner")
#

print(f"Prisen er x kroner")
#

print("Prisen er {x} kroner)
#

print(f"Prisen er {x} kroner")
#

print("Prisen er {x:5.2f} kroner")
#

print("Prisen er x:{5.2f} kroner")
#

print(f"Prisen er x:{5.2f} kroner")
#

print(f"Prisen er {x:5.2f} kroner")
#
```

Svar på Quiz

Virker disse utskriftene og hva skriver de isåfall ut?

```
print("Prisen er x kroner")
# UTSKRIFT: Prisen er x kroner

print("Prisen er {x} kroner")
#

print("Prisen er %4.2f kroner" % x)
#

print(f"Prisen er %4.2f kroner" % x)
#

print(f"Prisen er {x} kroner")
#

print("Prisen er {x:4.2f} kroner")
#

print(f"Prisen er x:{4.2f} kroner")
#

print(f"Prisen er {x:4.2f} kroner")
#
```

Svar på Quiz

Virker disse utskriftene og hva skriver de isåfall ut?

```
print("Prisen er x kroner")
# UTSKRIFT: Prisen er x kroner

print("Prisen er {x} kroner")
# UTSKRIFT: Prisen er {x} kroner

print("Prisen er %4.2f kroner" % x)
#

print(f"Prisen er %4.2f kroner" % x)
#

print(f"Prisen er {x} kroner")
#

print("Prisen er {x:4.2f} kroner")
#

print(f"Prisen er x:{4.2f} kroner")
#

print(f"Prisen er {x:4.2f} kroner")
#
```

Svar på Quiz

Virker disse utskriftene og hva skriver de isåfall ut?

```
print("Prisen er x kroner")
# UTSKRIFT: Prisen er x kroner

print("Prisen er {x} kroner")
# UTSKRIFT: Prisen er {x} kroner

print("Prisen er %4.2f kroner" % x)
# UTSKRIFT: Prisen er 0.50 kroner

print(f"Prisen er %4.2f kroner" % x)
#

print(f"Prisen er {x} kroner")
#

print("Prisen er {x:4.2f} kroner")
#

print(f"Prisen er x:{4.2f} kroner")
#

print(f"Prisen er {x:4.2f} kroner")
#
```

Svar på Quiz

Virker disse utskriftene og hva skriver de isåfall ut?

```
print("Prisen er x kroner")
# UTSKRIFT: Prisen er x kroner

print("Prisen er {x} kroner")
# UTSKRIFT: Prisen er {x} kroner

print("Prisen er %4.2f kroner" % x)
# UTSKRIFT: Prisen er 0.50 kroner

print(f"Prisen er %4.2f kroner" % x)
# UTSKRIFT: Prisen er 0.50 kroner

print(f"Prisen er {x} kroner")
#

print("Prisen er {x:4.2f} kroner")
#

print(f"Prisen er x:{4.2f} kroner")
#

print(f"Prisen er {x:4.2f} kroner")
#
```

Svar på Quiz

Virker disse utskriftene og hva skriver de isåfall ut?

```
print("Prisen er x kroner")
# UTSKRIFT: Prisen er x kroner

print("Prisen er {x} kroner")
# UTSKRIFT: Prisen er {x} kroner

print("Prisen er %4.2f kroner" % x)
# UTSKRIFT: Prisen er 0.50 kroner

print(f"Prisen er %4.2f kroner" % x)
# UTSKRIFT: Prisen er 0.50 kroner

print(f"Prisen er {x} kroner")
# UTSKRIFT: Prisen er 0.5 kroner

print("Prisen er {x:4.2f} kroner")
#

print(f"Prisen er x:{4.2f} kroner")
#

print(f"Prisen er {x:4.2f} kroner")
#
```

Svar på Quiz

Virker disse utskriftene og hva skriver de isåfall ut?

```
print("Prisen er x kroner")
# UTSKRIFT: Prisen er x kroner

print("Prisen er {x} kroner")
# UTSKRIFT: Prisen er {x} kroner

print("Prisen er %4.2f kroner" % x)
# UTSKRIFT: Prisen er 0.50 kroner

print(f"Prisen er %4.2f kroner" % x)
# UTSKRIFT: Prisen er 0.50 kroner

print(f"Prisen er {x} kroner")
# UTSKRIFT: Prisen er 0.5 kroner

print("Prisen er {x:4.2f} kroner")
# UTSKRIFT: Prisen er {x:4.2f} kroner

print(f"Prisen er x:{4.2f} kroner")
#

print(f"Prisen er {x:4.2f} kroner")
#
```

Svar på Quiz

Virker disse utskriftene og hva skriver de isåfall ut?

