

Velkommen til repetisjonskurs om prosedyrer, funksjoner, programflyt og skop!

Kurset holdes av Ahmed (ahmedws@uio.no) og Tiera (trierag@uio.no)

Agenda

- Teoretisk gjennomgang
- Interaktive oppgaver
- Oppgaveløsning og liveprogrammering

Prosedyrer

- Bidrar til gjenbruk av kode og strukturere kode
- Kan ta inn opptil flere (og ingen) parametere
- Det som er inne i prosedyren kalles ofte "innmaten" til prosedyren
- Prosedyren utføres ved at vi "kaller" på prosedyren
- Kjennetegnes ved at det ikke er noen returverdier



```
2  
3 def minProsedyre(parameter1, parameter2):  
4     tekst = parameter1 + parameter2  
5     tekst += parameter1  
6     print(tekst)  
7  
8 minProsedyre("argument1", "argument2")
```

Eksempel på prosedyre som viser forskjellen på parameter og argument



Funksjoner

- Funksjoner returnerer verdier (kan også returnere ingenting)
- Mange av de samme formålene som prosedyrer; strukturere kode, legge kode inn i blokker



```
def summerTallene(tall1, tall2):  
    return tall1 + tall2
```

```
summerTallene(100, 30)
```

Eksempel på en funksjon



Oppgave 1

Skriv et program som:

- a. Tar inn to tall som parameter
- b. printer det største tallet

Oppgave 2

Skriv et program som:

- a. Tar inn to tall som parameter
- b. Returnerer det største tallet

Oppgave 3 - Hva er forskjellen oppgave 1 og 2?

returverdi

opg1 er prosedyre opg2 er funksjon

Oppgave 1 = prosedyre Oppgave 2 = funksjon

Oppgave1 var en prosedyre og oppgave2 en funksjon

1 - prosedyre 2 - funksjon

prosedyre og funksjon

oppgave1 er prosedyre, oppgave2 funksjon

oppgave 1 er en prosedyre, mens oppgave 2 er funksjon

Forskjell: Første gir en output, returnerer None Andre: returnerer verdi

Oppgave 3 - Hva er forskjellen oppgave 1 og 2?

