

DIG111

Algoritmer og programmering

David Grellscheid

Kommunikasjon

- <https://mitt.uib.no/courses/40867>
gjærne bruk Discord for spørsmål

- Discord: <https://discord.gg/K4Jyhev6ZN>

The image shows two overlapping screenshots. The background screenshot is the DIGI111 course page on the University of Bergen's website. It features a red sidebar with navigation links: Konto, Dashboard, Emner, Kalender, Meldinger, Historikk, and Hjelp. The main content area is titled 'DIGI111 22H / Algoritmar og programmering' and includes a 'Tidsplan' section with information about the course schedule and contact details for the lecturer, David Grellescheid. The foreground screenshot is a Discord server interface for 'UiB_DIGI111_22H'. It displays a list of channels under the categories 'INFORMATION', 'TEXT CHANNELS', and 'VOICE CHANNELS'. The 'announcements' channel is highlighted under the 'INFORMATION' category.

DIGI111

Høst 2022

Hjem

Diskusjoner

Personer

Sider

Emnets videoer

Mine videoer

Videseinar

Tidsplan

Forelesningene begynner i uke 35 (31.08.) og holdes [online på Zoom hver onsdag kl. 18:00](#).

Opptak av Zoom-møtene er tilgjengelige gjennom hele semesteret.

Videre informasjon legges ut her, tidlig i uke 35.

Om du har spørsmål om emnet, ta kontakt med foreleseren david.grellescheid@uib.no eller studieveileder ii@uib.no.

UiB_DIGI111_22H

INFORMATION

announcements

notes-resources

TEXT CHANNELS

general

technical-help

off-topic

VOICE CHANNELS

Lounge

Study Room 1

Study Room 2

Faglige spørsmål: Discord-kanaler

(melding på mitt.uib)

(David.Grellscheid@uib.no)

Drop-in:
Høyteksenteret - Datablokken, 407P1



Organisatoriske spørsmål: studieveiler@ii.uib.no

mitt.uib -> Moduler

☰ INF100 > Moduler

Høst 2020

[Hjem](#)

[Kunngjøringer](#)

[Diskusjoner](#)

[Personer](#)

[Sider](#)

[Emneoversikt](#)

Moduler

[Emnets videoer](#)

[Videoseminar](#)

[Digital litteraturliste](#)

[Mine videoer](#)

▼ Uke 1 - Verktøy, interaktiv bruk



[Gruppetime, Oppgaver](#) ↗

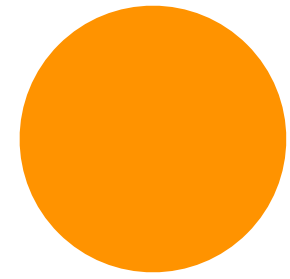
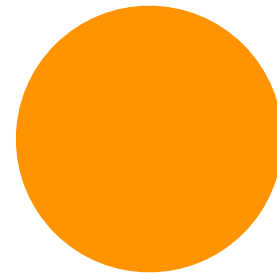
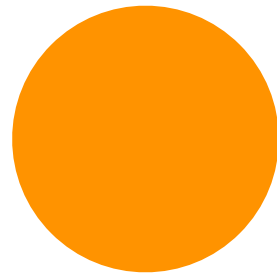
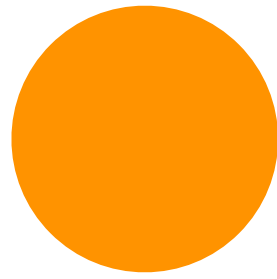
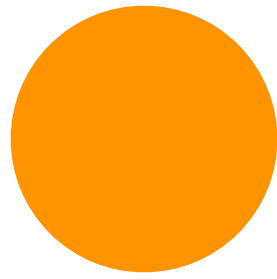
Hva er algoritmer?

