Mentimeter

Uke 5 - problemløsning

Gruppe 14

Kristin - kristlgj@ifi.uio.no

Mentimeter



Mentimeter

Plan for gruppetimen

- Peer-instruction
- Oppgaver/parprogrammering

Hva skrives ut til terminalen?

```
b = 10
a = 2

def subtraher(a, b):
    return a - b


print( subtraher(b, a) )
```

0 ✓	0 ✗	0 ✗	0 ✗
8	-8	0	Feilmelding (NameError)

Mentimeter

Diskuter!

Hva skrives ut til terminalen?

```
b = 10
a = 2

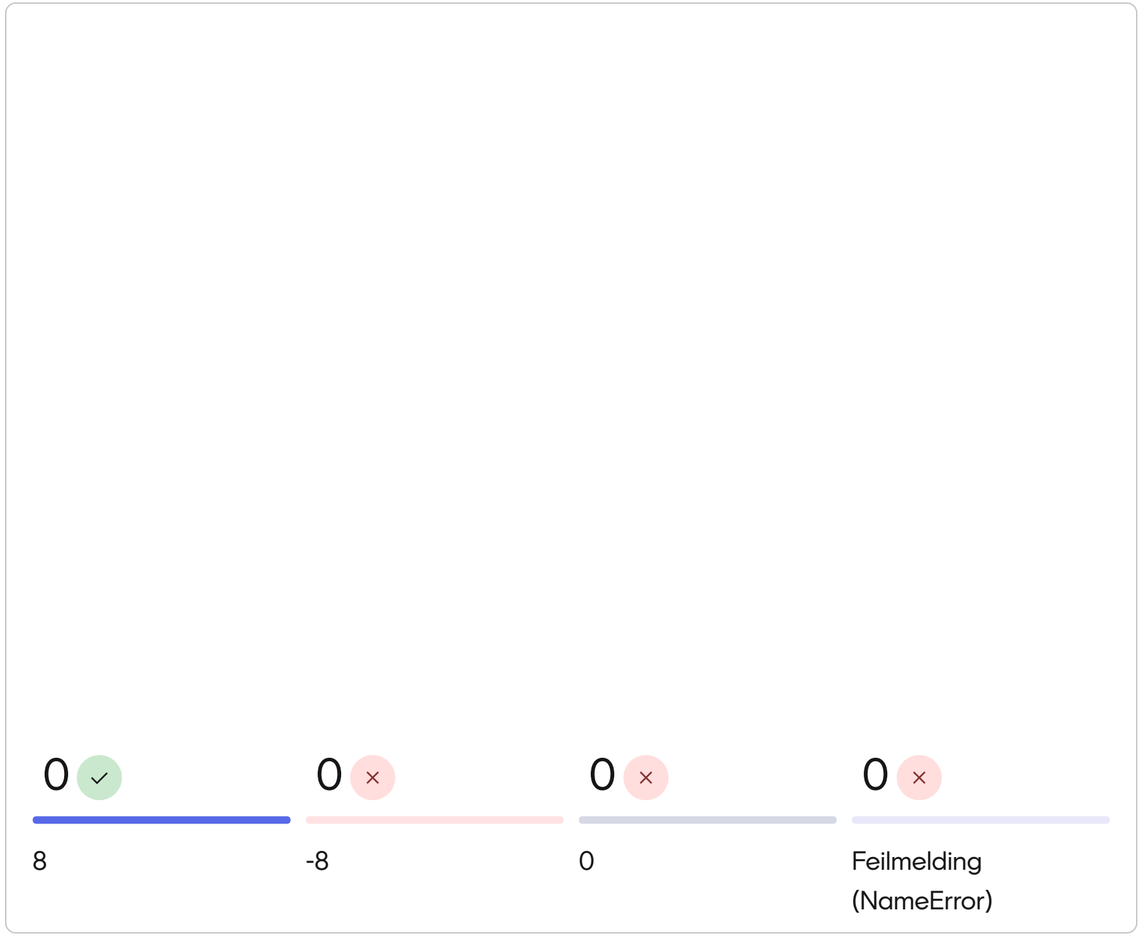
def subtraher(a, b):
    return a - b


print( subtraher(b, a) )
```