Prosedyre, ingen returverdi

funk

prosedyre da den ikke returnerer

Oppgave 3 - Er oppgave 1 en funksjon eller en prosedyre? Hvorfor

Prosedyre, ingen returverdi

prosedyre

1 er prosedyre pga ingen returverdi

prosedyre

prosedyre

prosedyre, returnerer ikke

Funksjon = returna
Prosedyre = ingen retur

prosedyre, returnerer ikke

funksjon

Oppgave 3 - Er oppgave 1 en funksjon eller en prosedyre? Hvorfor

funksjon, returnerer verdi

funksjon da det er returverdi

Oppgave 3 - Er oppgave 2 en funksjon eller en prosedyre? Hvorfor

funksjon, returnerer

funksjon pga returverdi

Funksjon fordi den returnerer en verdi

prosedyre, returnerer tall

Funksjon, har en returverdi

funksjon

Metoder

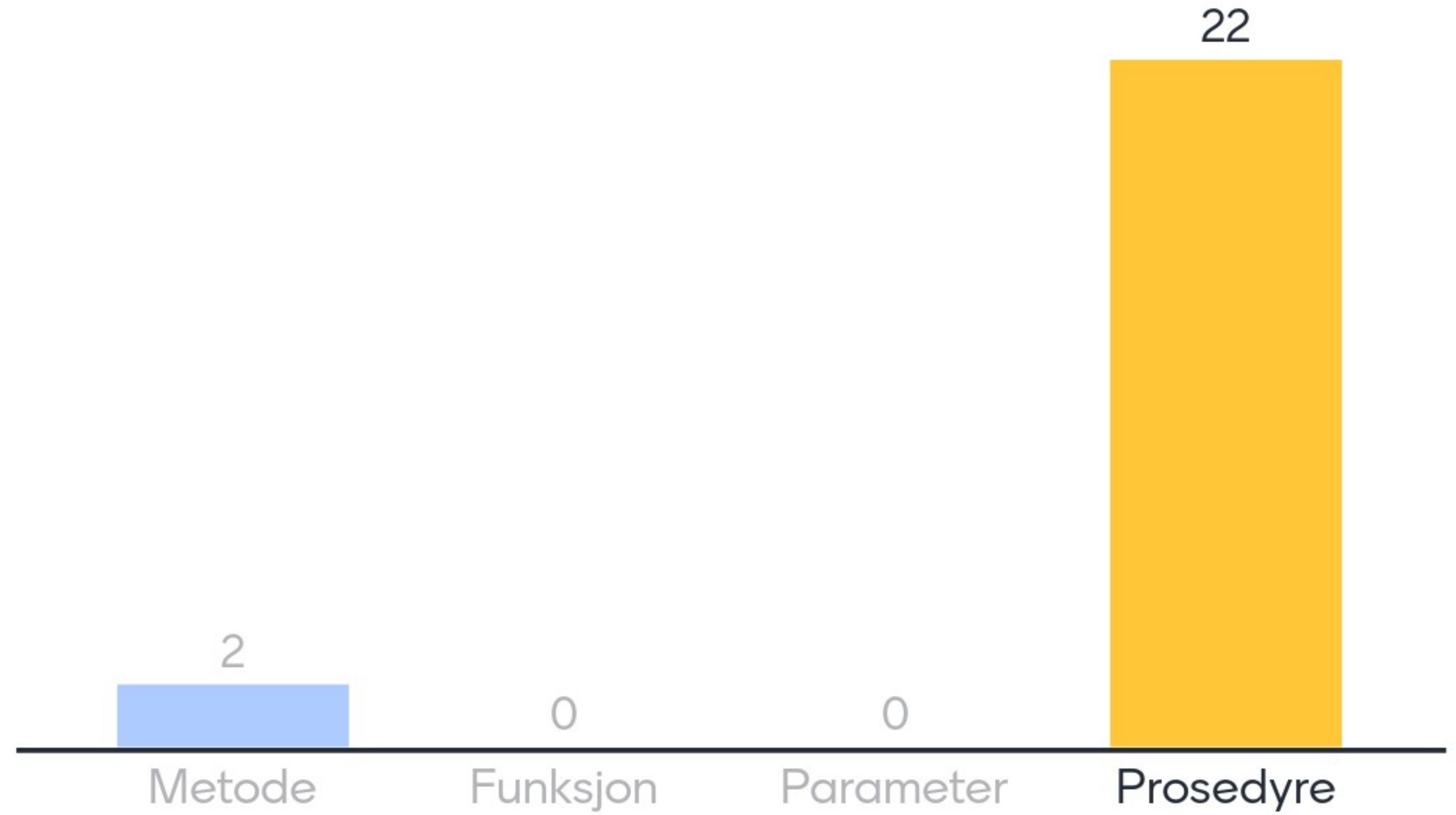
```
class Person:  
    def __init__(self, navn):  
        self._navn = navn  
  
    def skrivNavn(self):  
        print(self._navn)  
  
    def endreNavn(self, nyttNavn):  
        self._navn = nyttNavn
```

Eksisterer i klasser, og kalles på gjennom "dot-notasjon". Eksempel: "hei".lower(). Gitt et Person-objekt: `ola = Person("Ola")`, så kan vi skrive `ola.skrivNavn()` for å kalle på metoden.

Merk at en metode ikke alltid må returnere!

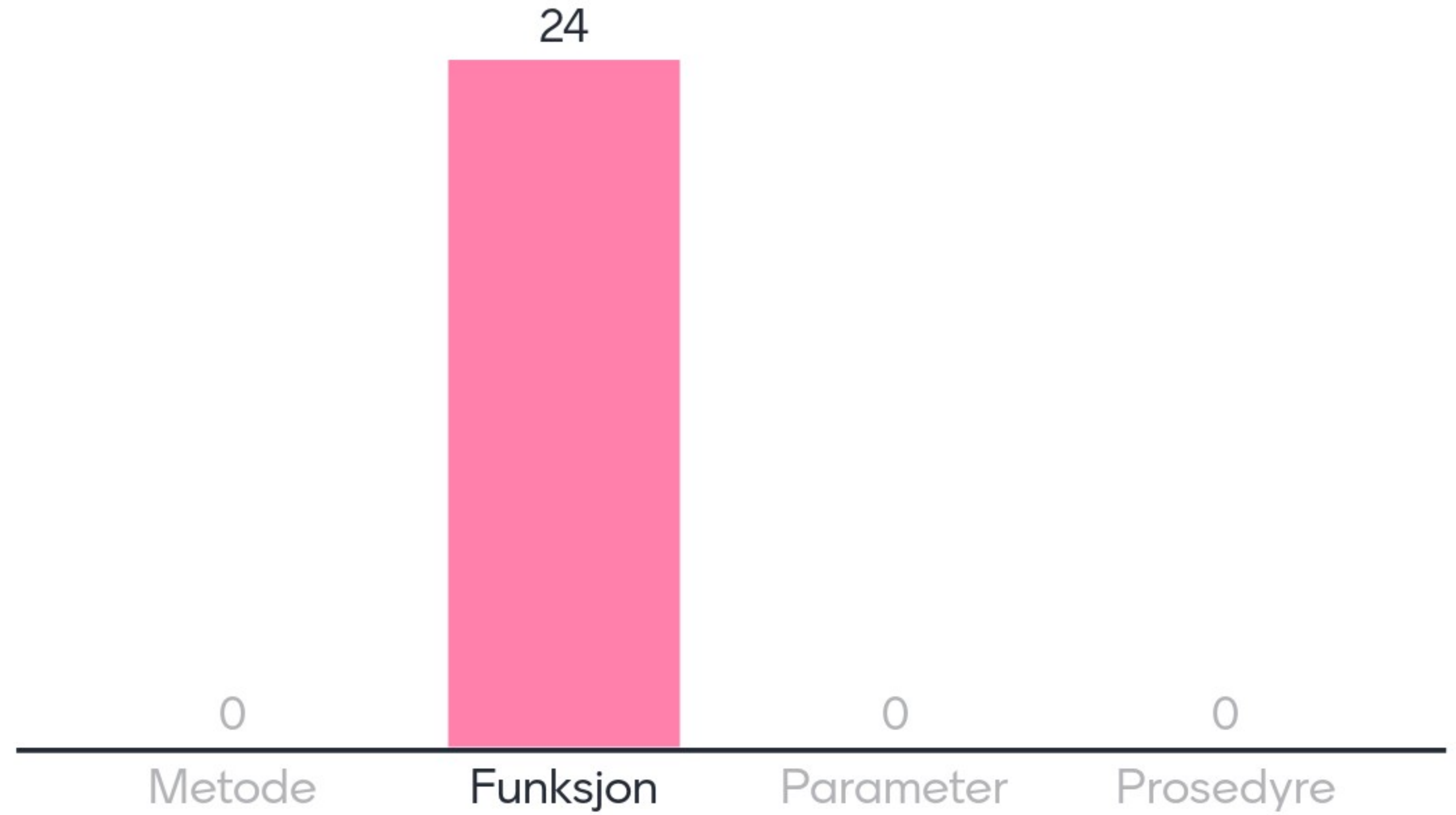
Hva er dette?

```
def skrivHilsen(navn):  
    print("Hei", navn)
```