Exact Instructions Challenge - THIS is why my kids hate me. | Josh Darnit

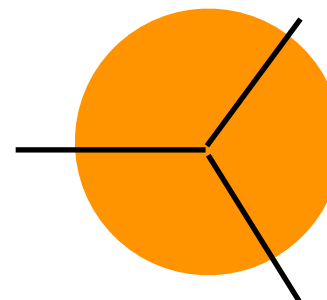
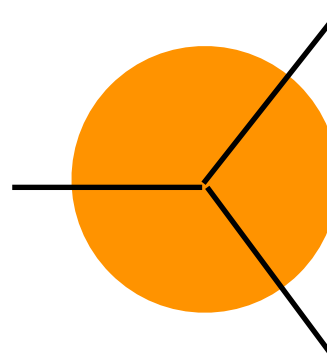
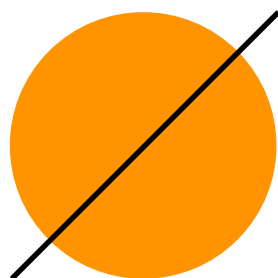
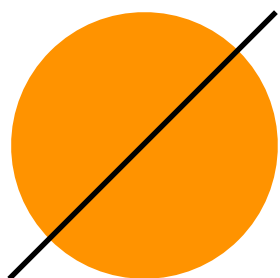
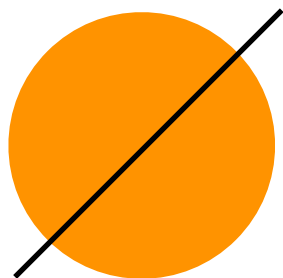




https://en.wikipedia.org/wiki/Rhind_Mathematical_Papyrus



<https://brilliant.org/wiki/egyptian-fractions/#using-egyptian-fractions-to-evenly-divide-an-object-into-equal-shares>



UNIVERSITY OF BERGEN

Kål - geit - ulv - problem



Kilde: Introduction to Computer Science Using Python: A Computational Problem-Solving Focus

Algoritme

En **algoritme** er

- et begrenset antall klart beskrevne, entydige, gjennomførbare steg ...
- som kan følges systematisk ...
- for å produsere et ønsket resultat for gitt input ...
- i en begrenset tidsperiode (det vil si at den til slutt avsluttes).



UNIVERSITY OF BERGEN

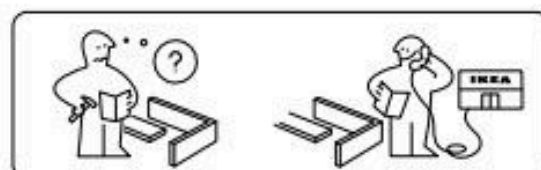
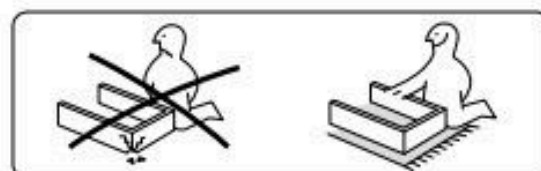
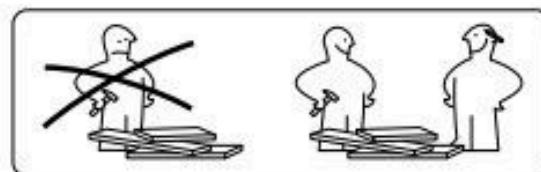
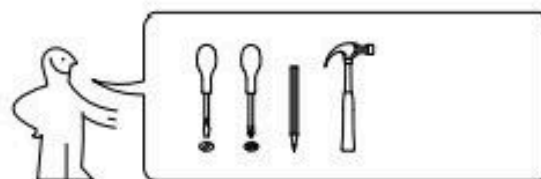
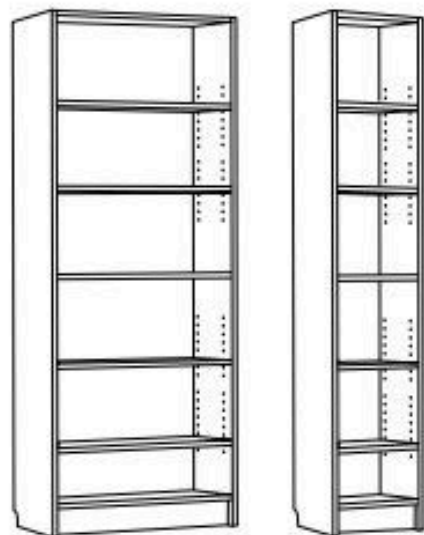
Programmeringsspråk

Hvordan beskriver vi en algoritme slik at maskinen forstår?

Vi trenger programmeringsspråk!