```
print("Prisen er x kroner")
# UTSKRIFT: Prisen er x kroner

print("Prisen er {x} kroner")
# UTSKRIFT: Prisen er {x} kroner

print("Prisen er %4.2f kroner" % x)
# UTSKRIFT: Prisen er 0.50 kroner

print(f"Prisen er %4.2f kroner" % x)
# UTSKRIFT: Prisen er 0.50 kroner

print(f"Prisen er {x} kroner")
# UTSKRIFT: Prisen er 0.5 kroner

print("Prisen er {x:4.2f} kroner")
# UTSKRIFT: Prisen er {x:4.2f} kroner

print(f"Prisen er x:{4.2f} kroner")
# FEILMELDING: SyntaxError: invalid syntax

print(f"Prisen er {x:4.2f} kroner")
#
```

Svar på Quiz

Virker disse utskriftene og hva skriver de isåfall ut?

```
print("Prisen er x kroner")
# UTSKRIFT: Prisen er x kroner

print("Prisen er {x} kroner")
# UTSKRIFT: Prisen er {x} kroner

print("Prisen er %4.2f kroner" % x)
# UTSKRIFT: Prisen er 0.50 kroner

print(f"Prisen er %4.2f kroner" % x)
# UTSKRIFT: Prisen er 0.50 kroner

print(f"Prisen er {x} kroner")
# UTSKRIFT: Prisen er 0.5 kroner

print("Prisen er {x:4.2f} kroner")
# UTSKRIFT: Prisen er {x:4.2f} kroner

print(f"Prisen er x:{4.2f} kroner")
# FEILMELDING: SyntaxError: invalid syntax

print(f"Prisen er {x:4.2f} kroner")
# UTSKRIFT: Prisen er 0.50 kroner
```

Lage en tabell

Vi ønsker å lage et program som skriver ut denne tabellen:

```
-20.0  -4.0
-15.0   5.0
-10.0  14.0
 -5.0  23.0
  0.0  32.0
  5.0  41.0
 10.0  50.0
 15.0  59.0
 20.0  68.0
 25.0  77.0
 30.0  86.0
 35.0  95.0
 40.0 104.0
```

Venstre kolonne er Celcius-grader (C), høyre kolonne er tilsvarende Fahrenheit-grader (F). Formelen er $F = (9/5)C + 32$.

Enkel (naiv) løsning

Det er lett å lage en enkelt linje i tabellen:

```
C = -20
F = 9/5 * C + 32
print(f"{C:3.1f}  {F:3.1f}")
```

Vi kan gjøre det samme mange ganger:

```
C = -20;  F = 9/5*C + 32;  print(f"{C:3.1f}  {F:3.1f}")

C = -15;  F = 9/5*C + 32;  print(f"{C:3.1f}  {F:3.1f}")

C = -10;  F = 9/5*C + 32;  print(f"{C:3.1f}  {F:3.1f}")

...

C =  40;  F = 9/5*C + 32;  print(f"{C:3.1f}  {F:3.1f}")
```

Enkel (naiv) løsning

Løsningen over er ikke spesielt god:

- Tungvint hvis tabellen har mange (f.eks. 1000) rader
- Lett å programmere feil, f.eks. hoppe over en tabellrad
- Vi utnytter ikke at samme operasjon utføres hver gang

Bedre alternativ: lage en programløkke!

Programløkker i Python

Det er to forskjellige løkketyper i Python og begge er viktige:

While-løkker: Dette er den mest generelle løkketypen. Når vi trenger en løkke, kan vi *alltid* benytte en while-løkke.

For-løkker: Passer ikke til alle problemer, men når de *kan* brukes gir de ofte penere kode. Vi kommer til å bruke denne mest.

Eksempel på while-løkke

```
a = 1
while a < 50:
    print(a)
    a = a * 2
```

```
1
2
4
8
16
32
```

Fakta om while-løkker

- Repeterer programsetninger helt til en gitt logisk betingelse ikke lenger er sann.
- Generell form:

```
while logiskBetingelse:
    <setning 1>
    <setning 2>
    ...
    <setning n>
```

- Alle setningene i løkka må ha innrykk

While-løkke for temperatur-tabellen

```
C = -20

while C <= 40:
  F = (9/5)*C + 32
  print(f"{C:3.1f}  {F:3.1f}")
  C = C + 5


print("Når vi kommer hit er løkka slutt")
print(f"Nå er C = {C}")
```

While-løkke for temperatur-tabellen

```
-20.0  -4.0
-15.0  5.0
-10.0  14.0
-5.0  23.0
0.0  32.0
5.0  41.0
10.0  50.0
15.0  59.0
20.0  68.0
25.0  77.0
30.0  86.0
35.0  95.0
40.0  104.0
Når vi kommer hit er løkka slutt
Nå er C = 45
```

Hva skjer egentlig når løkka kjører?