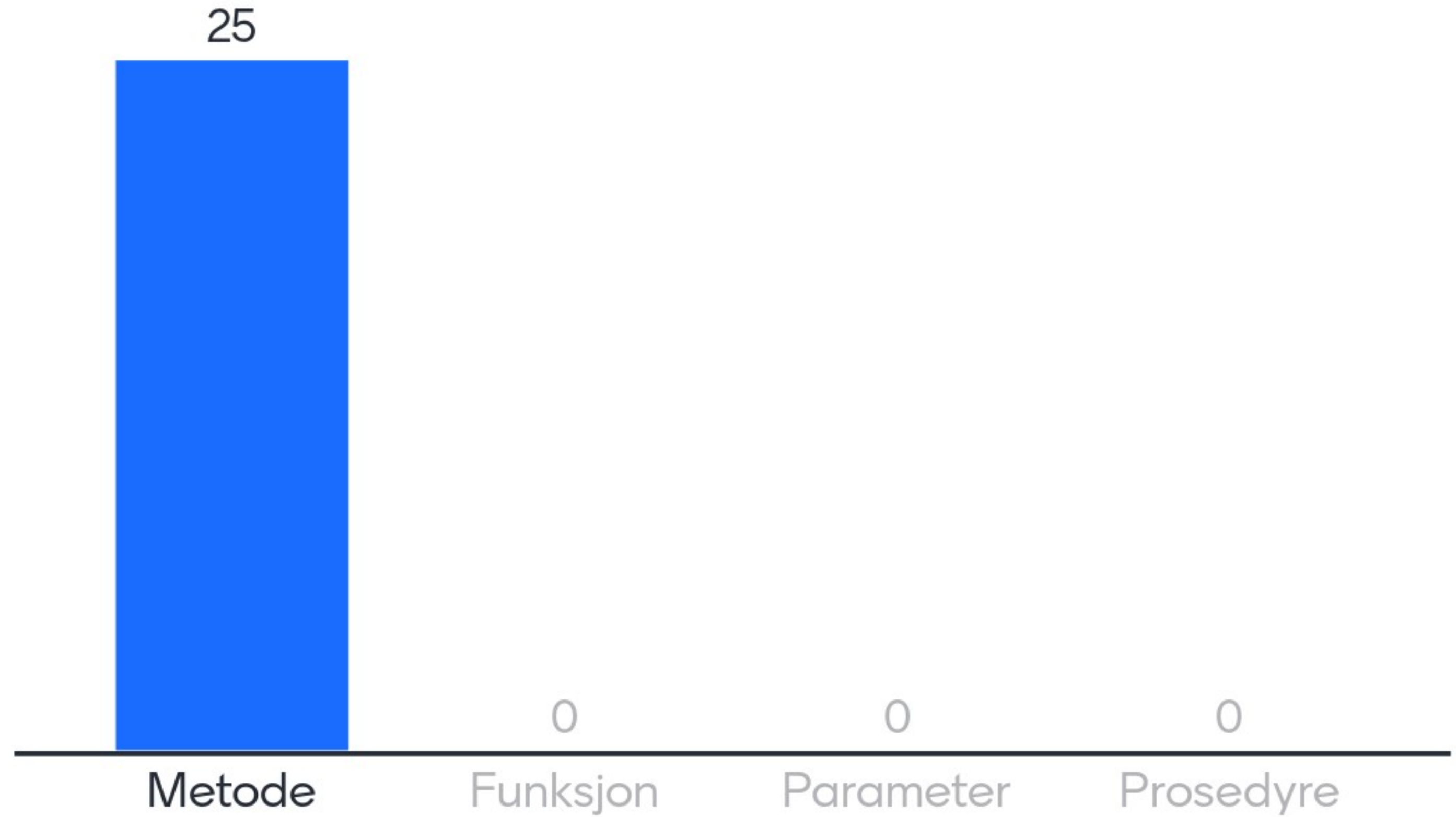
Hva er dette?