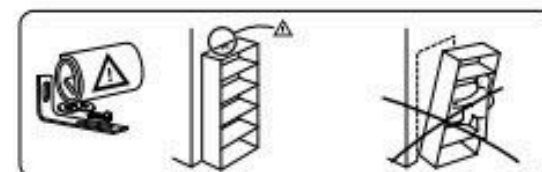


BILLY



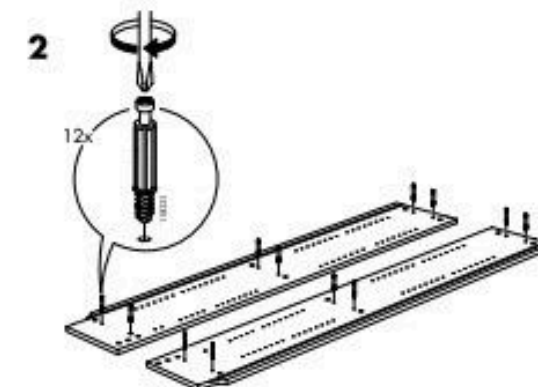
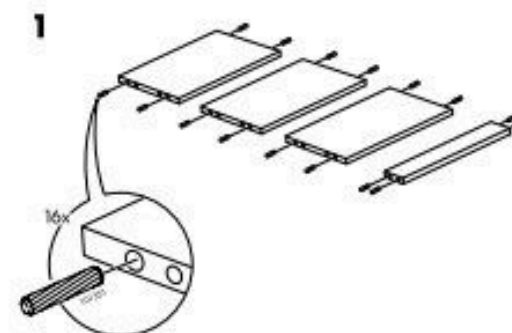
2

ALVENEK



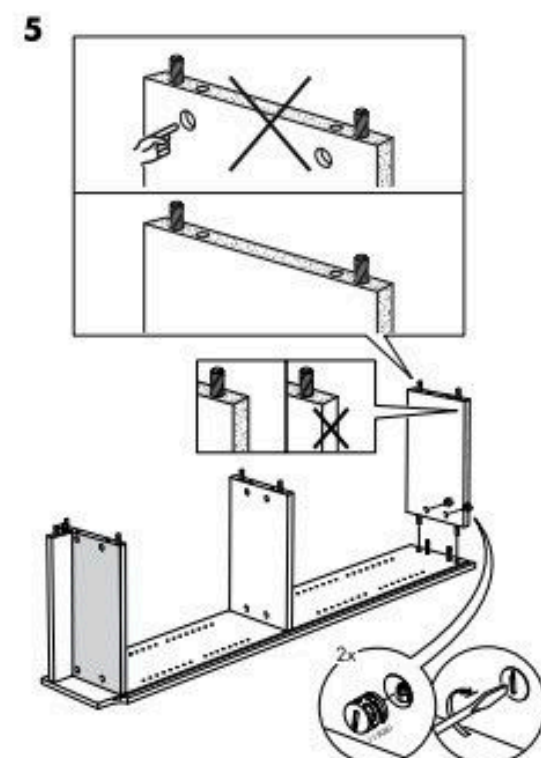
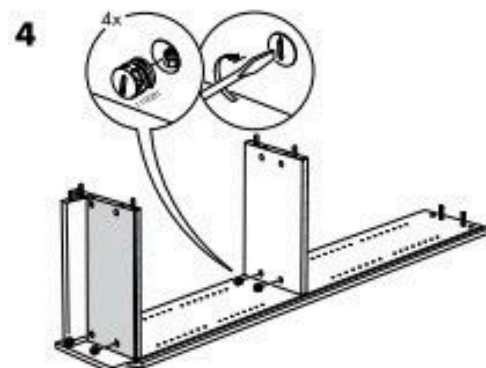
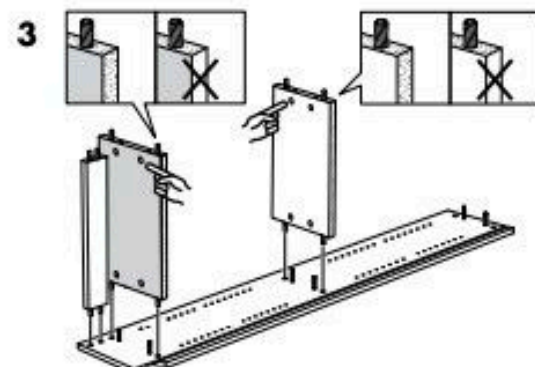
3