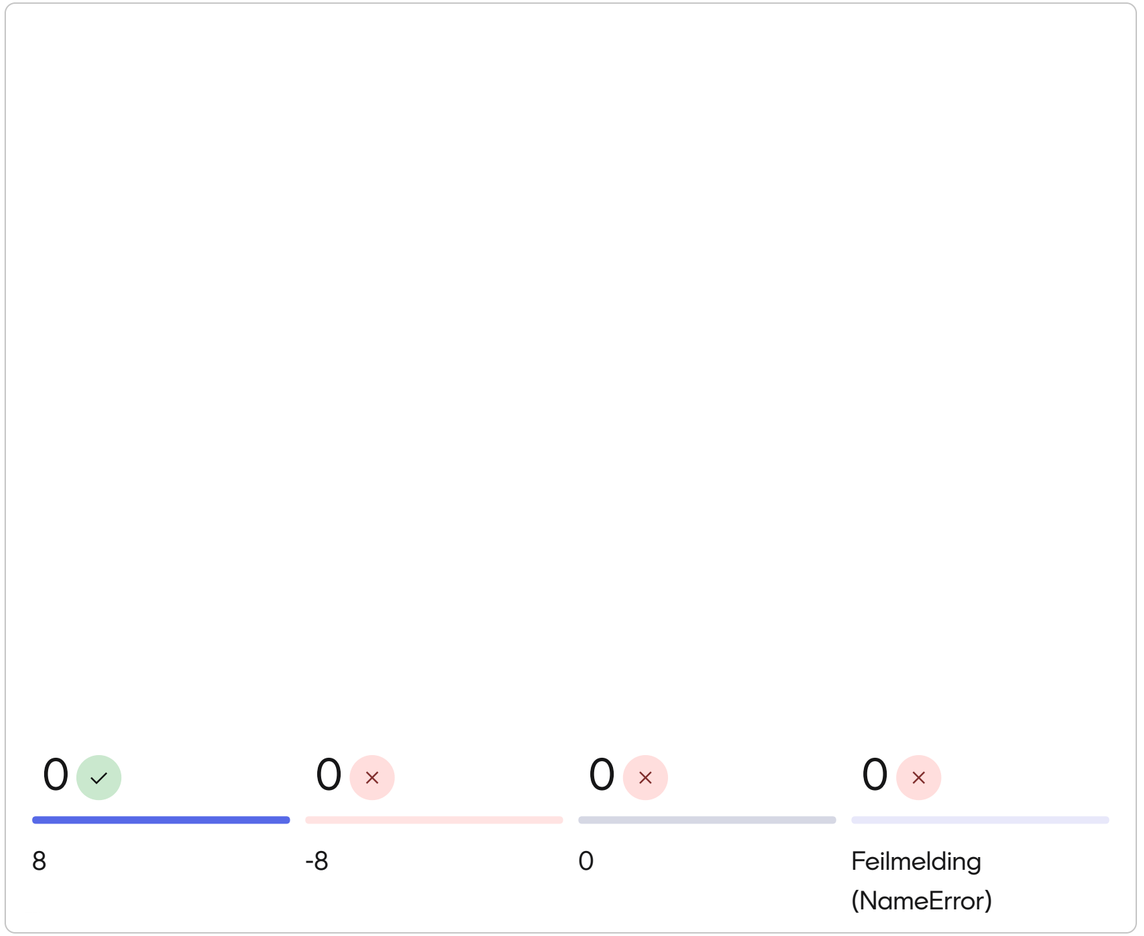
0 ✓	0 ✗	0 ✗	0 ✗
8	-8	0	Feilmelding (NameError)

Compare slides

Hva skrives ut til terminalen?



Hva skrives ut til terminalen?



Hva skrives ut til terminalen?

```
def oppdater(x):
    x = x + 10
    return x

x = 5
oppdater(x)
print(x)
```

0 ✓	0 ×	0 ×	0 ×
5	10	15	None

Mentimeter

Diskuter!

Hva skrives ut til terminalen?

```
def oppdater(x):
    x = x + 10
    return x

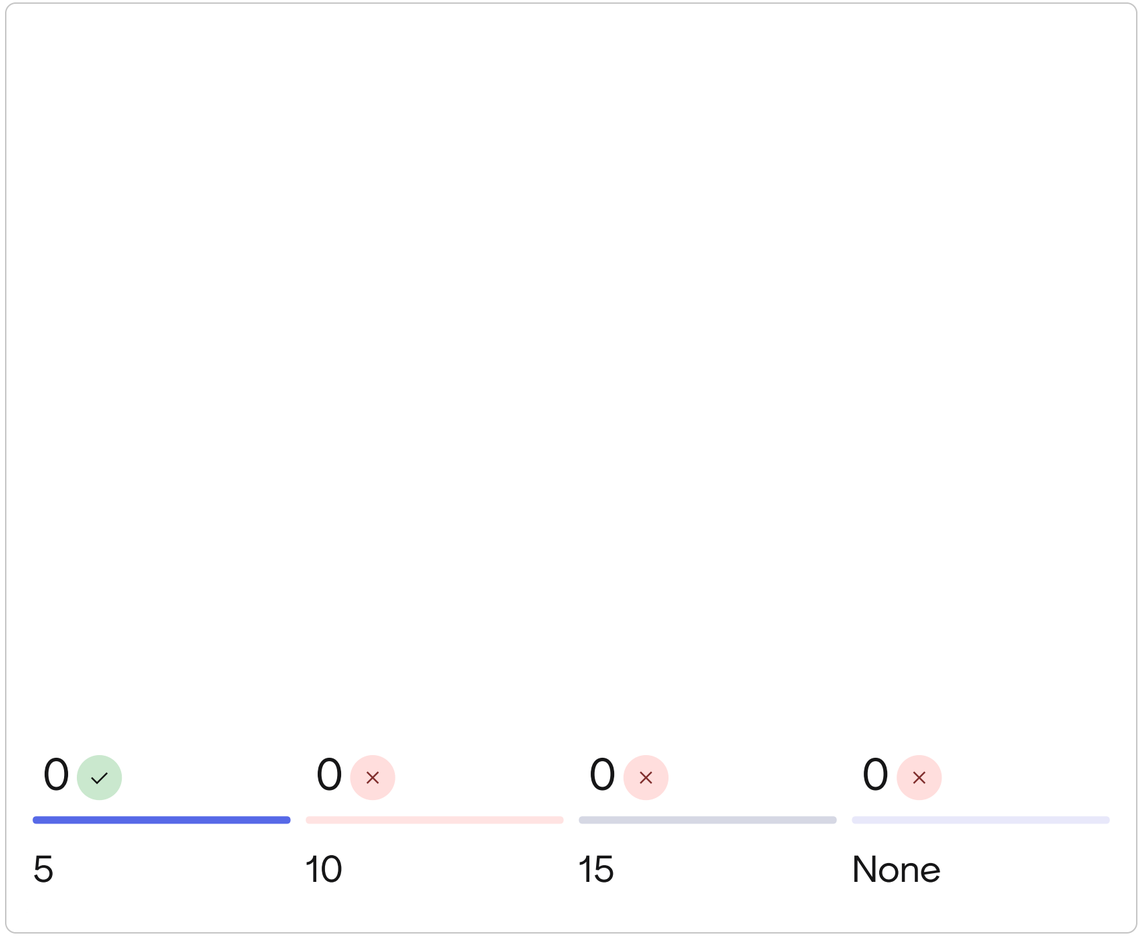
x = 5
oppdater(x)
print(x)
```