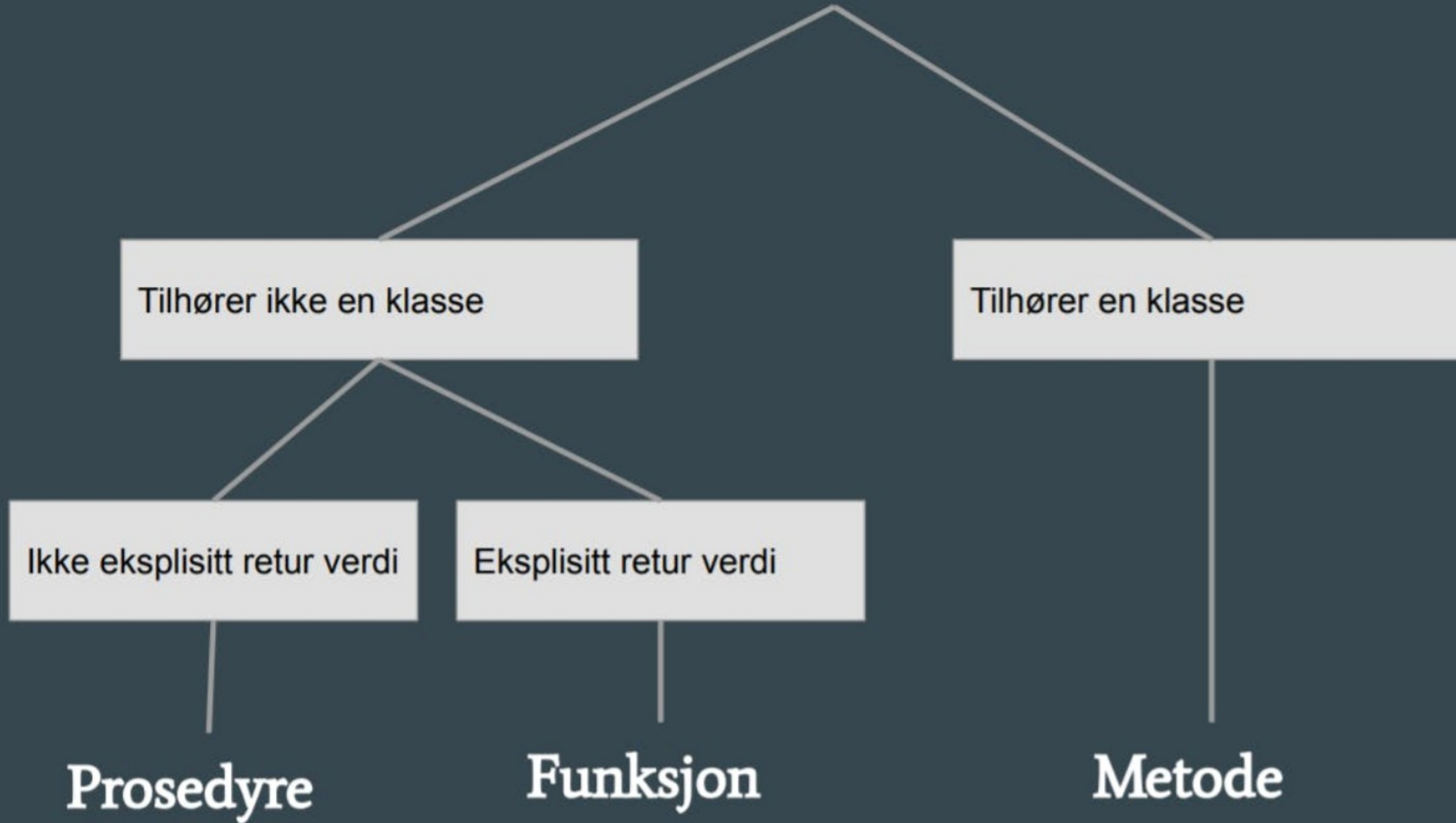
```
def skrivHilsen(navn):  
    return "Hade " + navn
```



Hva er dette?

```
class Fag:  
    def __init__(self, navn):  
        self._navn = navn  
        self._studentliste = []  
  
    def leggTilStudent(self, student):  
        self._studentliste.append(student)
```