ALVENEK



4

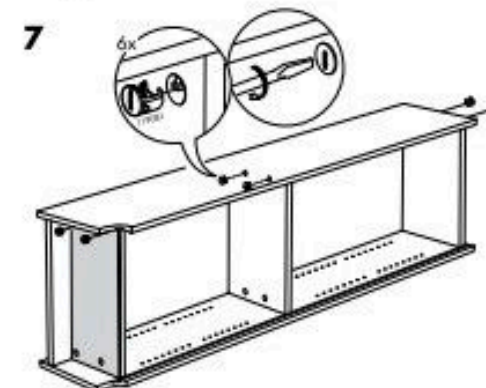
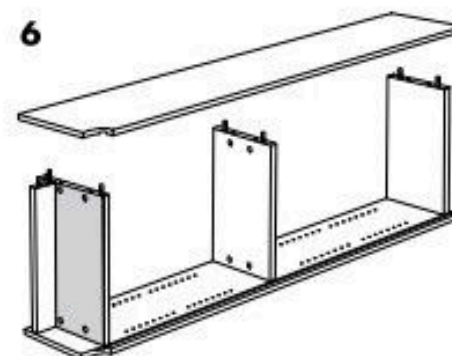
ALVENEK



5

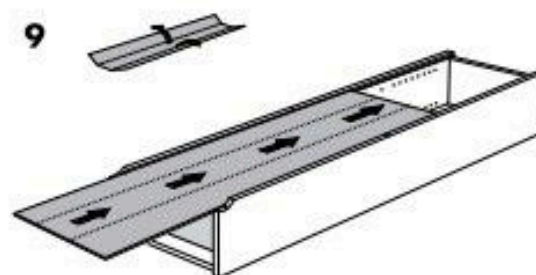
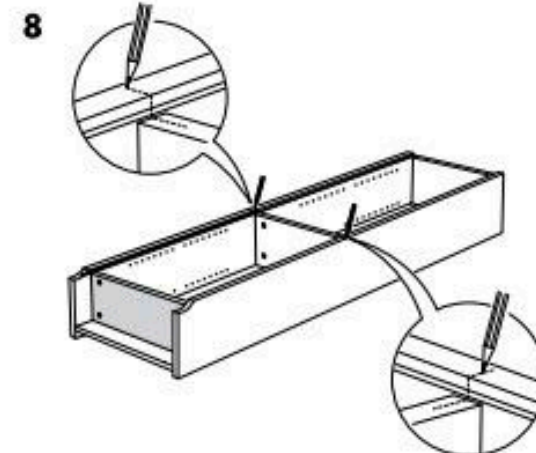
6

