INNFØRING I PROGRAMMERING

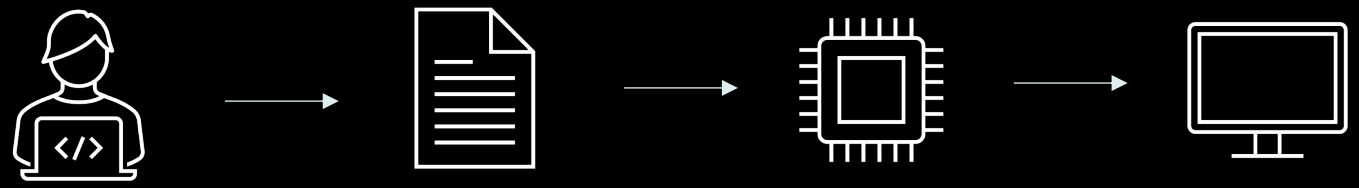
INF100

VÅR 2025

Torstein Strømme

Torstein Strømme

HVA ER PROGRAMMERING?



```python
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total earnings: {total_earnings}')
```

```
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total earnings: {total_earnings}')
```

Variabler	Verdier

```python
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total earnings: {total_earnings}')
```

Variabler	Verdier
	200

```python
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total earnings: {total_earnings}')
```

Variabler	Verdier
hourly_wage	200

```
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total earnings: {total_earnings}')
```

Variabler	Verdier
hourly_wage	200

```
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total earnings: {total_earnings}')
```

Variabler	Verdier
hourly_wage	200
	7.5

```
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total earnings: {total_earnings}')
```

Variabler	Verdier
hourly_wage	200
daily_hours	7.5

```
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total earnings: {total_earnings}')
```

Variabler	Verdier
hourly_wage	200
daily_hours	7.5

```python
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total earnings: {total_earnings}')
```

Variabler	Verdier
hourly_wage	200
daily_hours	7.5
	20

```python
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total earnings: {total_earnings}')
```

Variabler	Verdier
hourly_wage	200
daily_hours	7.5
days_worked	20

```
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total earnings: {total_earnings}')
```

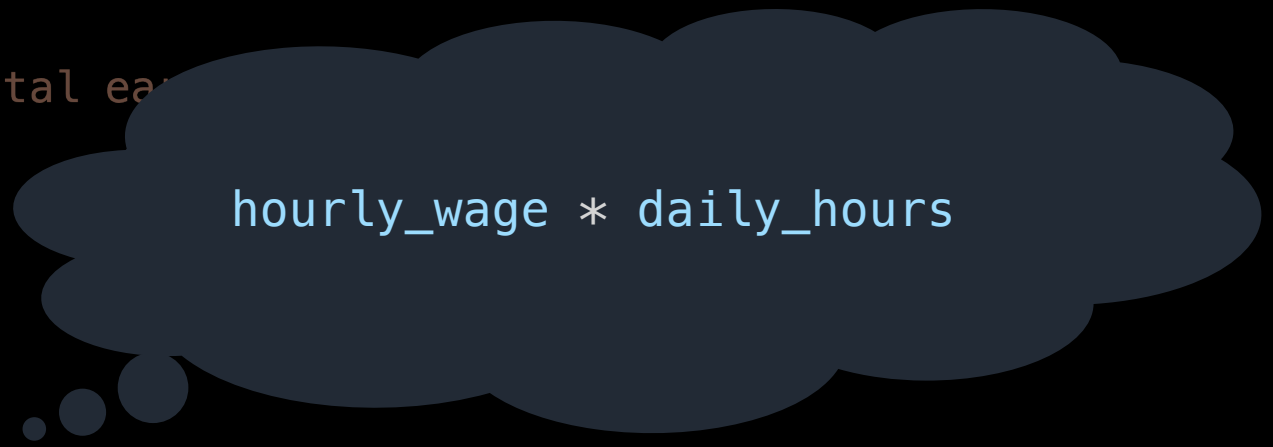
Variabler	Verdier
hourly_wage	200
daily_hours	7.5
days_worked	20

```
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total ea
```

Variabler	Verdier
hourly_wage	200
daily_hours	7.5
days_worked	20



Variabler	Verdier
hourly_wage	200
daily_hours	7.5
days_worked	20

```
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total ea
```

200 * daily_hours

Variabler	Verdier
hourly_wage	200
daily_hours	7.5
days_worked	20

```
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total ea
```

200 * 7.5

```python
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total ea
```

Variabler	Verdier
hourly_wage	200
daily_hours	7.5
days_worked	20

```python
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total earnings: {total_earnings}')
```

Variabler	Verdier
hourly_wage	200
daily_hours	7.5
days_worked	20
	1500

```python
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total earnings: {total_earnings}')
```

Variabler	Verdier
hourly_wage	200
daily_hours	7.5
days_worked	20
daily_earnings	1500

```python
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total earnings: {total_earnings}')
```

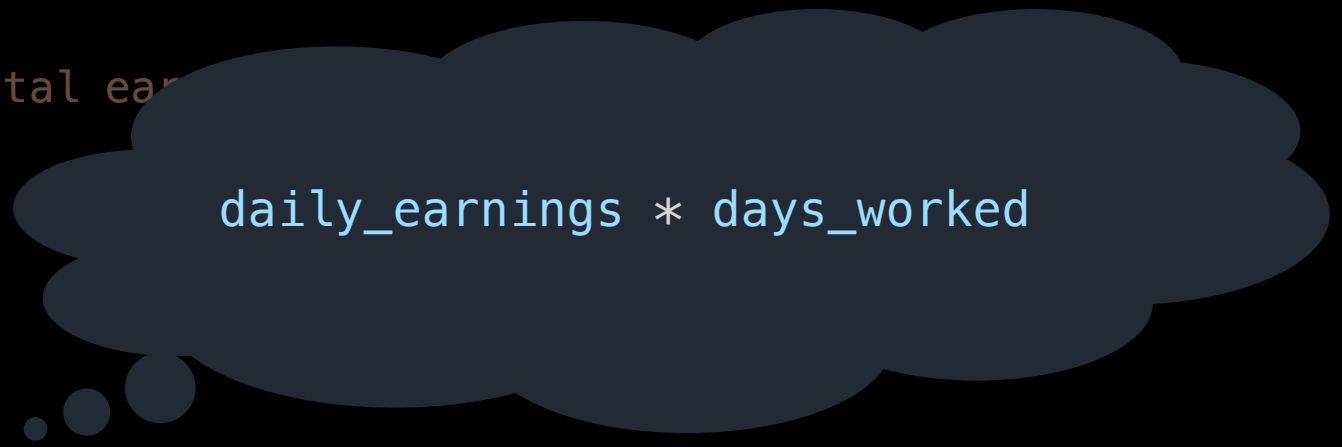
Variabler	Verdier
hourly_wage	200
daily_hours	7.5
days_worked	20
daily_earnings	1500

