

Repetisjon/oppgaver i BED-1304 Python-Lab

Markus J. Aase

Forelesning 3 - Funksjoner

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 3 - Funksjoner**.

Det vil være tre ulike oppgavetyper, og litt repetisjon av fagstoffet som er gjennomgått i forelesning 3.

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

Først, litt enkel repetisjon

Hva er en funksjon?

En funksjon er en gjenbrukbar blokk med kode som utfører en bestemt oppgave. Funksjoner gjør koden mer ryddig og modulær, og lar deg unngå duplisering.

Innebygde funksjoner i Python

Python har mange innebygde funksjoner, blant annet:

- `print()` – skriver ut tekst eller variabler.
- `len()` – returnerer lengden av en sekvens (f.eks. liste, string).
- `max()` – finner største verdi.
- `min()` – finner minste verdi.
- `sum()` – summerer verdier i en sekvens.
- `type()` – viser datatypen til en variabel.

Hvordan definere en funksjon

For å definere en funksjon bruker vi `def`-nøkkelordet:

```
1 def funksjonsnavn(param1, param2):  
2     # innrykk er viktig!  
3     resultat = param1 + param2  
4     return resultat
```

Viktig: Alt som hører til funksjonen må ha innrykk (typisk 4 mellomrom). Funksjonen kan returnere en verdi med `return` eller gjøre en handling uten returverdi.

Parametere og argumenter

- **Parametere** er variablene som defineres i funksjonen (`param1, param2`).
- **Argumenter** er verdiene du sender inn når du kaller funksjonen (`funksjonsnavn(3,5)`).

Eksempel: funksjon med returverdi

```
1 def kvadrat(x):  
2     return x**2  
3  
4 print(kvadrat(5)) # Output: 25
```

Eksempel: funksjon som skriver ut noe

```
1 def hils(navn):  
2     print(f"Heisann {navn}!")  
3  
4 hils("Ola") # Output: Heisann Ola!
```

Funksjoner med flere parametere

```
1 def summer(a, b, c):  
2     return a + b + c  
3  
4 print(summer(2,3,4)) # Output: 9
```

Del 1: Flervalgsoppgaver

1. Hva brukes en funksjon til?
 - a) Gjenbrukbar kodeblokk
 - b) Skrive kommentarer
 - c) Endre variabeltyper
 - d) Tegning
2. Hvordan definerer man en funksjon i Python?
 - a) `function navn():`
 - b) `def navn():`
 - c) `fun navn():`
 - d) `define navn():`
3. Hva er en parameter?
 - a) Verdien som sendes inn
 - b) Variabelen som defineres i funksjonen
 - c) Navnet på funksjonen
 - d) En kommentar
4. Hva er et argument?
 - a) Verdien som sendes inn når funksjonen kalles
 - b) Variabelen som defineres i funksjonen
 - c) Navnet på funksjonen
 - d) Returnverdien
5. Hvilken funksjon gir lengden på en liste?
 - a) `size()`
 - b) `len()`
 - c) `length()`
 - d) `count()`
6. Hva gjør `return` i en funksjon?
 - a) Stopper funksjonen og returnerer en verdi
 - b) Printer en verdi
 - c) Definerer en parameter
 - d) Endrer en variabel
7. Hva vil `print(max(3,5,2))` gi?
 - a) 2
 - b) 3
 - c) 5
 - d) Feil