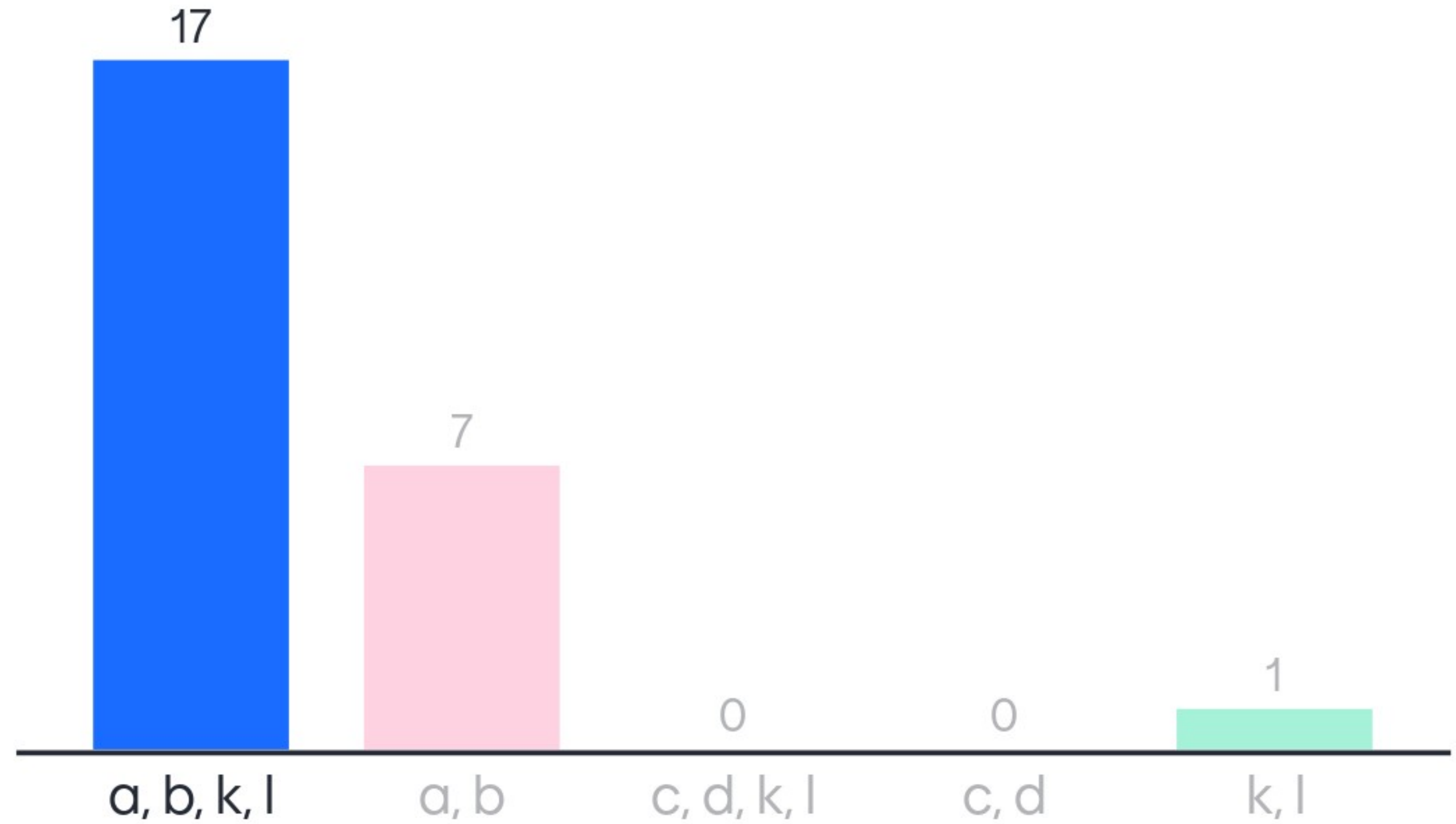
Skop

- Spesifiserer hvilke variabler en har tilgang på
- Det globale skopet
- Variabler inne i funksjoner/prosedyrer har et eget lokalt skop