Variabler	Verdier
hourly_wage	200
daily_hours	7.5
days_worked	20
daily_earnings	1500

```
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total ea
```



```
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total ea
```

Variabler	Verdier
hourly_wage	200
daily_hours	7.5
days_worked	20
daily_earnings	1500

```
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total ea
```

Variabler	Verdier
hourly_wage	200
daily_hours	7.5
days_worked	20
daily_earnings	1500

```
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total ea
```

Variabler	Verdier
hourly_wage	200
daily_hours	7.5
days_worked	20
daily_earnings	1500

```python
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total earnings: {total_earnings}')
```

Variabler	Verdier
hourly_wage	200
daily_hours	7.5
days_worked	20
daily_earnings	1500
	30000

```
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total earnings: {total_earnings}')
```

Variabler	Verdier
hourly_wage	200
daily_hours	7.5
days_worked	20
daily_earnings	1500
total_earnings	30000

```python
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total earnings: {total_earnings}')
```

Variabler	Verdier
hourly_wage	200
daily_hours	7.5
days_worked	20
daily_earnings	1500
total_earnings	30000

```python
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total earnings: {total_earnings}')
```

Variabler	Verdier
hourly_wage	200
daily_hours	7.5
days_worked	20
daily_earnings	1500
total_earnings	30000

```python
# Input
hourly_wage = 200
daily_hours = 7.5
days_worked = 20

# Calculation
daily_earnings = hourly_wage * daily_hours
total_earnings = daily_earnings * days_worked

# Output
print(f'Total earnings: {total_earnings}')
```

→

Variabler	Verdier
hourly_wage	200
daily_hours	7.5
days_worked	30
daily_earnings	1500
total_earnings	30000

HVEM ER VI?

Torstein Strømme

Administrator for INF100

- Tilrettelegging
- Emneregistrering
- Studieprogram

Linn Astrid Blix Torget

Lab-admin

- Utsettelser
- Feil med retting

Gutama Ibrahim Mohammad

Hilde Jordal

Amy Zhang

Aleksandr «Sasha» Popov

GRUPPELEDERE

EN UKE I INF100

Fre
kickstart

	Man	Tirs	Ons	Tors	Fre
8-10					
10-12					
12-14					
14-16		forelesning			kickstart

Fredag 14:00
Lab publiseres

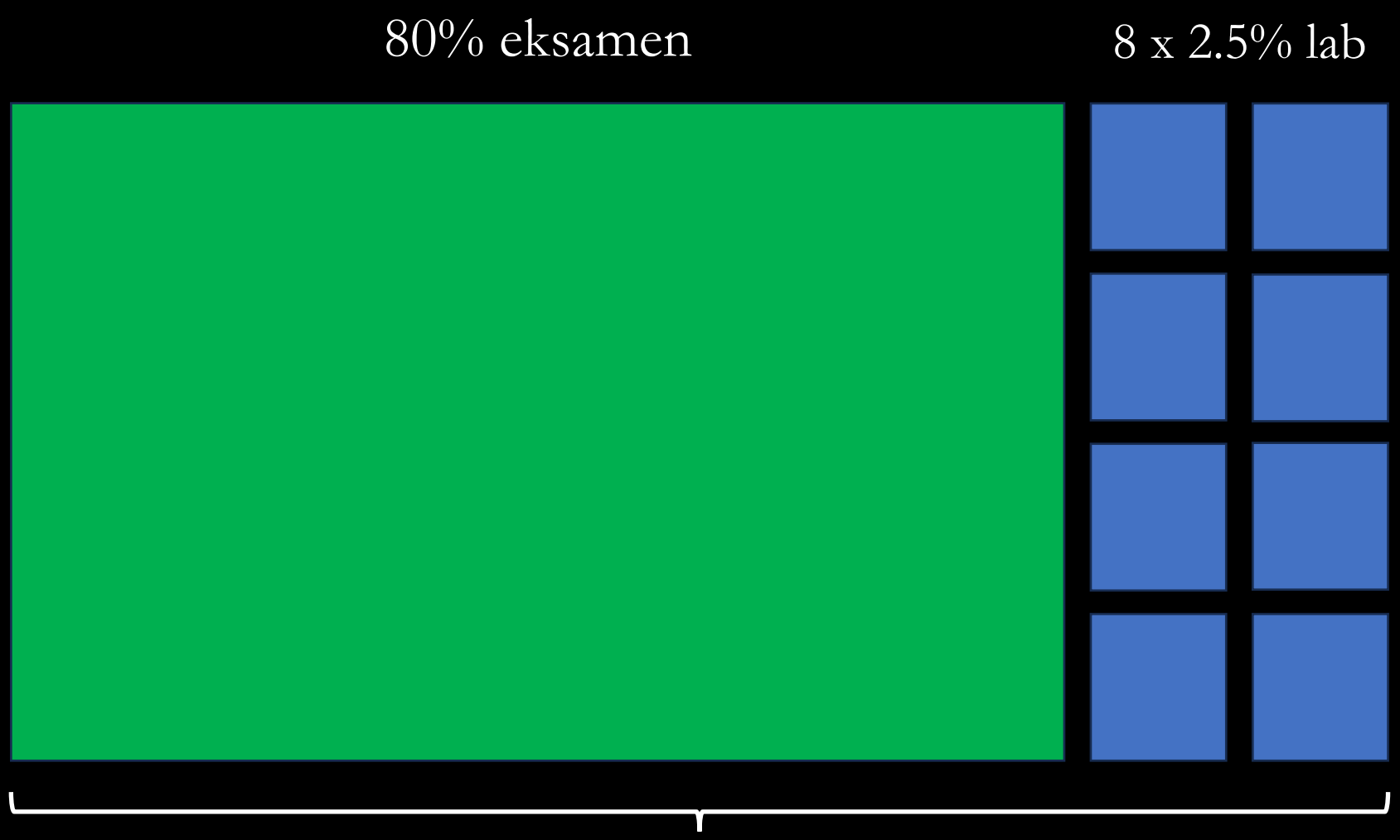
I løpet av uken
- gruppetime
- drop-in gruppetime
- egenstudie