ALVENEK



6

7



8

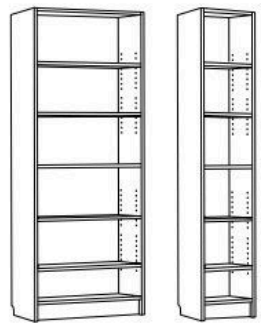
9

7

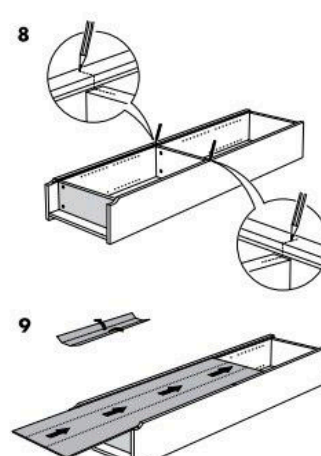
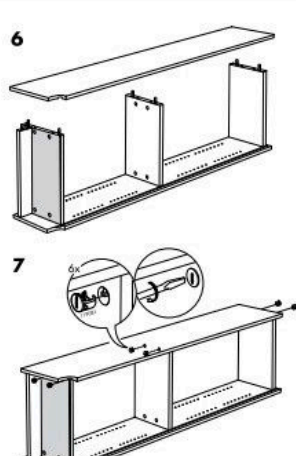
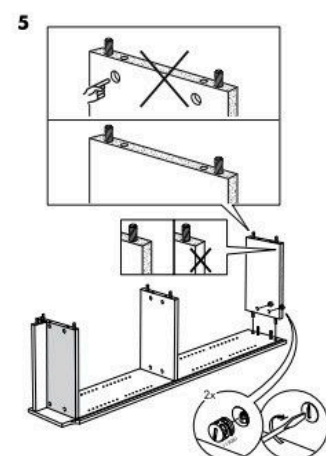
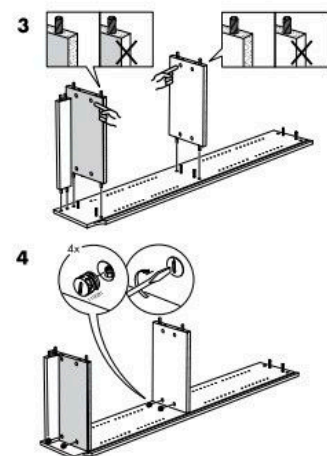
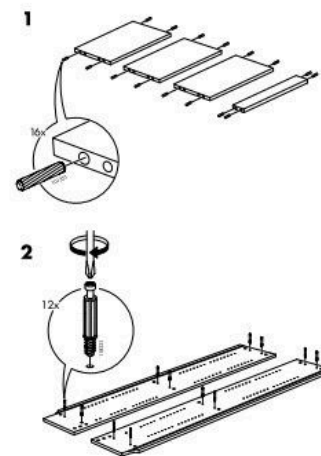
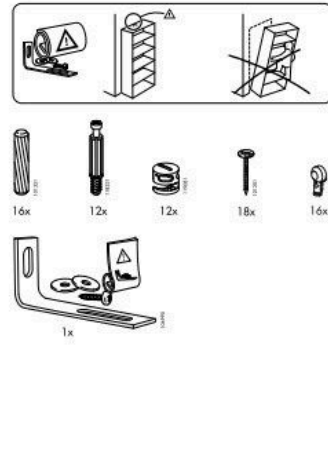
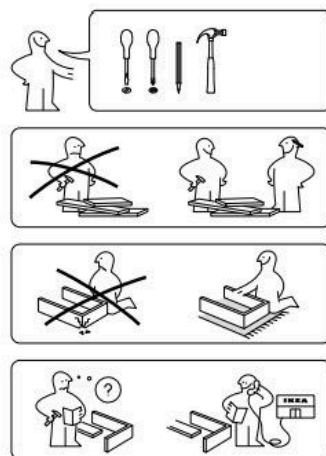
8