```
C = -20
while C <= 40:
  F = (9/5)*C + 32
  print(f"{C:3.1f}  {F:3.1f}")
  C = C + 5
```

Når løkka over kjører skjer følgende:

```
C = -20

C <= 40                                  # True
F = (9/5)*C + 32                         # F beregnes til -4
print(f"{C:3.1f}     {F:3.1f}")          # Skriv ut "-20.0, -4.0"
C = C + 5

C <= 40                                  # True
F = (9/5)*C + 32                         # F beregnes til 5
print(f"{C:3.1f}     {F:3.1f}")          # Skriv ut "-15.0, 5.0"
C = C + 5

C <= 40                                  # True
(osv ...)
```

Logiske uttrykk

Et *logisk uttrykk* (= Boolsk uttrykk) er noe Python kan regne ut og som gir True eller False som resultat:

```
C = 18          # Ett likhetstegn betyr tilordning
C == 18         # Er C lik 18? (True)
C == 19         # Er C lik 19? (False)
C != 20         # Er C forskjellig fra 20 (True)
C > 20          # Er C større en 20? (False)
C >= 20         # Er C større enn eller lik 20? (False)
C < 20          # Er C mindre enn 20? (True)
C <= 20         # Er C mindre enn eller lik 20? (True)
```

Operatoren **and**

```
True and True
True and False
False and True
False and False
```

```
True
False
False
False
```

Operatoren **and**

Eksempel på bruk av **and**:

```
x = 0; y = 0
while x<3 and y<3:
    print(f"x+y = {x+y:d}")
    x = x + 1
    y = 2*y + x
```

x	y	x<3 and y<3	Utskrift	Oppdatering x	Oppdatering y

Operatoren **and**

Eksempel på bruk av **and**:

```
x = 0; y = 0
while x<3 and y<3:
    print(f"x+y = {x+y:d}")
    x = x + 1
    y = 2*y + x
```

x	y	x<3 and y<3	Utskrift	Oppdatering x	Oppdatering y
0	0	True	x+y = 0	x = 0 + 1	y = 2*0 + 1

Operatoren **and**

Eksempel på bruk av **and**:

```
x = 0; y = 0
while x<3 and y<3:
    print(f"x+y = {x+y:d}")
    x = x + 1
    y = 2*y + x
```

x	y	x<3 and y<3	Utskrift	Oppdatering x	Oppdatering y
0	0	True	x+y = 0	x = 0 + 1	y = 2*0 + 1
1	1	True	x+y = 2	x = 1 + 1	y = 2*1 + 1

Operatoren **and**

Eksempel på bruk av **and**:

```
x = 0; y = 0
while x<3 and y<3:
    print(f"x+y = {x+y:d}")
    x = x + 1
    y = 2*y + x
```

x	y	x<3 and y<3	Utskrift	Oppdatering x	Oppdatering y
0	0	True	x+y = 0	x = 0 + 1	y = 2*0 + 1
1	1	True	x+y = 2	x = 1 + 1	y = 2*1 + 1
2	4	False			

Operatoren **or**

```
True or True
True or False
False or True
False or False
```

```
True
True
True
False
```

Quiz

Hva skriver dette programmet ut?

```
x = 0; y = 0
while x<=2 or y<2:
    print(f"x+y = {x+y:d}")
    x = x + 1
    y = 2*x + 2
```

Quiz

Hva skriver dette programmet ut?

```
x = 0; y = 0
while x<=2 or y<2:
    print(f"x+y = {x+y:d}")
    x = x + 1
    y = 2*x + 2
```

x	y	x<=2 or y<2	Utskrift	Oppdatering x	Oppdatering y

Quiz

Hva skriver dette programmet ut?