Fredag 23:59
Lab-frist

EN UKE I INF100

	Man	Tirs	Ons	Tors	Fre
8-10					
10-12					
12-14					
14-16		for			

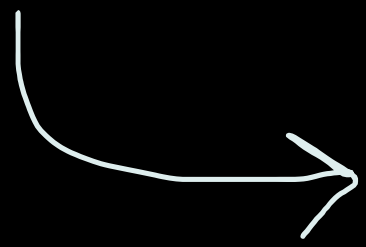
Drop-in gruppetime


80% eksamen
8 x 2.5% lab
Karakter

ARBEIDSKRAV

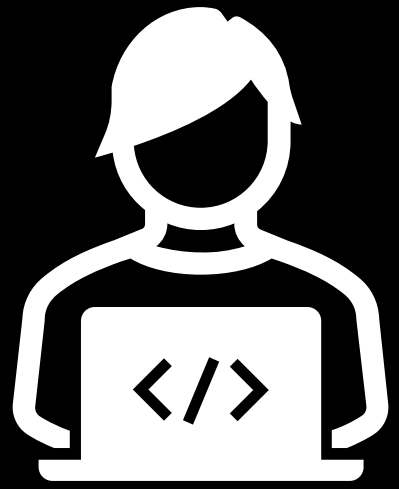
- Gjennomfør 2/2 quizer
- Gjennomfør 3/3 presentasjoner med god innsats
- Få minst 50% poeng på labene

Ved uforskyldte forhold som sykdom, heimevernsøvelse etc., søk om utsettelse eller fritak til Gutama.

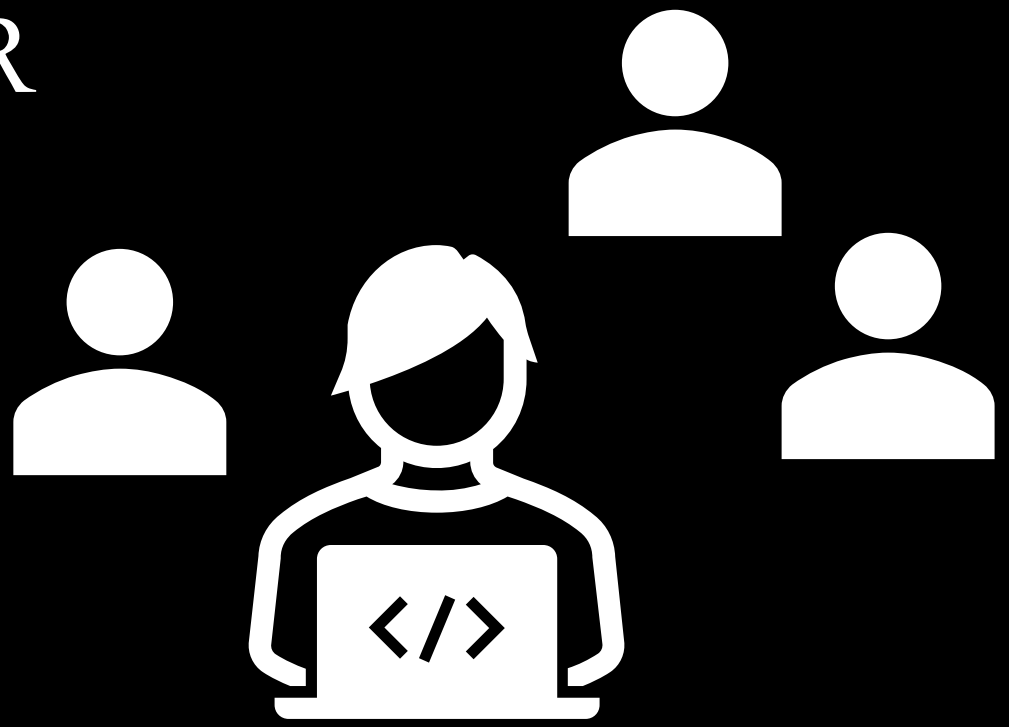


LAB

- 25 poeng per lab
- Frist: 1 uke



PRESENTASJONER

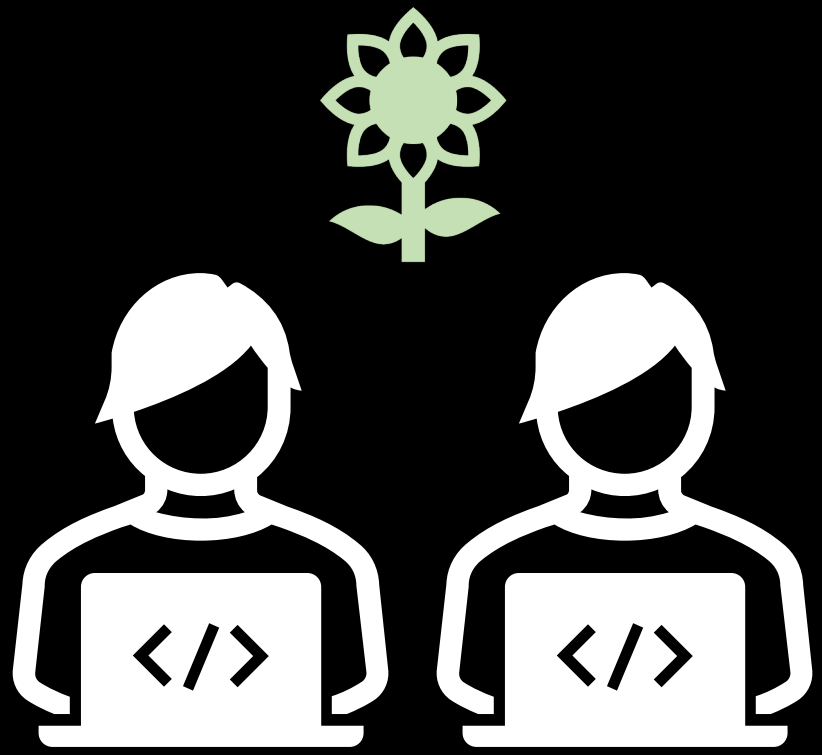


- Tre ganger i semesteret
- Publikum:
  din gruppeleder + ca 3 medstudenter
- Varighet:
  10 minutter presentasjon + spørsmål og diskusjon (x4)

VIKTIG!