ALVENEK

BILLY

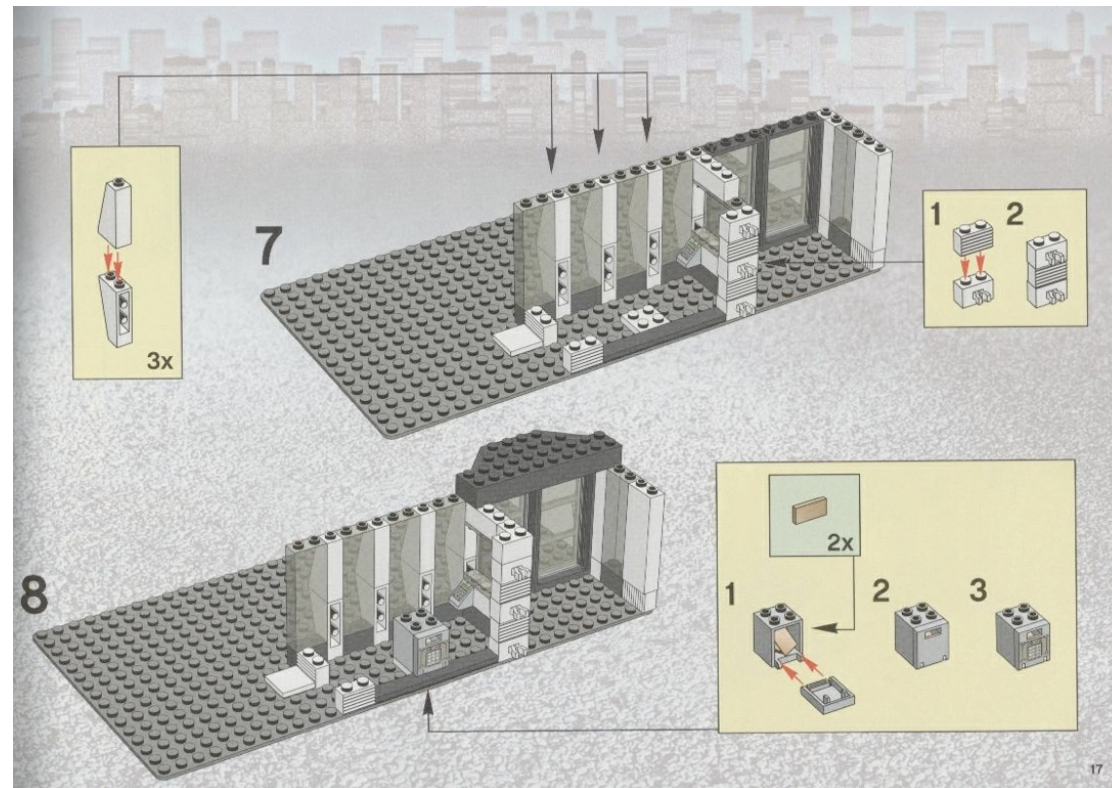


IKKEA
Design and Quality
Since 1946





programmtext





```
10 PRINT CHR$(205.5+RND(1)); : GOTO 10
```

NICK MONTFORT, PATSY BAUDOIN,
JOHN BELL, IAN BOGOST, JEREMY DOUGLASS,
MARK C. MARINO, MICHAEL MATEAS,
CASEY REAS, MARK SAMPLE, NOAH VAWTER

Språk

[illegible]

Språk

Java:

```
class HelloWorld
{
    public static void main(String args[])
    {
        System.out.println("Hello, World");
    }
}
```

Språk

Java:

```
class HelloWorld
{
    public static void main(String args[])
    {
        System.out.println("Hello, World");
    }
}
```

Piet:



Språk

Java:

```
class HelloWorld
{
    public static void main(String args[])
    {
        System.out.println("Hello, World");
    }
}
```

Piet:



brainf*:**

```
+++++++ [ >+++++>+++++>+++<<<- ] >++.>+.+++++
. .+++.>+>.<<+++++>+.>+.+++>+>.
```


Språk

Java:

```
class HelloWorld
{
    public static void main(String args[])
    {
        System.out.println("Hello, World");
    }
}
```

Piet:

**brainf***:**

+++++ [>++++>++++>+++<<<-] >+ . >+ . +++++
 . . +++ . >+ . <<+++++++ . > . +++ . ----- . ----- . >+ .

Python:

```
print("Hello, World")
```

Programmering

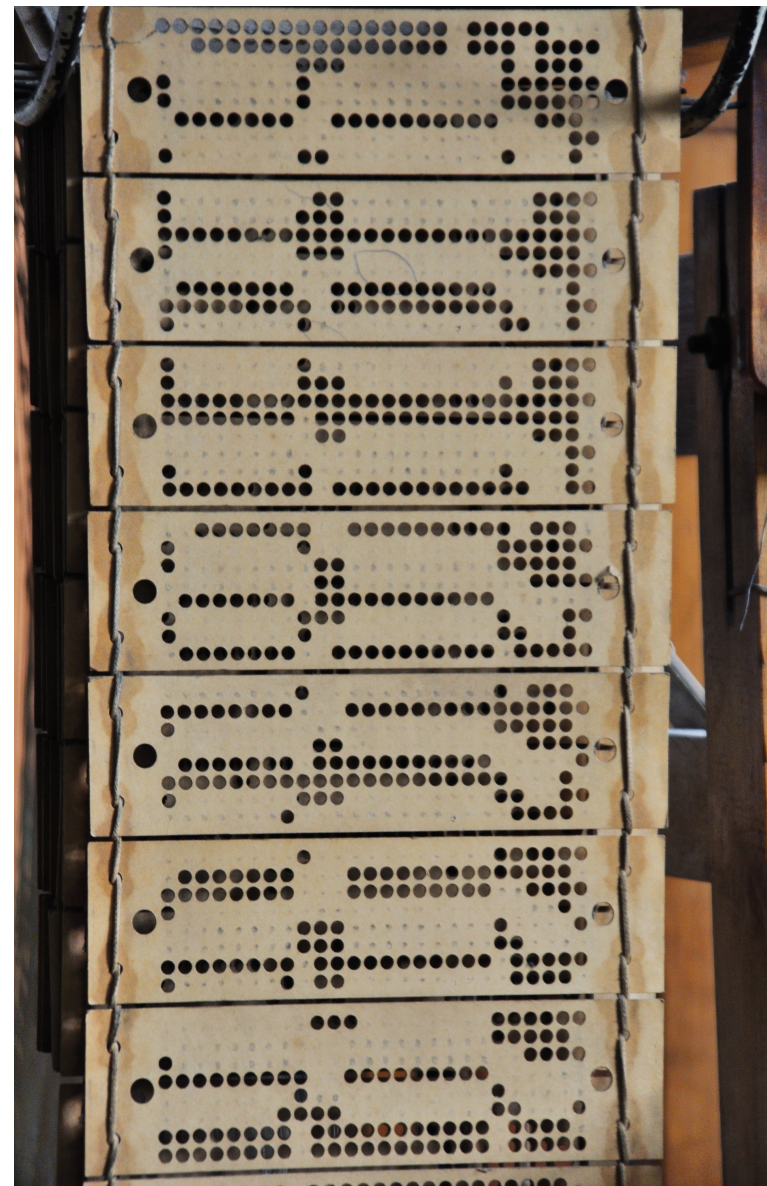
Vanskelig?