0 ✓	0 ×	0 ×	0 ×
5	10	15	None

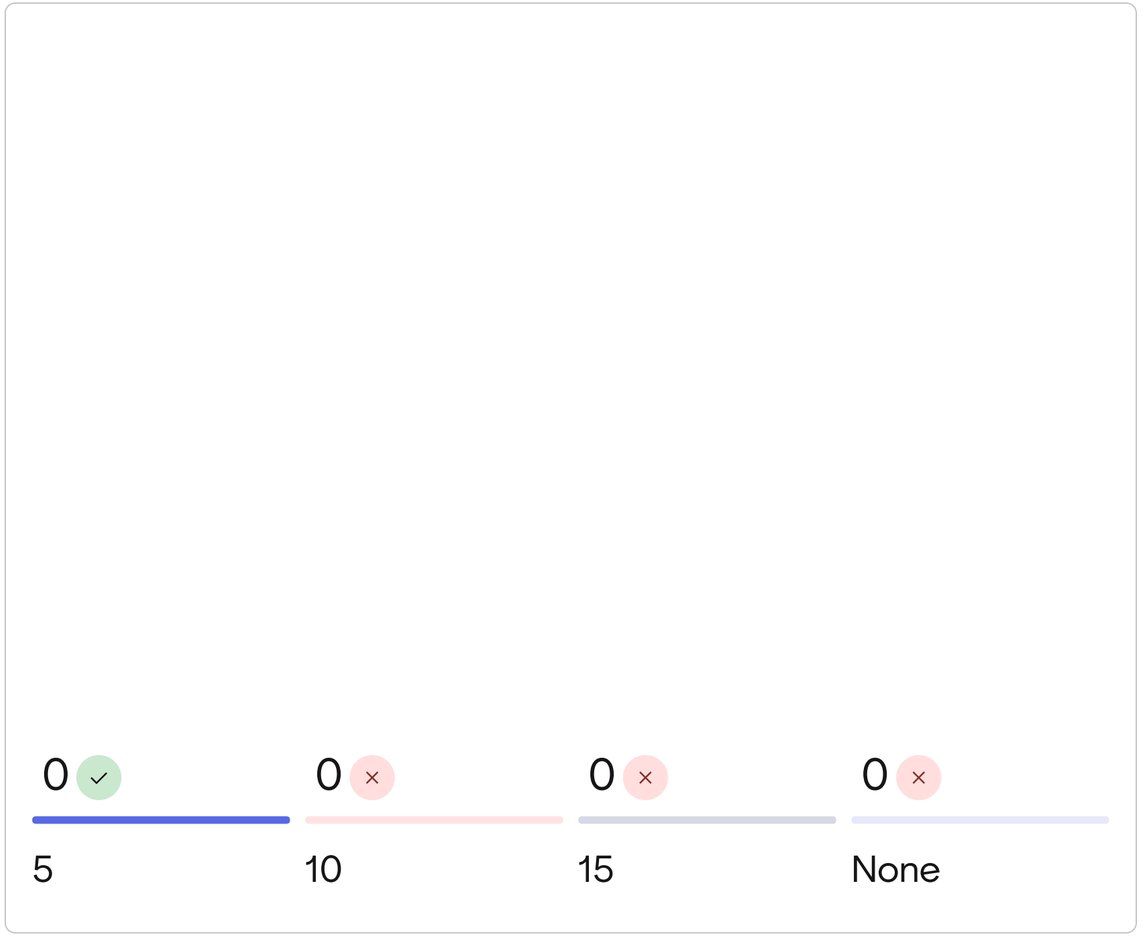
Mentimeter

Compare slides

Hva skrives ut til terminalen?



Hva skrives ut til terminalen?



Hva skrives ut til terminalen?

```
fil = open("data.txt")

linje = fil.readline().strip()
deler = linje.split(",")

total = deler[1] * 2

print(total)
```

0 ✗	0 ✓	0 ✗	0 ✗
20	1010	epleeple	Feilmelding

Mentimeter

Diskuter!