MELD DEG INN I EN GRUPPE

SAMARBEID



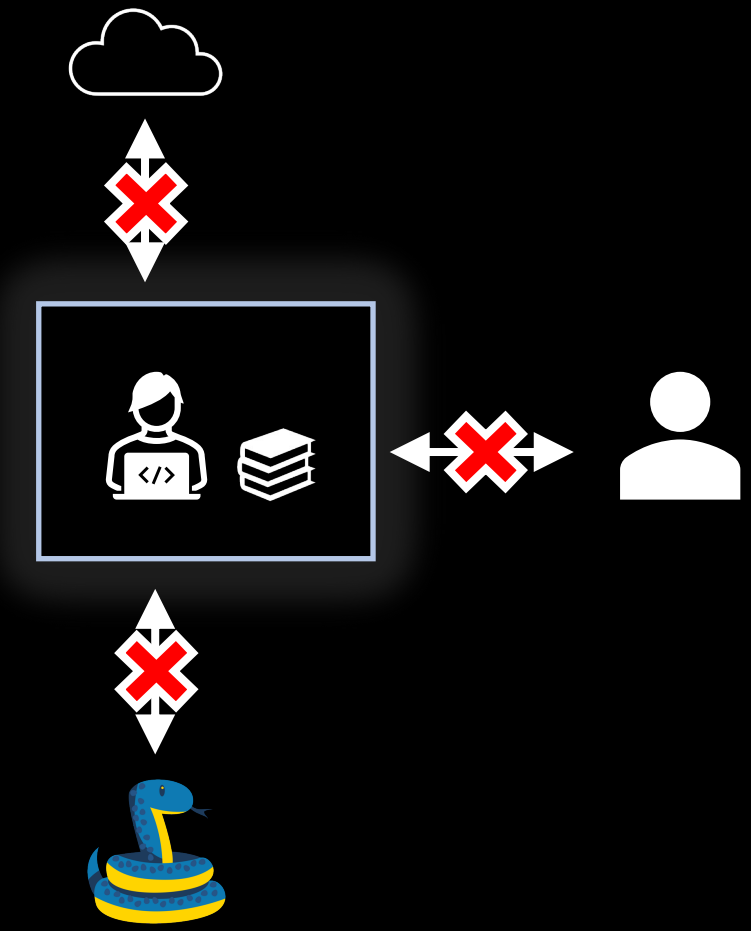
- Samarbeid er **bra**
- Innleveringer er **individuelle**
  - du må skrive og forstå alt selv

- **Viktig å sitere**
  - samarbeid
  - hjelp man mottar
  - kilder man benytter

husk **fullt navn** når du siterer

EKSAMEN

- Lukket digital eksamen
- Opp til 6 sider egne notater
- Viktig å sitere kilder



PLATTFORMER

mitt.uib.no
- Kunngjøringer
- Oppgaveinnlevering
- Strømming av forelesning
- Opptak av forelesning

inf100.ii.uib.no
- Kursnotater
- Oppgavetekster

CodeGrade
- Automatisk retting av lab'er
- Manuell feedback på lab'er
- Tilgang via mitt.uib.no

Discord
- Spørsmål og svar
- Tips og triks

SNARVEIPRØVE

- For deg som allerede kan programmere (altså helt frivillig)
- Automatisk rettet
- Over 70% →
  - Alle presentasjoner regnes som bestått
  - Poeng for labX beregnes som `max(labX_score, lab8_score)`

- Fredag 24. januar kl 16:15 – 17:15
  - Fysisk oppmøte i Audiotorium A, Allégaten 66
  - Tillatte hjelpemidler: enkel kalkulator, penn og papir
  - Ta med: student-id og egen laptop. Sjekk at Safe Exam Browser virker.

I DAG

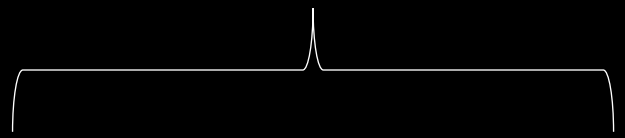
- Hva er programmering?
- Noen eksempler
- Ordbok

- Variabler
- Kodesporing
- Grafikk

LIVEKODING: EKSEMPLER

ORDBOK

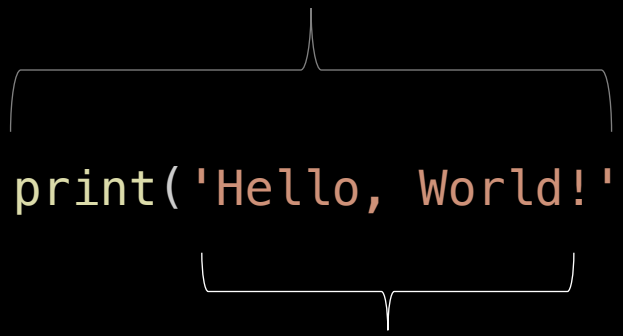
setning (engelsk: statement). Ett «steg» i et program, ofte én linje.



```
print('Hello, World!')
```

ORDBOK

setning (engelsk: statement). Ett «steg» i et program, ofte én linje.



verdi. En eller annen form for data som benyttes i programmet.

Eksempler på verdier:

```
'Hello, World!'   '42'
 42     3.14     True
      [4, 3, 2, 3]
```

ORDBOK

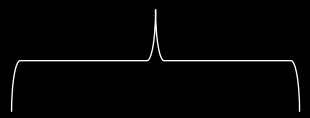
setning (engelsk: statement). Ett «steg» i et program, ofte én linje.

verdi. En eller annen form for data som benyttes i programmet.

funksjon. En «kommando» som kan få noe til å skje.

ORDBOK

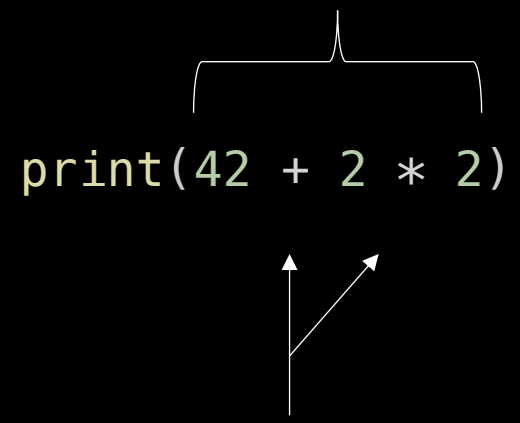
uttrykk (engelsk: expression). Et regnestykke som evaluerer til en verdi.



```
print(42 + 2 * 2)
```

ORDBOK

uttrykk (engelsk: expression). Et regnestykke som evaluerer til en verdi.