8. Hva vil `print(min([4,7,1]))` gi?
- a) 4
 - b) 1
 - c) 7
 - d) [4, 7, 1]
9. Hva gjør `sum([1,2,3])`?
- a) 6
 - b) 123
 - c) [1,2,3]
 - d) Feil
10. Hva skjer hvis du glemmer innrykk i en funksjon?
- a) Ingenting
 - b) Python gir en `IndentationError`
 - c) Funksjonen kjører som vanlig
 - d) Kommentarer ignoreres
11. Hva skjer hvis du glemmer å skrive `return` i en funksjon som skal gi tilbake en verdi?
- a) Funksjonen returnerer automatisk 0
 - b) Funksjonen returnerer `None`
 - c) Funksjonen krasjer
 - d) Python antar `return 1`
12. Hva er forskjellen på en parameter og et argument?
- a) Parametere er verdier, argumenter er variabler
 - b) Parametere defineres i funksjonen, argumenter sendes inn når den kalles
 - c) Ingen forskjell, de er synonymer
 - d) Parametere er alltid tall, argumenter alltid tekst
13. Hva gjør `len(Python")`?
- a) Returnerer antall bokstaver i ordet
 - b) Printer ordet baklengs
 - c) Gir en feilmelding
 - d) Multipliserer antall bokstaver med 2
14. Hva er en fordel med å bruke funksjoner i programmering?
- a) Man slipper å bruke variabler
 - b) Koden blir mer ryddig og gjenbrukbar
 - c) Python kjører raskere
 - d) Det er eneste måten å bruke `if`-setninger på

15. Hva vil denne koden skrive ut?

```
1  def test(x,y):  
2      return x + y  
3  print(test(3,7))  
4
```

- a) 10
- b) 37
- c) x+y
- d) Feilmelding

Del 2: Kortsvarsoppgaver

1. Hva blir output?

```
1 def kvadrat(x):  
2     return x**2  
3 print(kvadrat(4))  
4
```

2. Hva vil dette skrive ut?

```
1 def hils(navn):  
2     print(f"Hei {navn}!")  
3 hils("Anna")  
4
```

3. Hva blir resultatet?

```
1 def summer(a, b, c):  
2     return a+b+c  
3 print(summer(1,2,3))  
4
```

4. Skriv pseudokode for et program som lager en funksjon som regner kvadratet av et tall.
5. Skriv pseudokode for en funksjon som tar inn to tall og returnerer den største.

Del 3: Praktiske oppgaver

Når dere løser disse oppgavene, kan det være lurt å starte med å skrive en pseudokode med penn og papir, selv om du tror du vet hvordan du vil løse det. Det er en god øvelse!

1. Lag en funksjon `hils()` som tar et navn som parameter og printer en personlig hilsen.
2. Lag en funksjon `areal_rektangel()` som tar `lengde` og `bredde` som parametere, og returnerer arealet.
3. Lag en funksjon `faktorielle()` som tar et tall og returnerer faktorialen. **NB:** Utfordring!
4. Bruk innebygde funksjoner `max()`, `min()` og `sum()` på en liste med tall.
5. Lag en funksjon som tar to tall som parametere og returnerer det største.

Fasit

Del 1: Flervalgsoppgaver

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

Del 2: Kortsvar

1: 16, 2: Hei Anna!, 3: 6

Oppgave 4 - Pseudokode:

```
START
    FUNKSJON kvadrat(x)
        RETURN x * x
STOP
```

Oppgave 5 - Pseudokode:

```
START
    FUNKSJON maks(a,b)
        HVIS a > b
            RETURN a
        ELSE
            RETURN b
STOP
```

Del 3: Praktiske oppgaver - Løsningsforslag

Oppgave 1

```
1 def hils(navn):
2     print(f"Heisann {navn}!")
3 hils("Ola")
```

Oppgave 2

```
1 def areal_rektangel(lengde, bredde):
2     return lengde * bredde
3
4 print(areal_rektangel(4,5)) # Output: 20
```

Oppgave 3

```
1 def faktorielle(n):
2     resultat = 1
3     for i in range(1,n+1):
4         resultat *= i
5     return resultat
6
7 print(faktorielle(5)) # Output: 120
```

Oppgave 4

```
1 tall_liste = [3,7,1,5]
2 print(max(tall_liste)) # 7
3 print(min(tall_liste)) # 1
4 print(sum(tall_liste)) # 16
```

Oppgave 5

```
1 def maks(a,b):
2     if a > b:
3         return a
4     else:
5         return b
6
7 print(maks(5,9)) # Output: 9
```