```
x = 0; y = 0
while x<=2 or y<2:
    print(f"x+y = {x+y:d}")
    x = x + 1
    y = 2*x + 2
```

x	y	x<=2 or y<2	Utskrift	Oppdatering x	Oppdatering y
0	0	True	x+y = 0	x = 0 + 1	y = 2*1 + 2

Quiz

Hva skriver dette programmet ut?

```
x = 0; y = 0
while x<=2 or y<2:
    print(f"x+y = {x+y:d}")
    x = x + 1
    y = 2*x + 2
```

x	y	x<=2 or y<2	Utskrift	Oppdatering x	Oppdatering y
0	0	True	x+y = 0	x = 0 + 1	y = 2*1 + 2
1	4	True	x+y = 5	x = 1 + 1	y = 2*2 + 2

Quiz

Hva skriver dette programmet ut?

```
x = 0; y = 0
while x<=2 or y<2:
    print(f"x+y = {x+y:d}")
    x = x + 1
    y = 2*x + 2
```

x	y	x<=2 or y<2	Utskrift	Oppdatering x	Oppdatering y
0	0	True	x+y = 0	x = 0 + 1	y = 2*1 + 2
1	4	True	x+y = 5	x = 1 + 1	y = 2*2 + 2
2	6	True	x+y = 8	x = 2 + 1	y = 2*3 + 2

Quiz

Hva skriver dette programmet ut?

```
x = 0; y = 0
while x<=2 or y<2:
    print(f"x+y = {x+y:d}")
    x = x + 1
    y = 2*x + 2
```

x	y	x<=2 or y<2	Utskrift	Oppdatering x	Oppdatering y
0	0	True	x+y = 0	x = 0 + 1	y = 2*1 + 2
1	4	True	x+y = 5	x = 1 + 1	y = 2*2 + 2
2	6	True	x+y = 8	x = 2 + 1	y = 2*3 + 2
3	8	False			

Quiz

Hvilke logiske verdier gir disse uttrykkene?

```
x = 1; y = 2

x > y
#

(x != y) and (x < y)
#

not (x < y and y <= x)
#

(x < y and x > y) or (x != y)
#

y**y > x + 2*y
#
```

Svar på Quiz

Hvilke logiske verdier gir disse uttrykkene?

```
x = 1; y = 2

x > y
# SVAR: False

(x != y) and (x < y)
#

not (x < y and y <= x)
#

(x < y and x > y) or (x != y)
#

y**y > x + 2*y
#
```

Svar på Quiz

Hvilke logiske verdier gir disse uttrykkene?

```
x = 1; y = 2

x > y
# SVAR: False

(x != y) and (x < y)
# SVAR: True      (True and True --> True)

not (x < y and y <= x)
#

(x < y and x > y) or (x != y)
#

y**y > x + 2*y
#
```

Svar på Quiz

Hvilke logiske verdier gir disse uttrykkene?

```
x = 1; y = 2

x > y
# SVAR: False

(x != y) and (x < y)
# SVAR: True

not (x < y and y <= x)
# SVAR: True      (not (True and False) --> not (False) --> True)

(x < y and x > y) or (x != y)
#

y**y > x + 2*y
#
```

Svar på Quiz

Hvilke logiske verdier gir disse uttrykkene?

```
x = 1; y = 2

x > y
# SVAR: False

(x != y) and (x < y)
# SVAR: True

not (x < y and y <= x)
# SVAR: True

(x < y and x > y) or (x != y)
# SVAR: True      (True and False) or (True) --> True

y**y > x + 2*y
#
```

Svar på Quiz

Hvilke logiske verdier gir disse uttrykkene?

```
x = 1; y = 2

x > y
# SVAR: False

(x != y) and (x < y)
# SVAR: True

not (x < y and y <= x)
# SVAR: True

(x < y and x > y) or (x != y)
# SVAR: True

y**y > x + 2*y
# SVAR: False      (4 > 5 --> False)
```

For-løkker

For-løkker er et alternativ til while-løkker. Generell form:

```
for e in liste:
    <setninger som bruker verdien e>
```

Merk:

- Alle setninger i løkka må ha innrykk (som for while-løkker)
- For-løkken trenger en *liste* (som vi foreløpig ikke har lært om)

Lister

Frem til nå har en variabel hatt plass til en verdi:

```
C = -10
x = 2.255
s = 'Hello'
```

Vi kan også lage dataobjekter og variable som inneholder flere verdier:

```
C = [-10, -5, 0, 5, 10]
x = [2.255, 3.634, 6.4]
s = ['Hello', 'my', 'friend']
u = [3, 3.14, 'pi']
```

Få tak i enkeltverdier i en liste

Vi kan lett få tak i enkeltverdier i en liste.

Da bruker vi elementenes plassering (indeks) i listen:

```
a = [3.14, 3.1415926, 999999]
print(len(a))
print(a[0])
print(a[1])
print(a[2])
```

```
3
3.14
3.1415926
999999
```

Endre verdier i en liste

Det er lett å endre enkeltelementer i en liste:

```
a = [3.14, 3.1415926, 999999]
a[1] = 0.1
print(a)
```

```
[3.14, 0.1, 999999]
```

Fire nyttige liste-operasjoner

Fire nyttige listeoperasjoner: append, extend, insert, delete

```
a = [-10, -5, 0, 5, 10]
a.append(99)            # Legg til et nytt element på slutten
print(a)

a = a + [100, 200]      # Slå sammen (engelsk: concatenate) to lister
print(a)

a.insert(2, -99)        # Sett inn -99 slik at den får indeks 2
print(a)

del a[2]                # Fjern elementet med indeks 2
print(a)
```

```
[-10, -5, 0, 5, 10, 99]
[-10, -5, 0, 5, 10, 99, 100, 200]
[-10, -5, -99, 0, 5, 10, 99, 100, 200]
[-10, -5, 0, 5, 10, 99, 100, 200]
```

Søke etter verdier i en liste

```
a = [-10, -5, 0, 5, 10]
print(a.index(10))          # Hvor ligger verdien 10?
print(5 in a)               # Finnes verdien 5 i listen?

b = [1, 2, 1]
print(b.index(1))           # Bare treff nr 1 rapporteres
print(b.count(1))           #Hvor mange ganger fins verdien 1 i listen?

k1 = b.index(1)             #Treff nr 1
k2 = b.index(1, k1+1)       #Treff nr 2
print(k1, k2)
```

Start å lete i posisjon k1+1 (= 1)

```
4
True
0
2
0 2
```

Negative indekser

```
a = [-10, -5, 0, 5, 10]
print(a[-1])          # Siste element
print(a[-2])          # Nest siste element

somelist = ['eple', 'appelsin', 'ananas']
a, b, c = somelist  # Tilordne direkte til variabler
print(a)
print(b)
print(c)
```

```
10
5
eple
appelsin
ananas
```

Løpe gjennom en liste med en for-løkke

Vi kan bruke en for-løkke til å løpe gjennom alle verdiene i en liste:

```
CListe = [-20, -15, -10, 5, 0]
for C in CListe:
    F = (9.0/5) * C + 32
    print(f"{C:3.1f}  {F:3.1f}")
```

```
-20.0  -4.0
-15.0  5.0
-10.0  14.0
5.0  41.0
0.0  32.0
```

Tilbake til eksemplet med temperaturtabell

```
Cdegrees = [-20, -15, -10, -5, 0, 5, 10, 15,
            20, 25, 30, 35, 40]
for C in Cdegrees:
    F = (9.0/5)*C + 32
    print(f"{C:3.1f}  {F:3.1f}")
```

```
-20.0  -4.0
-15.0  5.0
-10.0  14.0
-5.0  23.0
0.0  32.0
5.0  41.0
10.0  50.0
15.0  59.0
20.0  68.0
25.0  77.0
30.0  86.0
35.0  95.0
40.0  104.0
```

En for-løkke kan alltid oversettes til en while-løkke

Vi kan oversette for-løkker til while-løkker. Eksempel:

```
minliste = [0,2,4]
for x in minliste:
    print(x**2)
```

kan også skrives slik:

```
minliste = [0,2,4]
k = 0
while k < len(minliste):
    x = minliste[k]
    print(x**2)
    k += 1
```

Motsatt vei er ikke alltid mulig (med pen programmering).

Quiz

Lag en for-løkke som skriver ut tallene

1, 2, 4, 8, 16, 32, 64, 128, 256, 512

ved å bruke formelen $x_i = 2^i$ for $i = 0, 1, 2, \ldots, 9$.

Quiz (svar)

```
for i in [0,1,2,3,4,5,6,7,8,9]:
    xi = 2**i
    print(xi)
```

Alternativ løsning:

```
for i in range(10):
    xi = 2**i
    print(xi)
```

Plenumsoppgaver

Oppgaver i plenum (gjennomgås på torsdag):

- 1.4 (lengthconversion.py, side 43)
- 1.12 (egg.py, side 46)