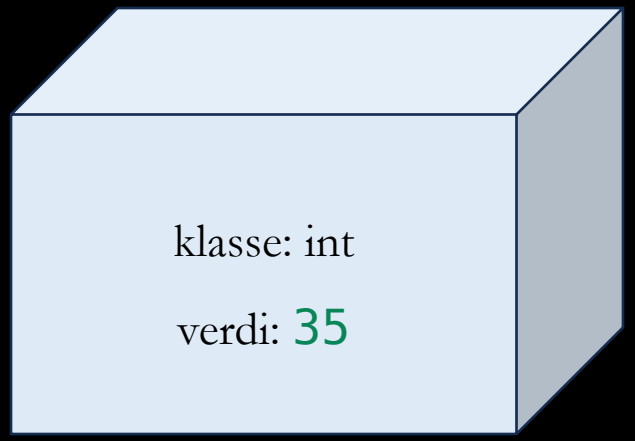


operasjon. En måte å kombinere to verdier for å produsere en ny verdi.

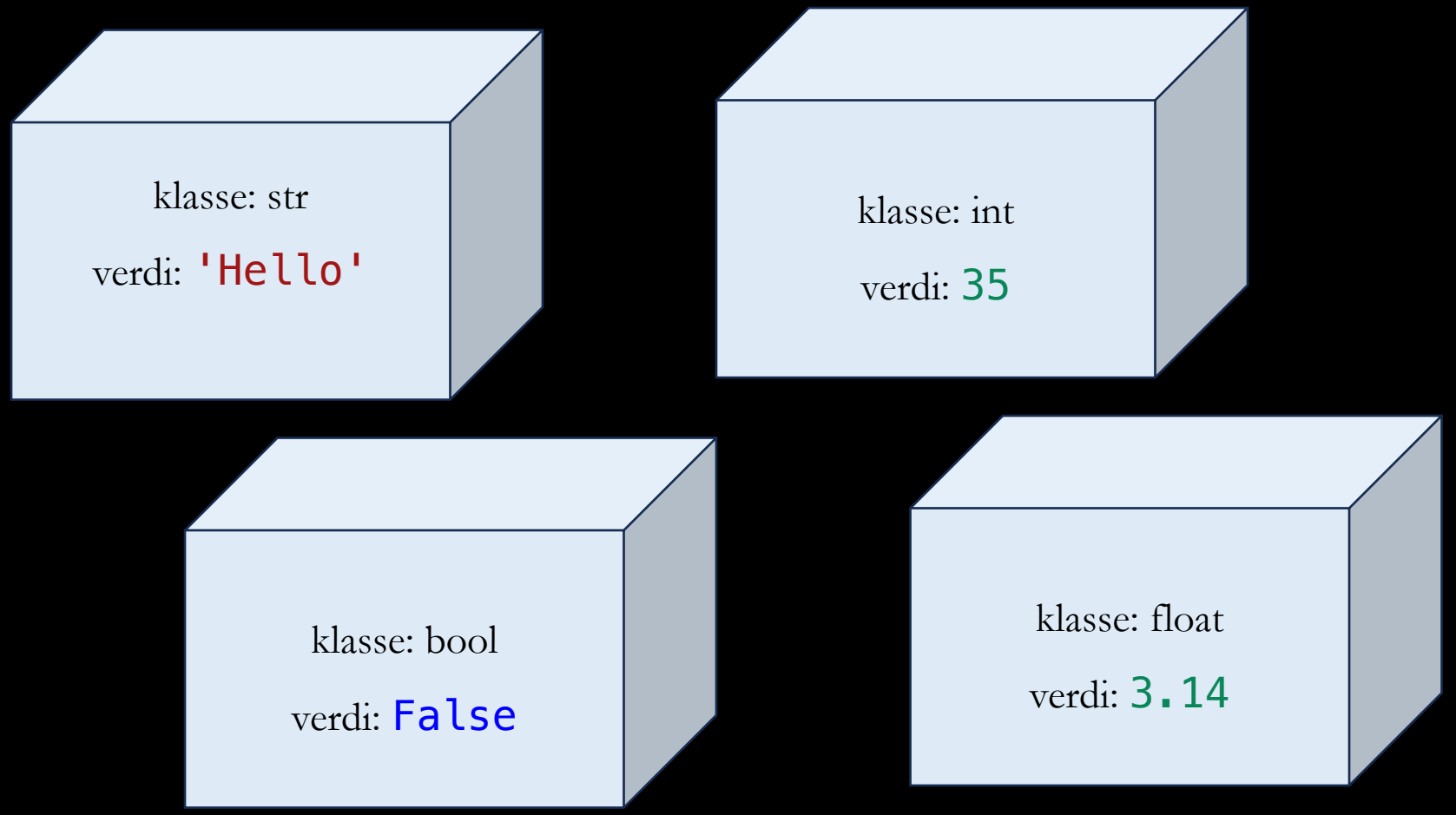
Eksempler på operasjoner:

```
+  -  *  /  **  //  %
```

