

Repetisjon/oppgaver i BED-1304 Python-Lab

Markus J. Aase

Forelesning 2 - Python Basics

Informasjon

Dette dokumentet er ment for å repetere og teste din forståelse av kjernepensum i **BED-1304 Python-Lab**. Oppgavene her vil minne litt om hva dere *potensielt* kan få på eksamen, og dekker tematikk knyttet til **Forelesning 2 - Python Basics**.

Det vil være tre ulike oppgavetyper, og litt repetisjon av fagstoffet som er gjennomgått i forelesning 1-2.

- **Flervalgsoppgaver:** Ett riktig svar per oppgave.
- **Kortsvarsoppgaver:** Tester kodeforståelse, pseudokode og feilsøking.
- **Praktiske programmeringsoppgaver:** Hvor du skal skrive kode for å løse et problem.

Lykke til!

Legg merke til at dette ikke nødvendigvis dekker **hele** pensum. Dere står selv ansvarlig for å kunne det nødvendige fra pensum (altså forelesning/seminar/videoene).

Først, litt enkel repetisjon

Hva er et programmeringsspråk?

Et programmeringsspråk er et formelt språk som brukes til å gi instruksjoner til en datamaskin. Språket gir oss muligheten til å beskrive algoritmer og løse problemer på en strukturert måte.

Eksempel: Høynivåspråk (Python)

Python er et høynivåspråk. Koden er enkel å lese og forstå, og minner mye om menneskelig språk. For eksempel kan vi skrive et lite program som regner ut summen av tallene fra 1 til 5:

```
1 total = 0
2 for i in range(1, 6):
3     total += i
4
5 print(f"Summen av tallene fra 1 til 5 er {total}")
```

Eksempel: Lavnivåspråk (Assembly)

Assembly er et lavnivåspråk, som ligger mye nærmere maskinkoden. Her gjør vi det samme som i Python-eksemplet (summen av tallene 1 til 5), men koden er mye mer detaljert og vanskeligere å lese:

```
1     MOV CX, 5          ; Teller (antall tall)
2     MOV AX, 0          ; Akkumulator = 0
3     MOV BX, 1          ; Startverdi = 1
4
5 LOOP_START:
6     ADD AX, BX         ; Legg til BX i AX
7     INC BX             ; k BX med 1
8     LOOP LOOP_START
9
10    ; Resultatet (summen) ligger n i AX
```

Hva er Python?

Python er et av verdens mest brukte programmeringsspråk. Det er kjent for å være lett å lese og lære, samtidig som det er kraftig nok til å brukes i avanserte prosjekter innen dataanalyse, maskinlæring, webutvikling, automatisering og mye mer.

Gode programmeringsvaner

- **Pen kode er bedre enn stygg kode.** Bruk innrykk, mellomrom og linjeskift for å gjøre koden ryddig.
- **Kommenter koden.** Skriv korte forklaringer slik at både du og andre forstår hensikten.
- **Gi gode navn til variabler.** Bruk *alder* i stedet for *x* dersom variabelen faktisk representerer en alder.
- **Unngå duplisering.** Ikke skriv samme kode flere ganger, lag heller funksjoner.
- **Test koden ofte.** Kjør korte tester underveis for å fange opp feil tidlig.

Variabeltyper

- **int** (heltall): `x = 5`
- **float** (desimaltall): `y = 3.14`
- **bool** (sann/usann): `z = True`
- **string** (tekst): `navn = "Anna"`

Utskrift med `print()`

Vi kan skrive ut data med funksjonen `print()`:

```
1 print("Hei verden")
```

Vi kan også bruke f-strenger for å formatere tekst:

```
1 navn = "Anna"
2 alder = 22
3 print(f"Hei, jeg heter {navn} og jeg er {alder} aar gammel.")
```

Vi har nå diskutert ulike variabel-typer, og programmeringsspråk.
La oss se på noen regler ved operatorene.