Hva skrives ut til terminalen?

```
fil = open("data.txt")

linje = fil.readline().strip()
deler = linje.split(",")

total = deler[1] * 2

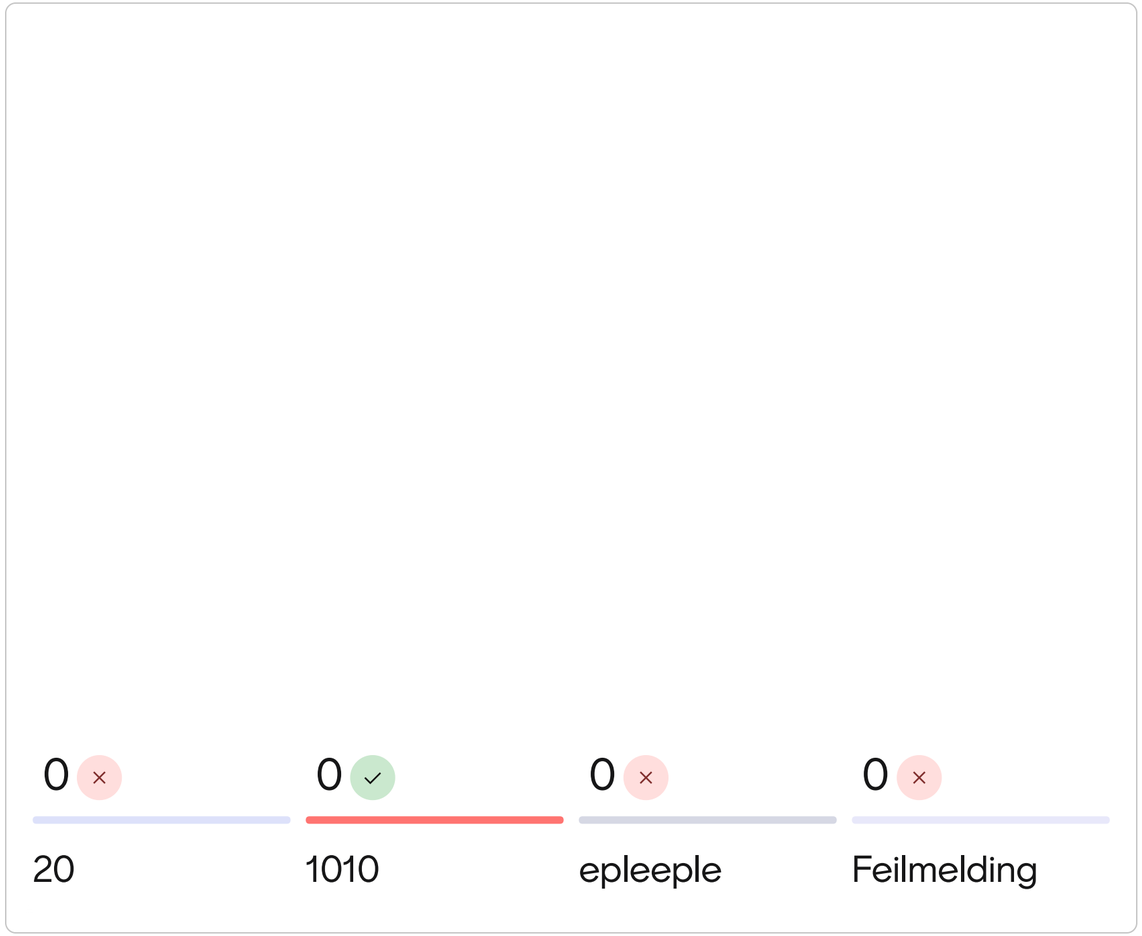
print(total)
```