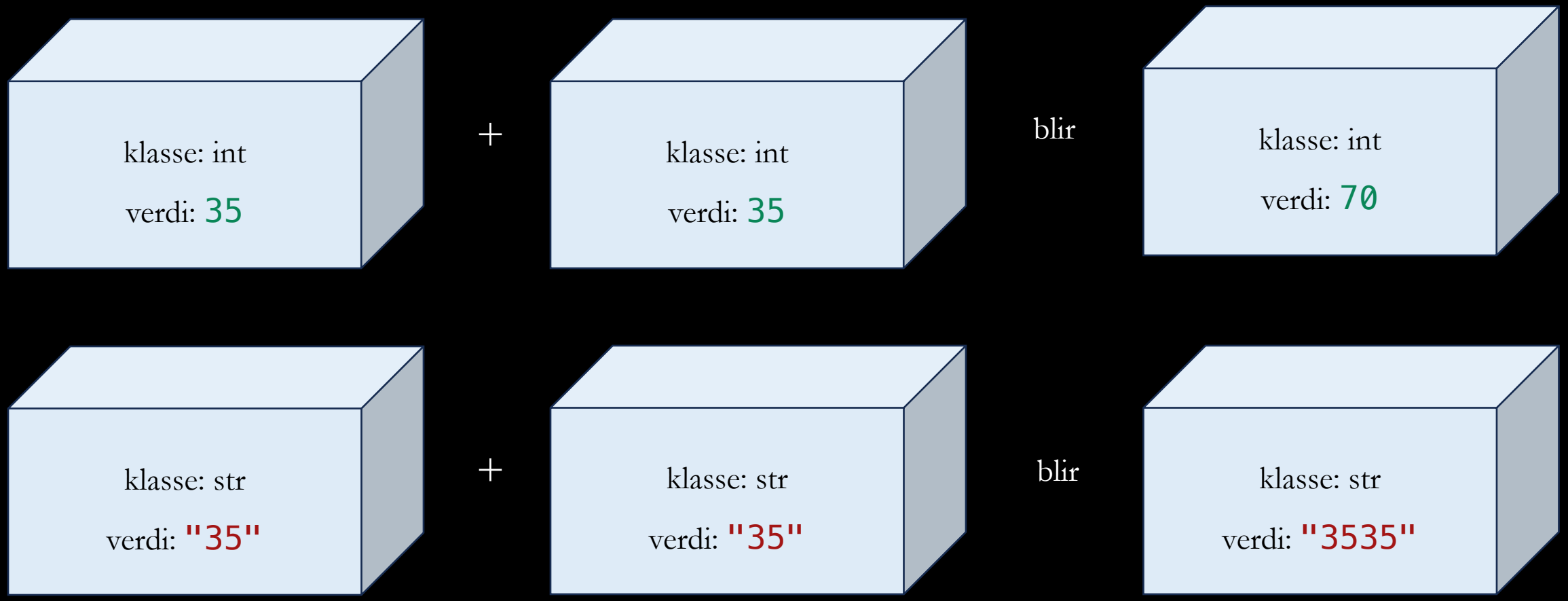
VERDIER/OBJEKTER



VERDIER/OBJEKTER



KLASSEN BESTEMMER OPERASJONEN



KONVERTERING MELLOM TYPER

```
print('Skriv inn første tall')
x = input()

print('Skriv inn andre tall')
y = input()



print('Summen av tallene er', x + y) # feil
```

KONVERTERING MELLOM TYPER

```
print('Skriv inn første tall')
x = input()

print('Skriv inn andre tall')
y = input()

x = int(x)
y = int(y)

print('Summen av tallene er', x + y) # riktig!
```

KONVERTERING MELLOM TYPER

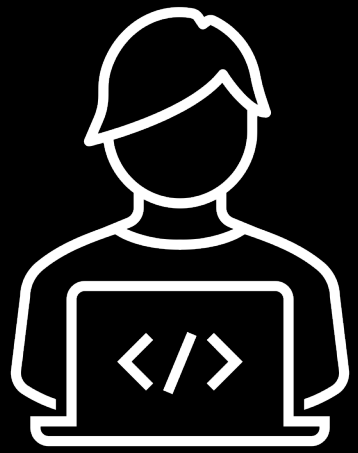
```
print('Skriv inn første tall')
x = int(input())

print('Skriv inn andre tall')
y = int(input())
```

```
print('Summen av tallene er', x + y) # riktig!
```