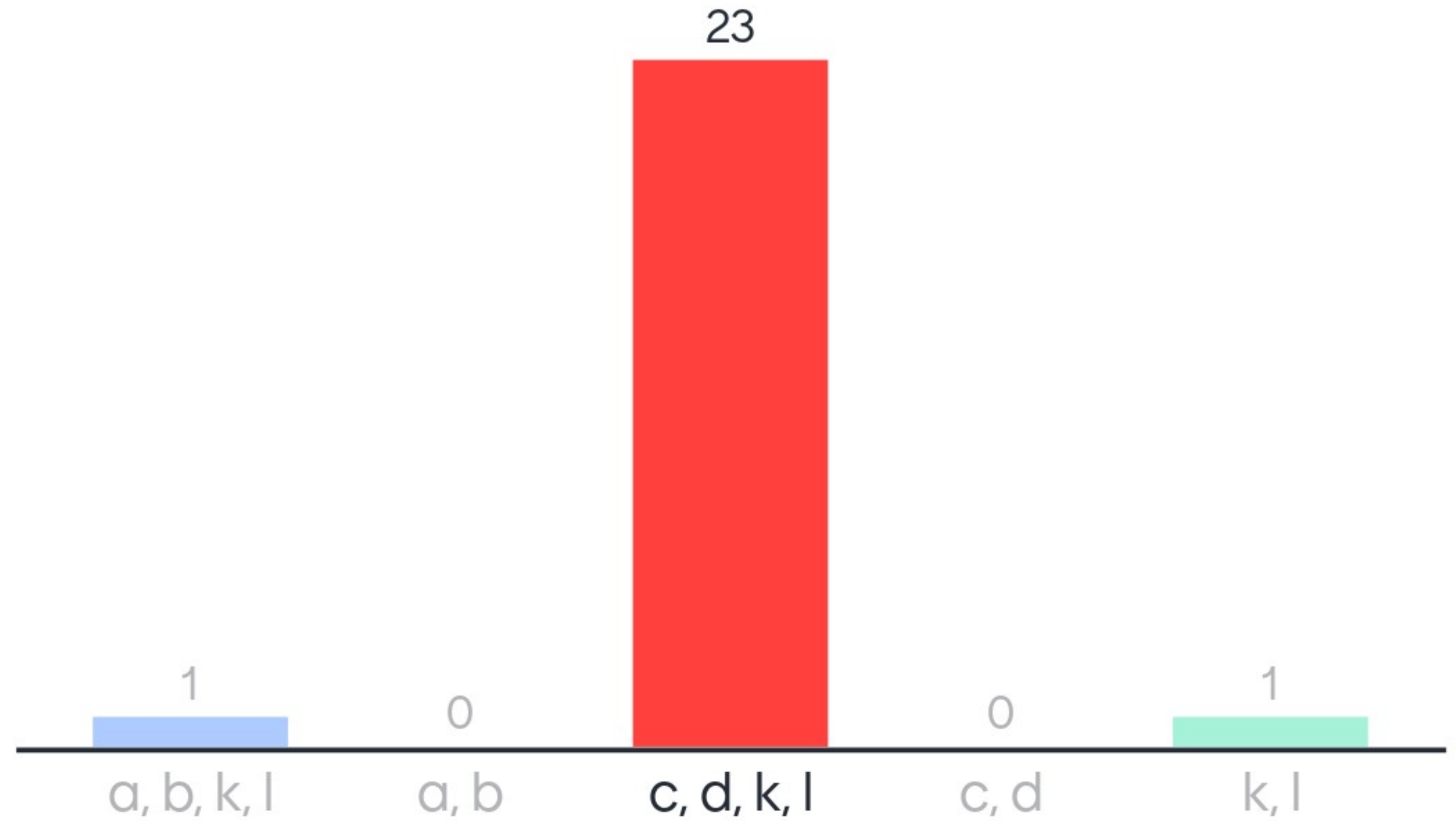
Hvilke variabler kan prosedyre() få tak i?

```
2  
3 def prosedyre():  
4     a = 10  
5     b = 23  
6     print(a + b)  
7  
8 def funksjon():  
9     c = 34  
10    d = 19  
11    return c + d  
12  
13 k = "hei du"  
14 l = 3.9
```



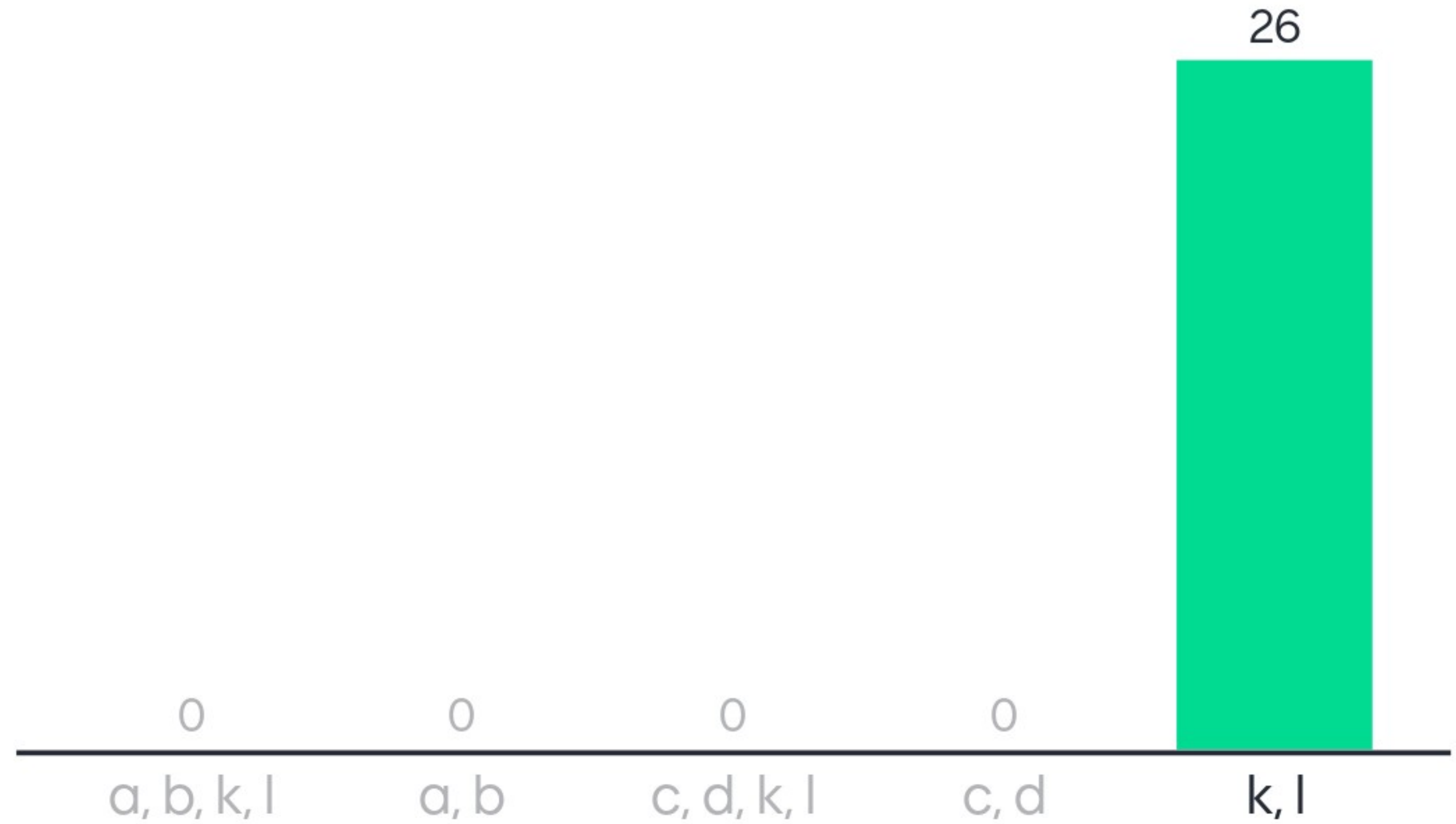
Hvilke variabler kan funksjon() få tak i?

```
2  
3 def prosedyre():  
4     a = 10  
5     b = 23  
6     print(a + b)  
7  
8 def funksjon():  
9     c = 34  
10    d = 19  
11    return c + d  
12  
13 k = "hei du"  
14 l = 3.9
```