0 ✗	0 ✓	0 ✗	0 ✗
20	1010	epleeple	Feilmelding

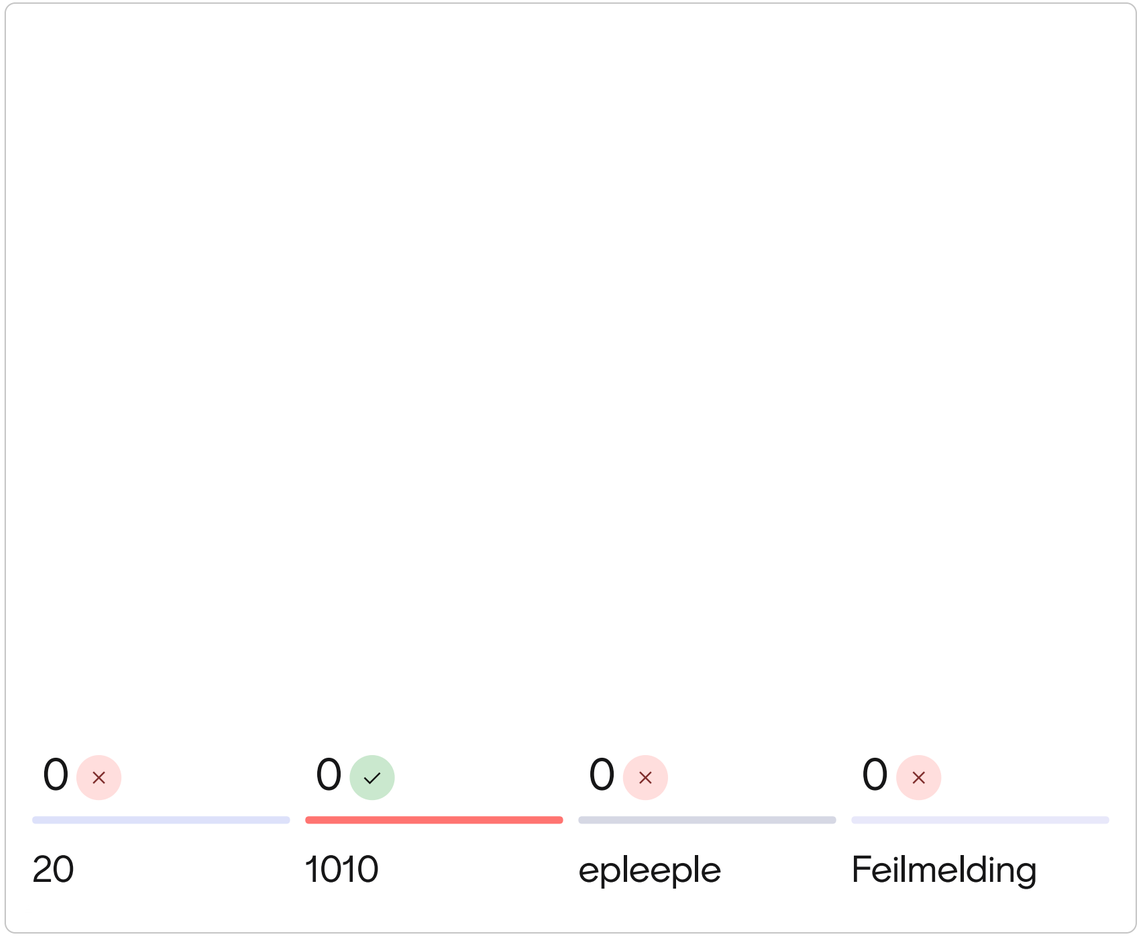
Mentimeter

Compare slides

Hva skrives ut til terminalen?



Hva skrives ut til terminalen?



Hva står i filen etter programmet har kjørt?

```
def lagre(filnavn, tekst):
    fil = open(filnavn, "w")
    fil.write(tekst)
    fil.close()

lagre("logg.txt", "Start ")
lagre("logg.txt", "Slutt")
```

0 ✗	0 ✗	0 ✗	0 ✓
Start Slutt	Start \nSlutt (\n betyr ny linje)	Start	Slutt

Mentimeter

Diskuter!

Hva står i filen etter programmet har kjørt?

```
def lagre(filnavn, tekst):
    fil = open(filnavn, "w")
    fil.write(tekst)
    fil.close()

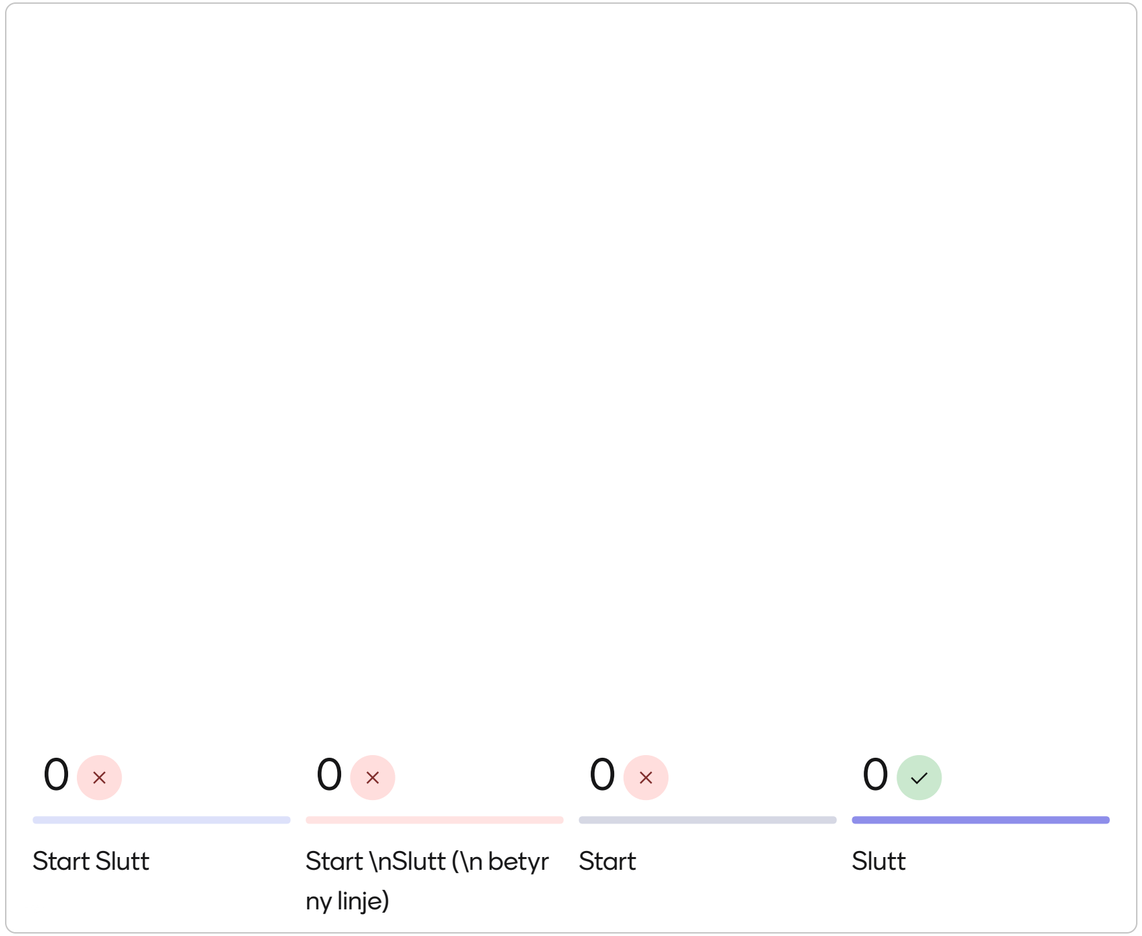
lagre("logg.txt", "Start ")
lagre("logg.txt", "Slutt")
```