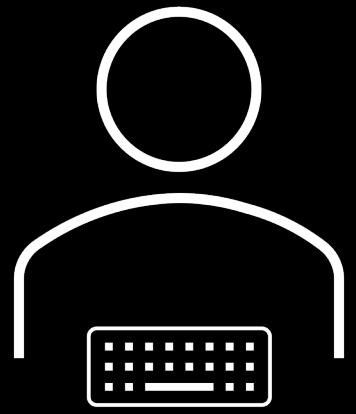
PYTHON
KILDEKODE

PYTHON
FORTOLKER



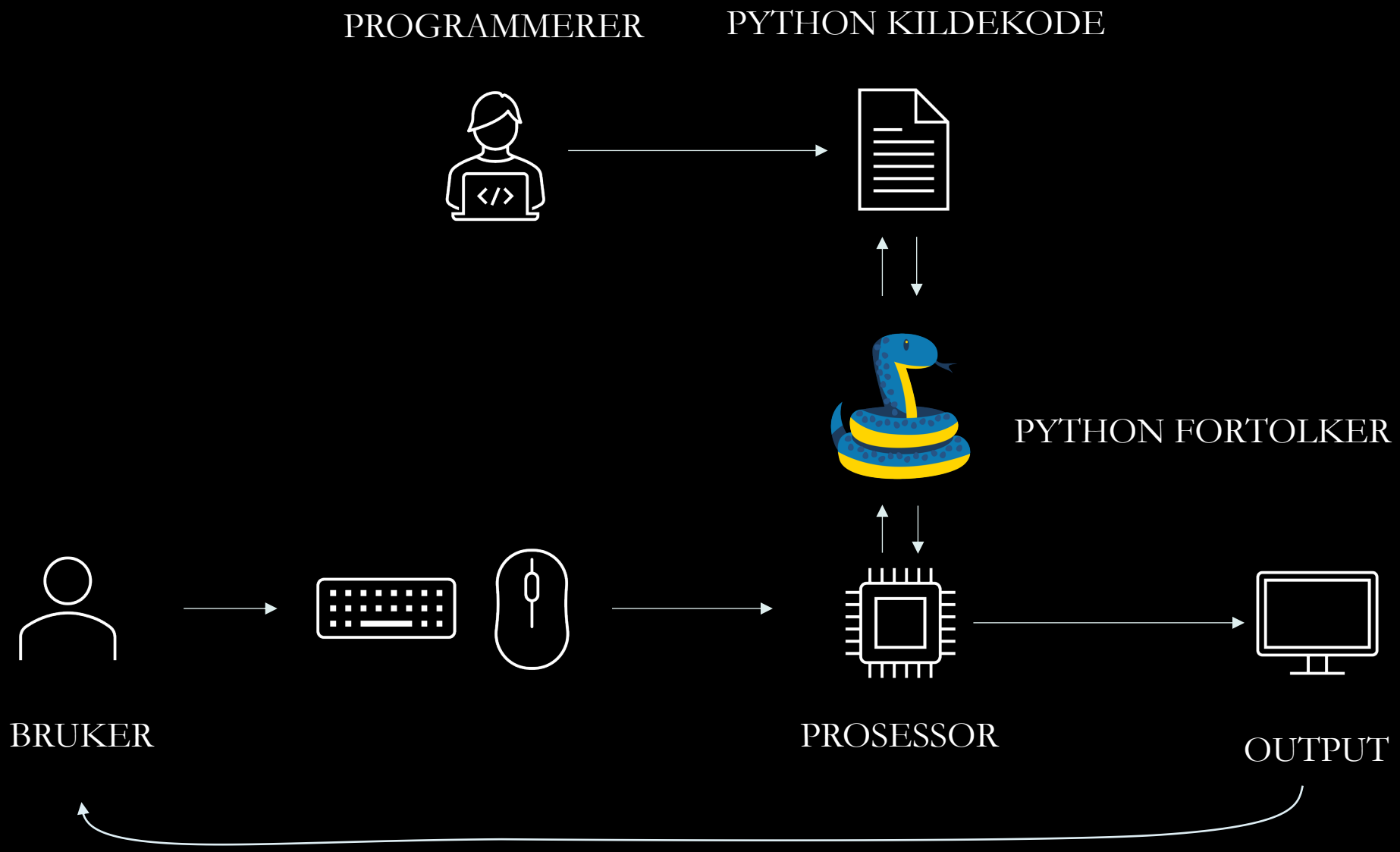
PROGRAMMERER

før programmet kjører



BRUKER

når programmet kjører


PROGRAMMERER
PYTHON KILDEKODE
PYTHON FORTOLKER
BRUKER
PROSESSOR
OUTPUT

KODESPORING

- Koden utføres linje for linje
- Forutsi hva som skjer i neste steg

KODESPORING: EKSEMPEL

```
balance = 1000
org_balance = balance

# Første år: 5% rente
interest_rate = 0.05
interest_amount = balance * interest_rate
balance = balance + interest_amount

# Andre år: 10% rente
interest_rate = 0.10
interest_amount = balance * interest_rate
balance = balance + interest_amount

difference = balance - org_balance
print(f'Etter to år har du tjent {difference} kr i renter')
```

KODESPORING

```
balance = 1000
org_balance = balance

# Førs
intere
intere
balanc

# Andr
intere
intere
balanc

differ
print(
```

linje	balance	org_balance	interest_rate	interest_amount	difference	print
1	~~1000~~					
2		1000				
5			~~0.05~~			
6				~~50.0~~		
7	~~1050.0~~					
10			0.10			
11				105.0		
12	1155.0					
15					155.0	
16						Etter to år har du tjent 155.0 kr i renter

EKSAMENSOPPGAVE HØST 24

1(b)

```
a = 1
b = 2
a = a + b
b = a * b
a -= 1
print(b - a)
```

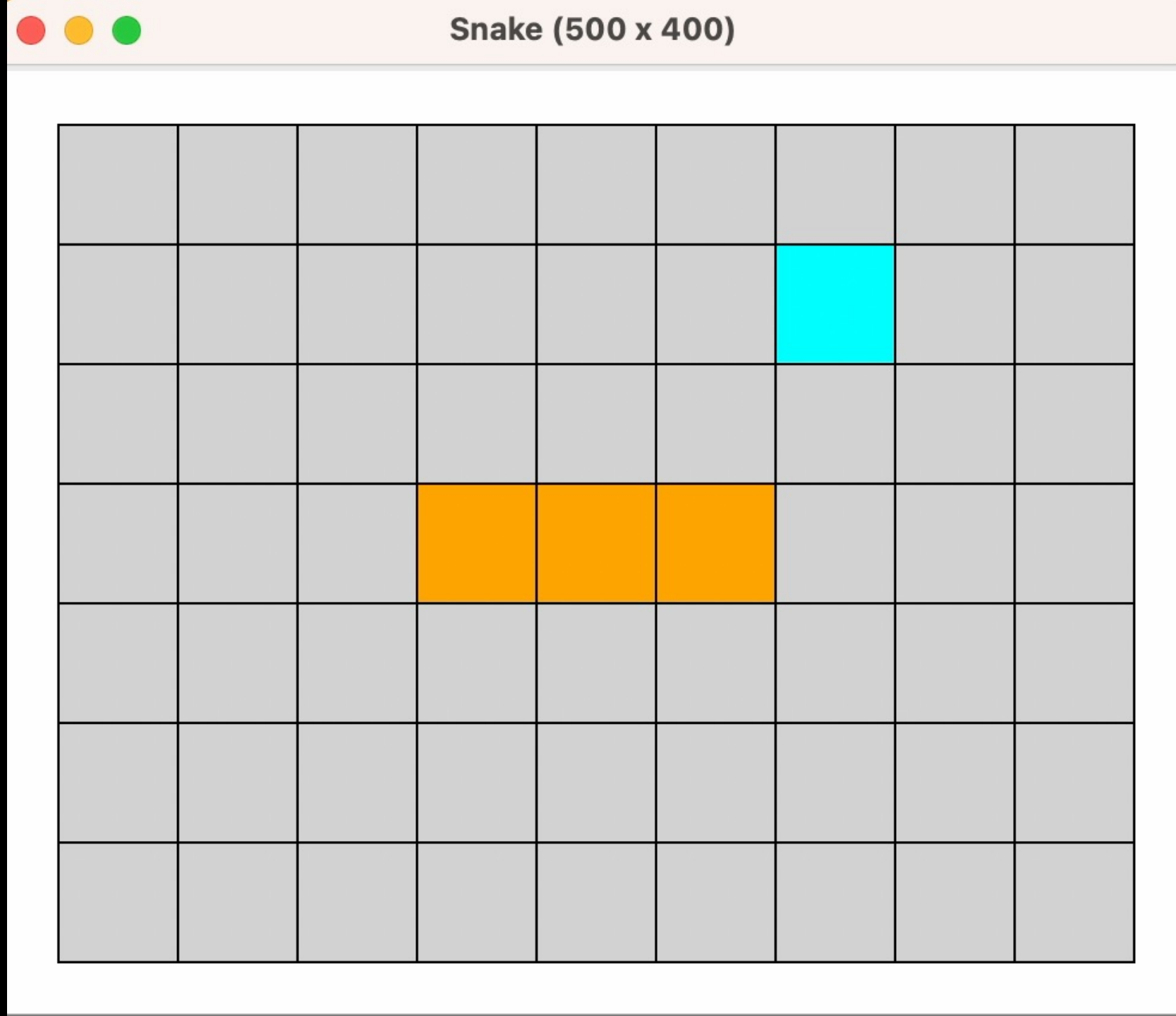
Kva skriv dette programmet ut? (hvis programmet krasjar, skriv berre Error)

