A Innføring i Python

Overskriftene viser til delkapitlene i bilag A i boka. Først kommer et formelark for hele bilaget. Under hver overskrift står gjennomregnede eksempler med løsning, og deretter oppgaver du skal løse uten PC. Skriv svarene for hånd, slik som på eksamen. Fasiten står bakerst.

Formelark

GRUNNLEGGENDE PYTHON

Regneoperasjoner

$+ \; - \; * \; / \; ** \; // \; \%$

Løkker

```
for x in range(0, 5):
```

Vilkår

```
if elif else
```

Funksjon

```
def f(x): return 2*x + 3
```

- Gange skrives `*`, potens `**`, og desimaltegnet er punktum.
- Koden under `if`, `for`, `while` og `def` rykkes inn.
- Plassene i en liste telles fra 0.

A.1.1 Tall og variabler

EKSEMPEL *Regne ut i Python*

Hva skriver programmet ut?

```
print(2 + 3 * 4)
print((2 + 3) * 4)
print(2**10)
print(7 / 2)
print(7 // 2)
print(7 % 2)
```

Løsning

Gange kommer før pluss, så første linje er 14. Parentesen i andre linje gir 20. Videre er $2^{10} = 1\,024$, $7 : 2 = 3{,}5$, og 2 går opp i 7 tre hele ganger med rest 1.

■

EKSEMPEL *Pris etter rabatt*

En vare koster 250 kr, og du får 20 % rabatt. Skriv et program som regner ut den nye prisen med vekstfaktor.

Løsning

```
pris = 250
rabatt = 20
ny = pris * (1 - rabatt / 100)
print(ny)
```

Utskriften er 200,0, fordi $250 \cdot 0{,}8 = 200$. Prisen etter 20 % rabatt er 200 kr.

■

EKSEMPEL *Variabler som endres*

Hva skriver programmet ut?

```
x = 5
x = x + 2
x = x * 2
print(x)
```

Løsning

Først er `x = 5`, så `x = 5 + 2 = 7`, så `x = 7 * 2 = 14`. Programmet skriver ut `14`.

■

EKSEMPEL *Avrunding*

Hva skriver programmet ut?

```
print(8 / 2)
print(round(100 / 7, 2))
print(0.1 + 0.2)
```

Løsning

Det første er `4.0`, fordi `/` gir et desimaltall. Det andre er `14.29`. Det tredje er `0.30000000000000004`: Datamaskinen lagrer desimaltall med begrenset presisjon, så små avvik kan oppstå. Derfor avrunder vi ofte svaret.

■

EKSEMPEL *abc-formelen i Python*

Programmet løser $x^2 - 2x - 15 = 0$. Hva skriver det ut?

```
a = 1
b = -2
c = -15
d = b**2 - 4*a*c
x1 = (-b + d**0.5) / (2*a)
x2 = (-b - d**0.5) / (2*a)
print(x1, x2)
```

Løsning

Her er $d = 4 + 60 = 64$ og $\sqrt{64} = 8$, så $x = \dfrac{2 \pm 8}{2}$. Utskriften er `5.0 -3.0`. Parentesene rundt `2*a` er nødvendige: Uten dem deler Python på 2 og ganger med `a` etterpå.

■

A.1 Hva blir svaret i Python?

a) `5 + 2 * 3`

b) `(5 + 2) * 3`

c) `2**4`

d) `20 - 4 * 3`

A.2 Hva blir svaret?

a) `9 // 2`

b) `9 % 2`

c) `9 / 2`

A.3 Hva skriver programmet ut?

Python

```
print(10 / 4)
print(12 / 4)
```

A.4 Hva skriver programmet ut?

Python

```
x = 3
x = x + 4
print(x)
```

A.5 Hva skriver programmet ut?

Python

```
a = 2
b = a * 5
a = b - 1
print(a, b)
```

A.6 Hva skriver programmet ut?

Python

```
pris = 400
rabatt = 10
ny = pris * (1 - rabatt / 100)
print(ny)
```

A.7 Hva skriver programmet ut?

Python

```
print(round(2 / 3, 2))
```

A.8 Hva skriver programmet ut?

Python

```
print(2**-1)
```

A.9 En av linjene gir feilmelding. Hvilken, og hvorfor?

Python

```
x = 4 * 5
y = 3x
print(y)
```

A.10 Hva blir verdien av `5*x**2 - 3` når `x = 2`?

A.11 Hva skriver programmet ut?

Python

```
minutter = 150
print(minutter / 60)
```

A.12 Hva skriver programmet ut?