0 ✗	0 ✗	0 ✗	0 ✓
Start Slutt	Start \nSlutt (\n betyr ny linje)	Start	Slutt

Mentimeter

Compare slides

Hva står i filen etter programmet har kjørt?



Hva står i filen etter programmet har kjørt?



Hva skrives ut i terminalen?

```
def les_tall(filnavn):
    fil = open(filnavn)
    tall_liste = []

    for linje in fil:
        tall_liste.append(int(linje))

    fil.close()
    return tall_liste

svar = les_tall("tall.txt")

print(svar[1] + svar[2])
```

0 ✗	0 ✓	0 ✗	0 ✗
3	5	6	Feilmelding (IndexError)

Mentimeter

Diskuter!

Hva skrives ut i terminalen?

```
def les_tall(filnavn):
    fil = open(filnavn)
    tall_liste = []

    for linje in fil:
        tall_liste.append(int(linje))

    fil.close()
    return tall_liste

svar = les_tall("tall.txt")

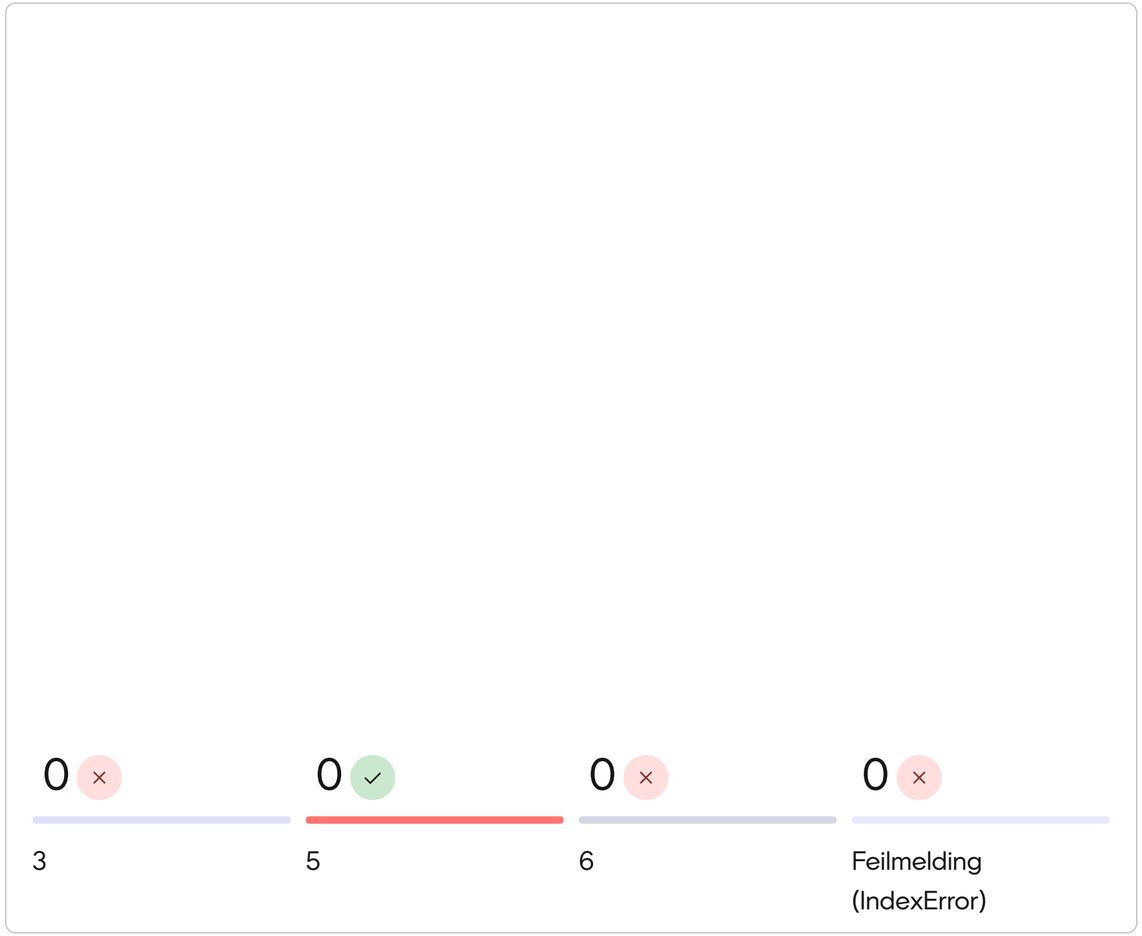
print(svar[1] + svar[2])
```