GRAFIKK

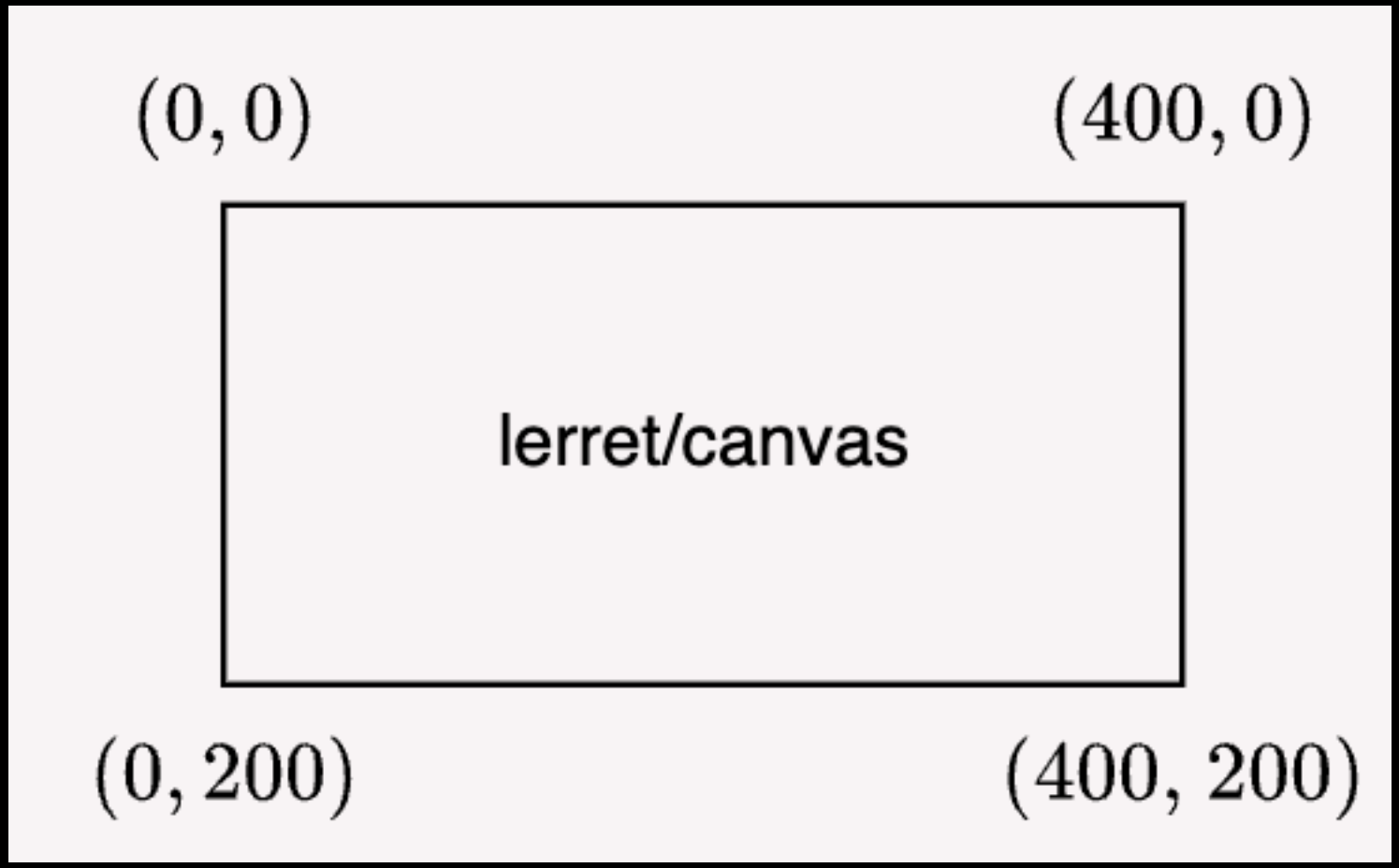
- Hvorfor grafikk?
  - **Visuell forståelse av programmering**
  - Nødvendig matematikk er på grunnskolenivå
  - Kan lage «ekte» applikasjoner
  - Lett å vise frem
  - Gøy

GRAFIKK

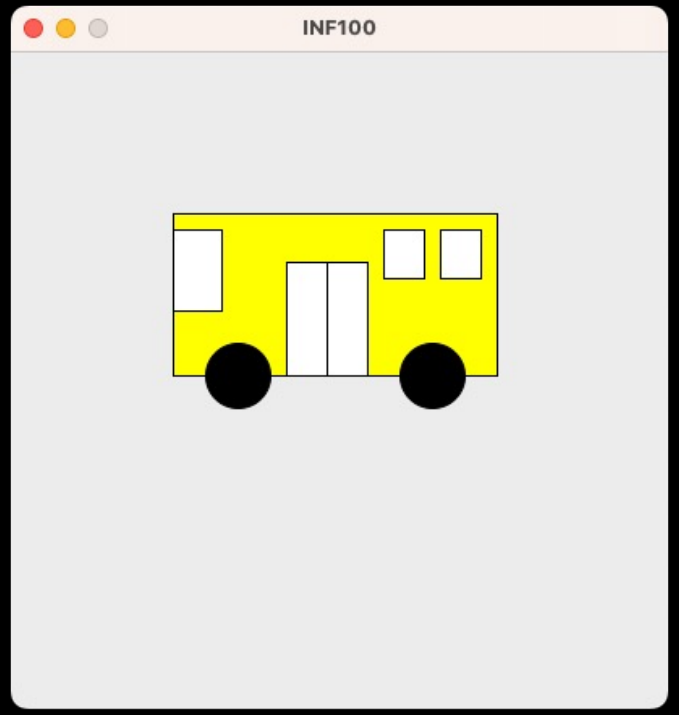
- Om noen uker:



GRAFIKK



GRAFIKK




```python
from uib_inf100_graphics.simple import canvas, display

# Bus body
canvas.create_rectangle(100, 100, 300, 200, fill='yellow')

# Wheels
canvas.create_oval(120, 180, 160, 220, fill='black')
canvas.create_oval(240, 180, 280, 220, fill='black')

# Windows
canvas.create_rectangle(100, 110, 130, 160, fill='white')
canvas.create_rectangle(230, 110, 255, 140, fill='white')
canvas.create_rectangle(265, 110, 290, 140, fill='white')

# Door
canvas.create_rectangle(170, 130, 220, 200, fill='white')
canvas.create_line(195, 130, 195, 200)

display(canvas)
```