Operatører i Python

I Python finnes det mange ulike operatører. Disse brukes til å utføre beregninger, sammenligninger og tilordninger. Her følger en oversikt over de vanligste:

• Aritmetiske operatører:

- `+` : addisjon (f.eks. `2 + 3 = 5`)
- `-` : subtraksjon (f.eks. `5 - 2 = 3`)
- `*` : multiplikasjon (f.eks. `4 * 3 = 12`)
- `/` : divisjon (flyttall, f.eks. `7 / 2 = 3.5`)
- `//` : heltallsdivisjon (avrunder ned, f.eks. `7 // 2 = 3`)
- `%` : modulus (resten etter divisjon, f.eks. `7 % 2 = 1`)
- `**` : eksponent (opphøyd i, f.eks. `2 ** 3 = 8`)

• Tilordningsoperatører:

- `=` : tilordning (lagrer en verdi i en variabel, f.eks. `x = 5`)
 - `+=`, `-=`, `*=`, `/=` : kombinerer operasjon og tilordning
- Eksempel:

```
1 x = 5
2 x += 3 # tilsvarer x = x + 3
3 print(x) # Output: 8
4
```

• Sammenligningsoperatører:

- `==` : er lik (f.eks. `5 == 5` gir `True`)
- `!=` : ikke lik (f.eks. `5 != 3` gir `True`)
- `>` : større enn
- `<` : mindre enn

- `>=` : større enn eller lik
- `<=` : mindre enn eller lik

Eksempel som viser flere operatorer i bruk:

```

1 a = 7
2 b = 2
3
4 print(a + b)    # 9
5 print(a / b)    # 3.5
6 print(a // b)   # 3
7 print(a % b)    # 1
8 print(a ** b)   # 49
9
10 print(a == b)  # False
11 print(a > b)   # True

```

Eksempel ved bruk av `//` og `%` i Python

- `//` er **heltallsdivisjon**. Det gir resultatet av divisjonen, men avrundet ned til nærmeste heltall.

```

1 print(7 // 3)  # Output: 2    (fordi 7 / 3 = 2.333..., avrundet ned til
2                2)

```

- `%` er **modulus-operatoren**, som gir resten etter heltallsdivisjon.

```

1 print(7 % 3)   # Output: 1    (fordi 7 = 3*2 + 1)
2

```

Disse to brukes ofte sammen, f.eks. når man skal finne ut om et tall er partall eller oddetall:

```

1 tall = 10
2 if tall % 2 == 0:
3     print("Partall")
4 else:
5     print("Oddetall")

```

Del 1: Flervalgsoppgaver

Velg **ett riktig svar** per oppgave.

1. Hva brukes Python hovedsakelig til?
 - a) Matlaging
 - b) Dataanalyse, programmering og automatisering
 - c) Tegning
 - d) Ingen av de over
2. Hva er output fra `print(2+3*4)`?
 - a) 20
 - b) 14
 - c) 24
 - d) 18
3. Hva blir resultatet av `5 ** 2`?
 - a) 7
 - b) 10
 - c) 25
 - d) 52
4. Hvilken datatype har `3.14`?
 - a) int
 - b) float
 - c) bool
 - d) string
5. Hvilken operator brukes for divisjon?
 - a) *
 - b) +
 - c) -
 - d) /
6. Hva er output fra `print(True and False)`?
 - a) True
 - b) False
 - c) 1
 - d) 0
7. Hva er riktig måte å lage en streng på?
 - a) `navn = '01a'`
 - b) `navn = "01a"`
 - c) Begge

- d) Ingen av de
8. Hva er resultatet av `10 // 3`?
- a) 3.33
 - b) 3
 - c) 4
 - d) 30
9. Hva gjør operatoren `%`?
- a) Potens
 - b) Rest ved divisjon
 - c) Multiplikasjon
 - d) Negasjon
10. Hva er resultatet av `2 * (3+4)`?
- a) 14
 - b) 10
 - c) 12
 - d) 20
11. Hva vil `print(type(True))` gi?
- a) int
 - b) bool
 - c) string
 - d) float
12. Hva skjer om du skriver `navn = 10; navn = navn + 5`?
- a) Feil
 - b) navn blir 15
 - c) navn blir "105"
 - d) navn blir 5
13. Hva blir output fra `print(2**3)`?
- a) 6
 - b) 8
 - c) 9
 - d) 12
14. Hvordan skriver man en kommentar i Python?
- a) `//` kommentar
 - b) `/*` kommentar `*/`
 - c) `#` kommentar
 - d) `-` kommentar

15. Hva er riktig utskrift, gitt at vi har definert `navn = "Markus"`?

- a) `print(f"Hei {navn}")`
- b) `print("Hei navn")`
- c) Begge virker likt
- d) Ingen virker

Del 2: Kortsvarsoppgaver

1. Hva er output?

```
1 x = 10
2 y = 3
3 print(x % y)
4
```

2. Hva blir skrevet ut?

```
1 a = 2
2 b = 5
3 print(a * b + 3)
4
```

3. Hva vil dette gi?

```
1 print("2" + "3")
2
```

4. Hva vil dette gi?

```
1 print("2" * 3)
2
```

5. Hva er output?

```
1 print(10 // 3)
2
```

6. Hva er output?

```
1 print(3 > 5)
2
```

7. Hva er output?

```
1 navn = "Ola"
2 alder = 20
3 print(f"{navn} er {alder} år")
4
```

8. Hva skrives ut?

```
1 x = 2
2 y = x
3 x = 5
4 print(y)
5
```

9. Skriv pseudokode for et program som:

- Spør brukeren om navn
- Spør om alder
- Skriver ut: "Hei NAVN, du er ALDER år gammel."