0 ✗	0 ✓	0 ✗	0 ✗
3	5	6	Feilmelding (IndexError)

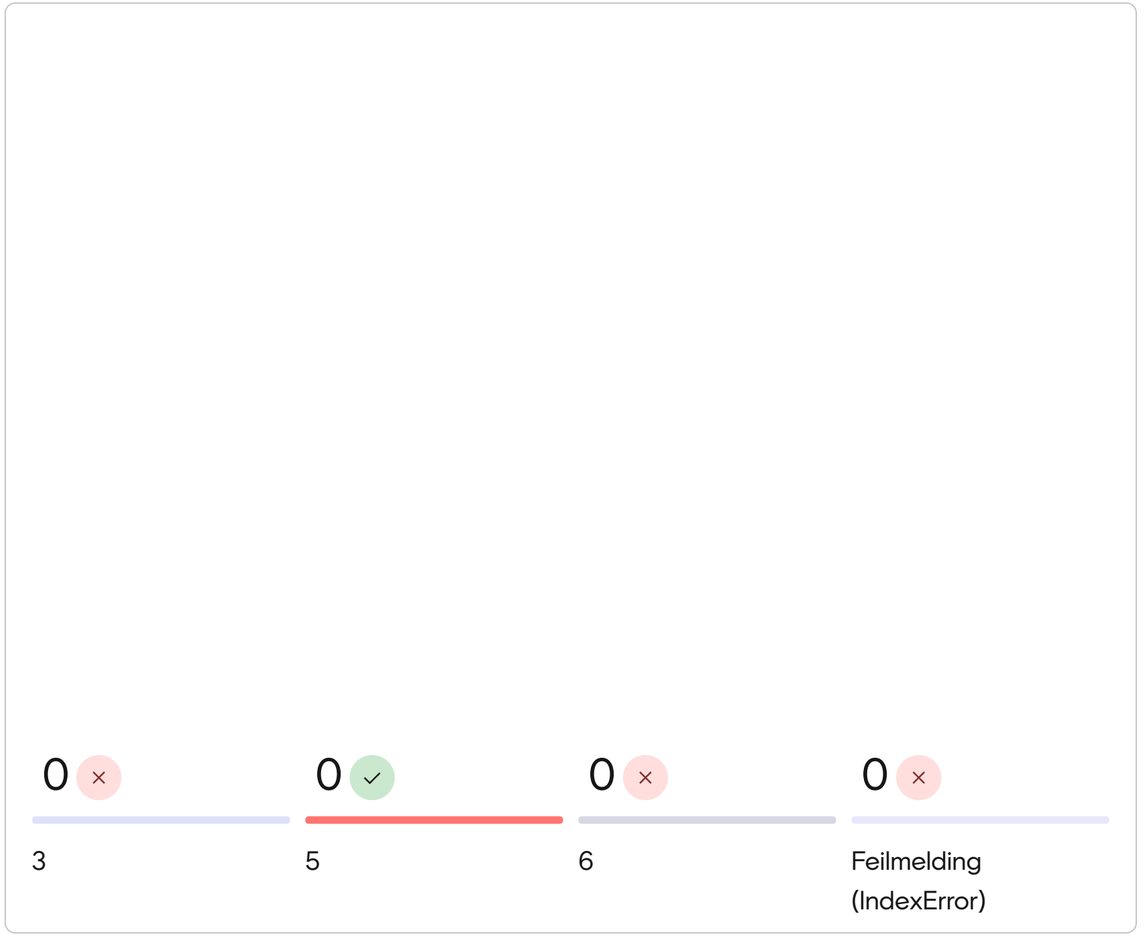
Mentimeter

Compare slides

Hva skrives ut i terminalen?



Hva skrives ut i terminalen?



Mentimeter

Parprogrammering!

Del dere opp i par (2 på hver gruppe) der dere har hver deres rolle:

- **Sjåfør - Skriver det navigatøren sier den skal skrive**
  - Skriver koden, med riktig syntaks etc.
- **Navigatør - Forteller sjåføren hva den skal skrive**
  - Tenker videre på neste steg i oppgaven, ser opp for logiske eller syntaktiske feil

Dere bytter rolle hvert 5 min

Bruk 1 minutter først på å individuelt tenke over hvordan du tenker å løse oppgaven

Deretter 2 minutter på å diskutere og snakke om hvordan løse oppgaven

Mentimeter

Per, Palle og Espen lurer på om de skal finne på noe til kvelden. Dersom nøyaktig to av dem ønsker å spille badminton går de for det, men ellers (om færre enn to eller flere enn to vil dette) gjør de heller noe annet.

Skriv en funksjon, badminton(per_vil, palle_vil, espen_vil), som tar inn tre boolske argumenter, og returnerer True dersom to (og kun to) av de tre argumentene er True.

Altså skal for eksempel kallet badmington(True, True, False) returnere True, mens kallet badmington(True, True, True) skal returnere False.

Oppgave 1

Mentimeter

Skriv en funksjon, summer_rabatt(vareliste, førpris, nypris). Den tar som argument en liste med varenavn (vareliste), en ordbok med varenavn som nøkler og førprisen som verdier (førpris), samt en ordbok med varenavn som nøkler og nyprisen som verdier (nypris).