Python

```
print(17 % 5, 17 // 5)
```

A.13 Hva skriver programmet ut?

Python

```
x = 4
y = x**2 + 2 * x
print(y)
```

A.14 Forklar hva `=` betyr i Python, og hvorfor `x = x + 1` er lov.

A.15 Hva blir svaret i Python?

a) `25**0.5`

b) `10**-3`

c) `27**(1/3)`

A.16 Hva skriver programmet ut?

Python

```
a = 1
b = -6
c = 5
d = b**2 - 4*a*c
x1 = (-b + d**0.5) / (2*a)
x2 = (-b - d**0.5) / (2*a)
print(x1, x2)
```

A.1.2 Vilkår og løkker

EKSEMPEL *Billettpris*

Programmet velger billettpris ut fra alderen. Hva skriver det ut når `alder = 14`? Hva når `alder = 70`?

```
alder = 14
if alder < 16:
    pris = 50
elif alder < 67:
    pris = 100
else:
    pris = 60
print(pris)
```

Løsning

For 14 år stemmer `alder < 16`, så prisen er 50. For 70 år stemmer ingen av de to første vilkårene, så `else` gir prisen 60. ∎

EKSEMPEL *Verditabell*

Skriv et program som lager en verditabell for $f(x) = 2x + 3$ for $x = 0, 1, ..., 4$.

Løsning

```
for x in range(0, 5):
    print(x, 2 * x + 3)
```

Første kolonne er x, og andre kolonne er $f(x)$. ∎

EKSEMPEL *Summere og telle*

a) Hva er summen av tallene fra 1 til 5?

b) Hvor mange tall fra 1 til 20 kan deles på 3?

```
total = 0
for i in range(1, 6):
    total = total + i
print(total)

antall = 0
for x in range(1, 21):
    if x % 3 == 0:
        antall = antall + 1
print(antall)
```

Løsning

Summen er $1 + 2 + 3 + 4 + 5 = 15$. Tallene som kan deles på 3, er $3, 6, 9, 12, 15, 18$, altså 6 stykker. ∎

EKSEMPEL *Når når sparebeløpet 15 000 kr?*

10 000 kr står i banken til 4 % rente per år. Etter hvor mange år er beløpet minst 15 000 kr?

```
belop = 10000
ar = 0
while belop < 15000:
    belop = belop * 1.04
    ar = ar + 1
print(ar)
```

Løsning

Utskriften er `11`: Etter 10 år er beløpet 14 802, og etter 11 år er det 15 395. ∎

A.17 Hva skriver programmet ut når `alder = 10`? Hva når `alder = 30`?

Python

```
if alder < 16:
    pris = 50
elif alder < 67:
    pris = 100
else:
    pris = 60
print(pris)
```

A.18 Hva skriver programmet ut?

Python

```
for x in range(3):
    print(x)
```

A.19 Hva skriver programmet ut?

Python

```
for x in range(1, 4):
    print(x * 2)
```

A.20 Hva skriver programmet ut?

Python

```
for x in range(2, 9, 3):
    print(x)
```

A.21 Hva skriver programmet ut?

Python

```
total = 0
for i in range(1, 5):
    total = total + i
print(total)
```

A.22 Hvor mange av tallene fra 1 til 20 kan deles på 5? Programmet teller dem. Hva skriver det ut?

Python

```
antall = 0
for x in range(1, 21):
    if x % 5 == 0:
        antall = antall + 1
print(antall)
```

A.23 Hva skriver programmet ut?

Python

```
x = 1
while x < 20:
    x = x * 3
print(x)
```

A.24 Hva skriver programmet ut?

Python

```
belop = 100
ar = 0
while belop < 150:
    belop = belop * 1.2
    ar = ar + 1
print(ar)
```

A.25 Programmet stopper aldri. Hva er galt?

Python

```
x = 1
while x <= 5:
    print(x)
```

A.26 Hva skriver programmet ut?

Python

```
for x in range(0, 4):
    print(x, 3 * x - 1)
```

A.27 Hva er forskjellen på `=` og `==` i Python?

A.28 Hva skriver programmet ut?

Python

```
n = 0
for x in range(1, 11):
    if x % 2 == 0:
        n = n + 1
print(n)
```

A.29 Hvilke tall gir `range(1, 5)`? Er 5 med?

A.30 Hva skriver programmet ut?