10. Skriv pseudokode for et program som regner ut arealet av et rektangel.

Del 3: Praktiske oppgaver

1. Lag et program som ber brukeren skrive inn to tall, og deretter skriver ut summen.
2. Lag et program som beregner arealet av en sirkel gitt radius.
3. Lag et program som spør brukeren om et tall, og skriver ut kvadratet av tallet.
4. Lag et program som konverterer temperatur fra Celsius til Fahrenheit.
5. Lag et program som spør etter et navn, og skriver ut en personlig hilsen.

Fasit

Del 1: Flervalgsoppgaver

1: b, 2: b, 3: c, 4: b, 5: d, 6: b, 7: c, 8: b, 9: b, 10: a, 11: b, 12: b, 13: b, 14: c, 15: a

Del 2: Kortsvar

1: 1, 2: 13, 3: "23", 4: "222", 5: 3, 6: False, 7: "Ola er 20 år", 8: 2

Oppgave 9:

Pseudokode:

START

 INPUT navn

 INPUT alder

 PRINT "Hei " + navn + ", du er " + alder + " år gammel."

STOP

Mulig Python-kode:

```
1 navn = input("Hva heter du? ")
2 alder = input("Hvor gammel er du? ")
3 print(f"Hei {navn}, du er {alder} år gammel.")
```

Oppgave 9:

Pseudokode:

START

 INPUT lengde

 INPUT bredde

 area = lengde * bredde

 PRINT "Arealet er " + area

STOP

Mulig Python-kode:

```
1 lengde = float(input("Oppgi lengden: "))
2 bredde = float(input("Oppgi bredden: "))
3 areal = lengde * bredde
4 print(f"Arealet er {areal}")
```

Del 3: Praktiske oppgaver

Her skal dere kode selv, og under er løsningsforslag. Husk, at det er mange måter å løse ulike problem. Har dere noe annet en det som står her, betyr ikke at dere nødvendigvis har gjort noe galt!

Som vi snakket om i Forelesning 1, og Polyas heurstikker, alltid - **se over, og se om du tror på svaret du har fått!**

Oppgave 1

```
1 # Brukeren skriver inn to heltall, a og b
2 a = int(input("Skriv et tall: "))
3 b = int(input("Skriv et tall til: "))
4
5 print("Summen er", a+b) # en print-setning, og summen
```

Oppgave 2

```
1 # importerer numpy som np, slikat vi kan bruke numpy-biblioteket - for tallet
   pi
2 import numpy as np
3
4 # radius tilordnes en verdi
5 r = 4
6 areal = np.pi * r**2
7
8 print("Arealet er", areal) # en print-setning, og arealet
9
10 # Alternativt, kan vi bruke en f-streng
11 print(f'Arealet er {np.pi * r**2}.')
```

Her vil løsningen gi et svar med flere desimaler, vurder å bruke funksjonen `round(areal)`, 2 - hva gjør `round()`? Hva gjør tallet "2"?

Oppgave 3

```
1 # Ber brukeren om et tall, og vi sikrer
2 # at python tolker det som en "float".
3 tall = float(input("Skriv inn et tall: "))
4
5 print(f'Kvadratet av {tall} er {tall**2}!')
```

Oppgave 4 - Løsningsforslag

```
1 # Program som konverterer temperatur fra Celsius til Fahrenheit
2
3 celsius = float(input("Skriv inn temperatur i Celsius: "))
4
5 fahrenheit = (celsius * 9/5) + 32
6
7 print(f"{celsius} grader Celsius tilsvare {fahrenheit} grader Fahrenheit.")
```

Oppgave 5 - Løsningsforslag

```
1 # Program som tar inn om et navn og skriver ut en personlig hilsen
2
3 navn = input("Hva heter du? ")
4
5 print(f"Heisann {navn}, hyggelig! Vi kan bli venner:>")
```