Hvilke variabler kan det globale skopet få tak i?

```
2  
3 def prosedyre():  
4     a = 10  
5     b = 23  
6     print(a + b)  
7  
8 def funksjon():  
9     c = 34  
10    d = 19  
11    return c + d  
12  
13 k = "hei du"  
14 l = 3.9
```



Programflyt - rekkefølgen på programutførelsen

- Starter fra toppen og beveger oss nedover
- Leser inn (men kaller ikke) på funksjoner og prosedyrer
- Eksekverer kallene på funksjoner/prosedyrer i det de oppstår
- Dette kan virke abstrakt, og derfor skal vi benytte Pythontutor



Oppgave 4

a. Lag en funksjon som:

- i. Tar inn et tall som parameter
- ii. Finner summen av alle tallene opp til dette tallet og returnerer det
- iii. F.eks. hvis tallet er lik 4 er svaret 10 ($4 + 3 + 2 + 1$)

b. Lag en funksjon som:

- i. Tar inn et tall som parameter
- ii. Finner gjennomsnittet av summen av alle tallene opp til dette tallet
- iii. eks: hvis tallet er lik 4 er svaret 2,5 ($((4 + 3 + 2 + 1)/4)$)

Oppgave 5 - del 1

Du skal i denne oppgaven lage en kalkulator som kan addere, subtrahere, gange, dele, faktorisere og opphøye med instruksjoner som er gitt av brukeren

- a. Lag en funksjon adder som tar inn to tall som parameter og returnerer summen av disse
- b. Lag en funksjon substrater som tar inn to tall som parameter og returnerer det første tallet minus det andre
- c. Lag en funksjon gange som tar inn to tall som parameter og returnerer produktet av disse
- d. Lag en funksjon dele som tar inn to tall som parameter og returnerer det første tallet delt på det andre. Legg også en assert som sjekker om det andre tallet er 0 (for det er ikke lov til å dele noe på 0)
- e. Lag en funksjon faktor som tar inn et tall og finner faktoren av dette tallet.

Faktoren av et tall er tall $tall - 1$ tall - 2 ... 1.



Oppgave 5 - del 2

f. Lag en funksjon opphoy som tar inn to tall som parameter og returnerer det første tallet opphøyd i det andre

g. Lag en prosedyre meny som gjør disse valgene tydelig for brukeren:

- i. Hvis brukeren taster inn AC avsluttes programmet
- ii. Hvis brukeren taster inn + skal to tall legges sammen
- iii. Hvis brukeren taster inn - skal to tall subtraheres
- iv. Hvis brukeren taster inn * skal to tall ganges sammen
- v. Hvis brukeren taster inn / skal et tall deles på et annet
- vi. Hvis brukeren taster inn ! skal du finne faktoren til tallet
- vii. Hvis brukeren taster inn ^ skal du opphøye et tall i et annet

h. Vi skal nå lage et hovedprogram som får alt til å henge sammen



Oppgave 5 - del 3

Eksempel på kjøring (*/*meny*/* representerer at menyen blir printet ut)

```
Hei og velkommen til min kalkulator!!  
  
/*meny*/  
Skriv inn en ønsket kommando: +  
Skriv inn et tall: 10  
Skriv inn et tall som vi skal legge til 10: 8  
Det nye tallet ditt er 18  
/*meny*/  
Skriv inn en ønsket kommando: -  
Skriv inn tallet som vi skal trekke fra 18: 14  
Det nye tallet ditt er 4  
/*meny*/  
Skriv inn en ønsket kommando: !  
Det nye tallet ditt er 24  
/*meny*/  
Skriv inn en ønsket kommando: AC  
  
Takk for at du brukte min kalkulator !
```

h. Vi skal nå lage et hovedprogram som får alt til å henge sammen

i. Start med å printe ut menyen

ii. Spør brukeren om en kommando

iii. Lag en variabel tall (den skal ta vare på tallet underveis). Når denne blir initialisert av input fra brukeren.