Python

```
a = 2
b = 1
c = 3
d = b**2 - 4*a*c
if d > 0:
    print("to løsninger")
elif d == 0:
    print("én løsning")
else:
    print("ingen løsning")
```

A.1.3 Lister, funksjoner og grafer

EKSEMPEL *Gjennomsnittstemperatur*

Hva er gjennomsnittet av 3, 5, -2 og 4?

```
temperaturer = [3, 5, -2, 4]
print(sum(temperaturer))
print(len(temperaturer))
print(sum(temperaturer) / len(temperaturer))
```

Løsning

Summen er 10, antallet er 4, og gjennomsnittet er $\frac{10}{4} = 2{,}5$.

■

EKSEMPEL *Løkke gjennom en liste*

Hva skriver programmet ut?

```
temperaturer = [3, 5, -2, 4]
for t in temperaturer:
    if t < 0:
        print("frost", t)
```

Løsning

Bare -2 er under null, så programmet skriver ut `frost -2`.

■

EKSEMPEL *Funksjon og tabell*

Hva skriver programmet ut?

```
def f(x):
    return 2 * x + 3

print(f(4))
for x in range(0, 3):
    print(x, f(x))
```

Løsning

$f(4) = 11$, og tabellen viser $f(0) = 3$, $f(1) = 5$ og $f(2) = 7$.

■

EKSEMPEL *Saldo med rente*

Lag en funksjon som regner ut saldoen etter n år når startbeløpet er K og renten er p %. Hva er saldoen for 10 000 kr i 5 år til 3 %?

Løsning

```
def saldo(K, p, n):
    return K * (1 + p / 100) ** n

print(round(saldo(10000, 3, 5), 2))
```

Utskriften er `11592.74`, så saldoen er omtrent 11 593 kr.

■

EKSEMPEL *Gjennomsnittlig vekstfart*

La $f(x) = x^2 - 4x + 3$. Hva skriver programmet ut?

```
def f(x):
    return x**2 - 4*x + 3

def vekstfart(a, b):
    return (f(b) - f(a)) / (b - a)

print(vekstfart(1, 4))
print(round(vekstfart(2, 2.001), 3))
```

Løsning

$f(1) = 0$ og $f(4) = 3$, så vekstfarten er $\frac{3-0}{3} = 1$. Den andre linja bruker et svært lite intervall og gir `0.001`, som er nesten 0. Det er den momentane vekstfarten $f'(2) = 0$: I $x = 2$ har grafen bunnpunkt.

■

EKSEMPEL *Tegne en graf*

Hva gjør hver linje i programmet?

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-1, 5, 100)
y = x**2 - 4*x + 3
plt.plot(x, y)
plt.grid()
plt.show()
```

Løsning

```
#| echo: false
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-1, 5, 100)
y = x**2 - 4*x + 3
fig, ax = plt.subplots(figsize=(4.4, 3.2))
ax.plot(x, y, lw=2)
ax.grid(alpha=0.3)
ax.axhline(0, color="k", lw=0.8)
ax.axvline(0, color="k", lw=0.8)
plt.show()
```

`np.linspace` lager x-verdiene, `y = x**2 - 4*x + 3` regner ut y for hver, `plt.plot(x, y)` tegner kurven, `plt.grid()` legger på rutenett, og `plt.show()` viser figuren. Grafen er en parabel med nullpunkter i $x = 1$ og $x = 3$ og bunnpunkt i $(2, -1)$. ∎

A.31 Listen er `data = [2, 4, 6, 8]`. Hva er

a) `len(data)`?

b) `data[2]`?

c) `sum(data)`?

A.32 Hva skriver programmet ut?

Python

```
data = [2, 4, 6, 8]
print(sum(data) / len(data))
```

A.33 Hva skriver programmet ut?

Python

```
data = [2, 4, 6, 8]
print(data[-1])
```

A.34 Funksjonen er

Python

```
def f(x):
    return 4 * x + 1
```

Hva er `f(3)` og `f(0)`?

A.35 Hva skriver programmet ut?

Python

```
def f(x):
    return 4 * x + 1

for x in range(3):
    print(f(x))
```

A.36 Hva skriver programmet ut?

Python

```
def areal(l, b):
    return l * b

print(areal(4, 5))
```

A.37 Hva skriver programmet ut?

Python

```
def g(x):
    return x**2

print(g(3) + g(4))
```

A.38 Hva skriver programmet ut?

Python

```
x = []
x.append(5)
x.append(7)
print(sum(x))
```

A.39 Hva skriver programmet ut?

Python

```
temperaturer = [3, -1, -4, 2]
for t in temperaturer:
    if t < 0:
        print(t)
```

A.40 Hva skriver programmet ut? Hva er galt med funksjonen?

Python

```
def g(x):
    y = 2 * x

print(g(2))
```

A.41 Hva skriver programmet ut?

Python

```
def saldo(K, p, n):
    return K * (1 + p / 100) ** n

print(saldo(2000, 5, 2))
```

A.42 Koden tegner $y = x^2 - 1$. Hva er y når $x = 3$?

Python

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-2, 4, 100)
y = x**2 - 1
plt.plot(x, y)
plt.show()
```

A.43 Hva skriver programmet ut?