Funksjonen skal regne ut hvor mye rabatt en kunde har fått i sum dersom kunden har kjøpt varene i vareliste (én av hvert element i lista). Rabatten for en gitt vare er forskjellen mellom førpris og nypris for det gitte varenavnet, hvor disse prisene ligger i hver sin ordbok førpris og nypris. Du kan anta at alle varenavn i vareliste finnes som nøkler i både ordboken førpris og nypris, og du kan anta at nypris alltid er lavere eller lik førpris.

For eksempel skal kallet summer_rabatt(["laptop", "ryggsekk"], {"laptop":5000, "ryggsekk":900}, {"laptop":4000, "ryggsekk":600}) returnere 1300.

Oppgave 2

Mentimeter

Du er medgangssupporter for fotballaget Brann. Det betyr at så lenge Brann gjør det ganske bra heier du på dem. Om de gjør det veldig dårlig, er du kynisk og velger heller å heie på laget som leder på tabellen.

Skriv en funksjon, heie(tabellplass_ordbok), som tar som argument en ordbok hvor nøklene er navn på fotballag og verdiene er plassering på tabellen. Dersom verdien i ordboka for nøkkelen "Brann" er 3 eller mindre, skal funksjonen returnere strengen "Brann". Ellers skal funksjonen returnere nøkkelen som i ordboka er koblet til verdien 1. Du kan anta at nøkkelen "Brann" finnes i tabellplass_ordbok, og du kan anta at tabellplass_ordbok har et lag med verdi 1.

Altså skal kallet heie({"Rosenborg":4, "Odd":1, "Molde":3, "Brann":2}) returnere "Brann", mens kallet heie({"Rosenborg":2, "Odd":1, "Molde":3, "Brann":4}) skal returnere "Odd".

Oppgave 3

Mentimeter

Tenk at dere har fått utdelt en fil med gruppeoppmøte i IN1000. Filnavn: oppmøte_in1000.txt. Skriv et program som ved hjelp av en funksjon leser inn filen og skriver ut gruppene som har hatt 20 eller flere oppmøtte studenter i uke 11. Funksjonen har filnavn som parameter. Første linje i filen består av gruppenummer: gruppe_1, gruppe_2, gruppe_3, o.s.v., til og med gruppe_8. Gruppene er atskilte med ; (semikolon). På linjene nedenfor står antall oppmøtte i hver gruppe per uke. Antallene er atskilte med ; (semikolon). Andre linje i fila har oppmøteantall for uke 1, tredje linje i fila har oppmøte for uke 2, o.s.v., til og med uke 12. Her er et eksempel på hvordan de to første linjene i oppmøte_in1000.txt kan se ut:
gruppe_1; gruppe_2; gruppe_3; gruppe_ 4; gruppe_5; gruppe_6; gruppe_7; gruppe_8
10; 19; 25; 18; 23; 11; 13; 17

Oppgave 4

Mentimeter

Et land har følgende regler for reisekarantene:
Dersom en person er vaksinert, trenger personen ikke å gå i karantene (0 dager i karantene)
Dersom en uvaksinert person har vært i et land som har fargekode rød eller oransje, skal personen 10 dager i karantene.
Dersom en uvaksinert person har vært i et land som har fargekode grønn, skal personen 3 dager i karantene.

Skriv en funksjon, karantene(vaksinert, farge), som returner antall dager (heltall) en person må i karantene i henhold til reglene ovenfor. Parameteren vaksinert er enten True eller False (boolsk verdi) og parameteren farge er enten "rød", "oransje" eller "grønn" (streng).

Altså skal for eksempel kallet karantene(True, "roed") returnere 0, mens kallet karantene(False, "oransje") skal returnere 10.

Utvid programmet i a), slik at det leser inn en fil ved navn karantene.txt. Hver linje i filen består av vaksinasjonsstatusen til en person (True eller False) og fargen på landet de har vært i (rød, oransje eller grønn). Vaksinasjonsstatus og farge er atskilte med komma. Videre skal programmet summere og skrive ut antall karantenedager fra henholdsvis røde, oransje og grønne land (antall for hver farge separat samt antall karantenedager totalt).

Oppgave 5

Mentimeter

Du ønsker hjelp til å gruppere sammen personer som har felles interesse.

Skriv en funksjon, lag_interessegrupper(personers_interesse), som tar som argument en ordbok hvor nøklene er strenger som er navn på personer, mens hver verdi er en streng som sier hvilken interesse personen har.

Funksjonen skal returnere en ordbok, hvor hver nøkkel er en interesse (streng) og hver verdi er en liste av personer som var oppført med denne interessen i ordboken personers_interesse. Alle interessene som finnes i ordboken personers_interesse skal være med i den returnerte ordboken.

Altså skal for eksempel kallet lag_interessegrupper({"Per":"Mat", "Palle":"Film", "Espen":"Mat"}) evaluere til ordboken {"Mat":["Per","Espen"], "Film":["Palle"]}

Utvid programmet i a), slik at det lager en fil med interesser og antall personer som har hver interesse. Hver linje skal bestå av interesse og antall, atskilte med komma. Filen skal hete interesseantall.txt.

Oppgave 6

Skriv ned en ting som ble klarere og en ting du fortsatt sliter med:) Og tilbakemeldinger på gruppetimene!