iv. Start en while-loop som kjører så lenge kommandoen ikke er AC, og som kaller på de riktige funksjonene etter som hva bruker taster inn (hvis brukeren taster inn + kaller du på adder). Hvis brukeren taster inn alt annet enn ! må du huske å hente inn et tall til fra brukeren. (hvis dette er uklart se på eksempel av kjøring)



Oppgave 6

Du skal skrive et program som skal hjelpe nationalteateret å beregne hvor mye det koster for en gruppe å dra på et stykke. Dette skal du gjøre ved å skrive funksjon `prisForStykke(listeMedAldere)`. Funksjonen tar inn en liste med tall, hvor tallene representerer alderen til personen. Prisen er nemlig forskjellig etter alderen. Hvis barnet er opp til 2 år for den komme inn gratis(en baby), mellom 2-16 er barnebillett (denne koster 60 kr), 16-29 får studentrabatt, rabatten er på 25% av en voksenbillett, en voksenbillett koster 260kr og til slutt har du honnørbillett til 40% av voksenbilletten. Du skal finne summen for helle gruppe og returnere denne. Hvis gruppa er på over 5 personer (her teller vi ikke med babyer, barn opp til 2 år) får gruppe i tillegg 10% rabatt på summen av alle billettene.



Utfordring: I denne oppgaven får du friheten til å bestemme hvordan du skal strukturere koden. Men det er sterkt anbefalt at du benytter deg av bruk av prosedyrer og funksjoner der det er naturlig. Les hele oppgaven før du begynner

Det nærmer seg eksamen og du ønsker å lage et program som gjør det enklere og organisere de ulike oppgavene du skal gjøre før eksamen (f.eks. å løse en trix oppgave eller å løse en tidligere eksamen), også kalt en taks. Det skal være mulig for brukeren å legge til en task (en task inneholde navnet på oppgaven, denne er unik, og hvor lang tid det kommer til å ta gitt i minutter). Hvis brukeren ønsker skal den ha mulighet til å se hvor lang tid vedkommende har igjen på alle taskene sine. Brukeren skal også ha mulighet til jobbe med en task, dette gjøres ved at brukeren oppgir navnet på oppgaven og hvor mye tid vedkommende har brukt. Det skal også være mulig å skrive ut informasjon om alle taksene som man fortsatt må gjøre og alle taskene som er ferdig (en task er ferdig når antall minutter som gjenstår på tasken er 0 minutter eller mindre) .

Oppgave 7



Oppgave 8

Du skal skrive følgende to funksjoner/prosedyrer

- a. Lag en funksjon som tar inn en liste med tall som parameter, og som ganger alle tallene med 6. Returner den listen.
- b. Lag en prosedyre som tar inn en liste med tall som parameter, og som ganger alle tallene med 6.



Oppgave 8 - Er det noe forskjell i hvordan lista er etter at funksjonen har kjørt vs prosedyren?



Hvilke variabler har det globale skopet tilgang til ?

```
def funksjon():  
    a = 23  
    b = 45  
    c = a + b  
    return c  
  
def prosedyre():  
    a = a + 1  
    k = "HEI"  
    print(k , a)  
  
tekst = "Dette er en string"  
tall = 568
```

0	0	0	0
tekst, tall	a, b, c, k, tekst, tall	a, k	a, b, c



Hvilke variabler har funksjon tilgang til?

```
def funksjon():  
    a = 23  
    b = 45  
    c = a + b  
    return c  
  
def prosedyre():  
    a = a + 1  
    k = "HEI"  
    print(k , a)  
  
tekst = "Dette er en string"  
tall = 568
```

0
a, b, c

0
a, b, c, k,
tekst, tall

0
a, b, c, tekst,
tall



Hvilke variabler har prosedyre tilgang til?

```
def funksjon():  
    a = 23  
    b = 45  
    c = a + b  
    return c  
  
def prosedyre():  
    a = a + 1  
    k = "HEI"  
    print(k , a)  
  
tekst = "Dette er en string"  
tall = 568
```

0
a, b, c, tekst,
tall

0
a, k, tekst,
tall

0
a, b, c, k,
tekst, tall



Hvorfor vil koden kræsje når vi kaller på prosedyre?

```
def funksjon():  
    a = 23  
    b = 45  
    c = a + b  
    return c  
  
def prosedyre():  
    a = a + 1  
    k = "HEI"  
    print(k , a)  
  
tekst = "Dette er en string"  
tall = 568
```