Python

```
def f(x):
    return -x**2 + 6 * x

print(f(3))
```

A.44 Forklar hvorfor det er nyttig å bruke en funksjon i stedet for å skrive utregningen om igjen.

A.45 Hva skriver programmet ut?

Python

```
def f(x):
    return x**2 - 2*x

def vekstfart(a, b):
    return (f(b) - f(a)) / (b - a)

print(vekstfart(1, 5))
```

Fasit

A.1.1 Tall og variabler

A.1
a) 11
b) 21
c) 16
d) 8

A.2
a) `4`
b) `1`
c) `4.5`

A.3
```
2.5
3.0
```

A.4
`7`

A.5
`9 10`

A.6
`360.0`, fordi $400 \cdot 0{,}9 = 360$.

A.7
`0.67`

A.8
`0.5`, fordi $2^{-1} = \frac{1}{2}$.

A.9
Linja `y = 3x`. Gange må skrives ut: `y = 3 * x`.

A.10
$5 \cdot 4 - 3 = 17$

A.11
`2.5`

A.12
`2 3`, fordi 5 går opp i 17 tre ganger, med rest 2.

A.13
`24`, fordi $16 + 8 = 24$.

A.14
`=` gir variabelen en ny verdi. Det betyr ikke «er lik». Linja `x = x + 1` gjør at `x` får verdien «den gamle `x` pluss 1».

A.15
a) $\sqrt{25} = 5$ (`5.0`)
b) $10^{-3} = 0{,}001$
c) $\sqrt[3]{27} = 3$ (`3.0`)

A.16
$d = 36 - 20 = 16$, så $x = \frac{6 \pm 4}{2}$.

Utskriften er `5.0 1.0`.

A.1.2 Vilkår og løkker

A.17
For 10 år: `50`. For 30 år: `100`.

A.18
```
0
1
2
```

A.19
```
2
4
6
```

A.20
```
2
5
8
```

A.21
`10`, fordi $1 + 2 + 3 + 4 = 10$.

A.22
`4`. Tallene er 5, 10, 15, 20.

A.23
`27`. `x` blir $1, 3, 9, 27$, og $27 \geqslant 20$ stopper løkka.

A.24
`3`. Beløpet blir 120, 144 og 172,8. Etter 3 år er det over 150.

A.25
`x` endres ikke i løkka, så vilkåret er alltid sant. Løsningen er å legge til `x = x + 1` i løkka.

A.26
```
0 -1
1 2
2 5
3 8
```

A.27
`=` gir en variabel en verdi. `==` tester om to ting er like, og gir sant eller usant.

A.28
`5`. De like tallene er 2, 4, 6, 8, 10.

A.29
1, 2, 3, 4. Stopp-tallet er ikke med, så 5 er ikke med.

A.30
$d = 1 - 24 = -23 < 0$, så utskriften er `ingen løsning`.

A.1.3 Lister, funksjoner og grafer

A.31
a) `4`
b) `6`, fordi plassene telles fra 0.
c) `20`

A.32
`5.0`

A.33
`8`, siden `-1` er det siste elementet.

A.34
`f(3)` er 13, og `f(0)` er 1.

A.35
```
1
5
9
```

A.36
`20`

A.37
`25`, fordi $9 + 16 = 25$.

A.38
`12`

A.39
```
-1
-4
```

A.40
`None`. Funksjonen mangler `return`, så den gir ikke noe tilbake.

A.41
`2205.0`, fordi $2\,000 \cdot 1{,}05^2 = 2\,205$.

A.42
$3^2 - 1 = 8$

A.43
`9`. Det er toppunktet til parabelen.

A.44
Utregningen skrives én gang og kan brukes med nye tall. Det gir færre feil, og koden blir lettere å lese og rette.

A.45
$f(1) = -1$ og $f(5) = 15$, så $\frac{15 - (-1)}{4} = 4$. Utskriften er `4.0`.