Hvor ligger problemet?

- Verktøy: teksteditor, filhåndtering, kjøring
- Språk: regler, strukturer
- Design: omsetning av problemstilling; gode/dårlige løsninger
- Biblioteker man kan bruke

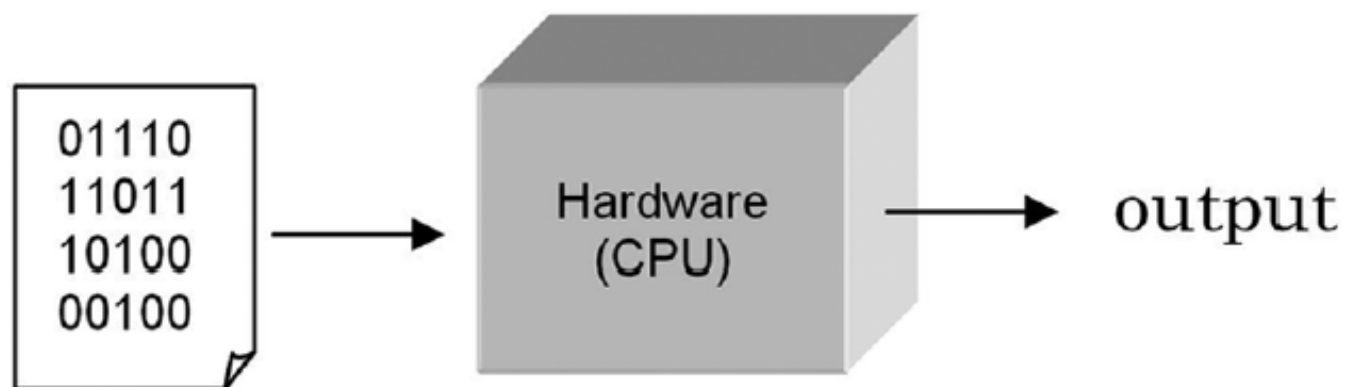
Programming

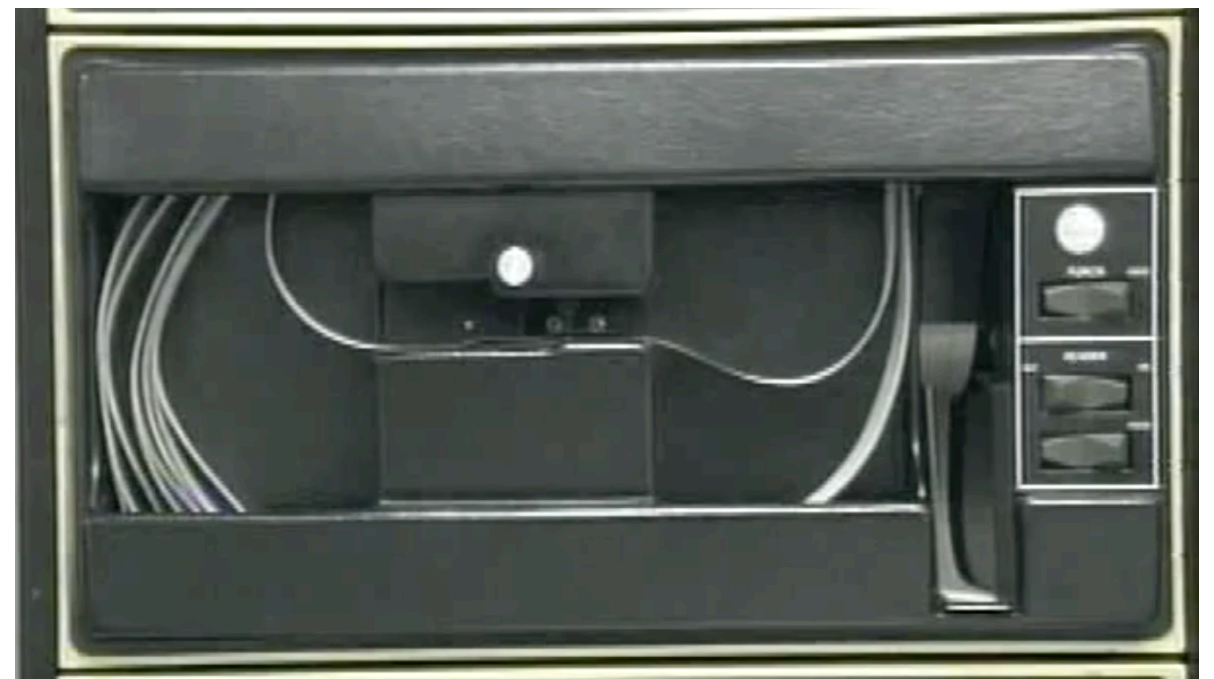
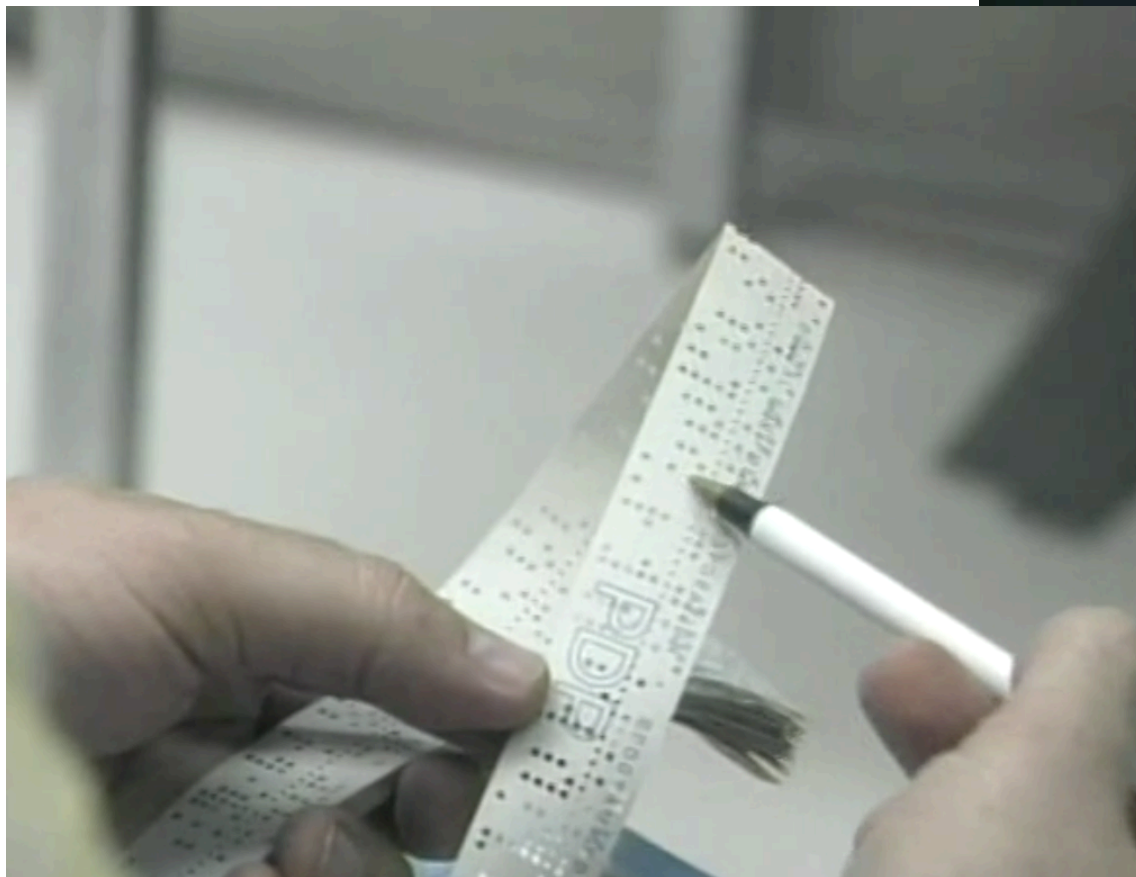
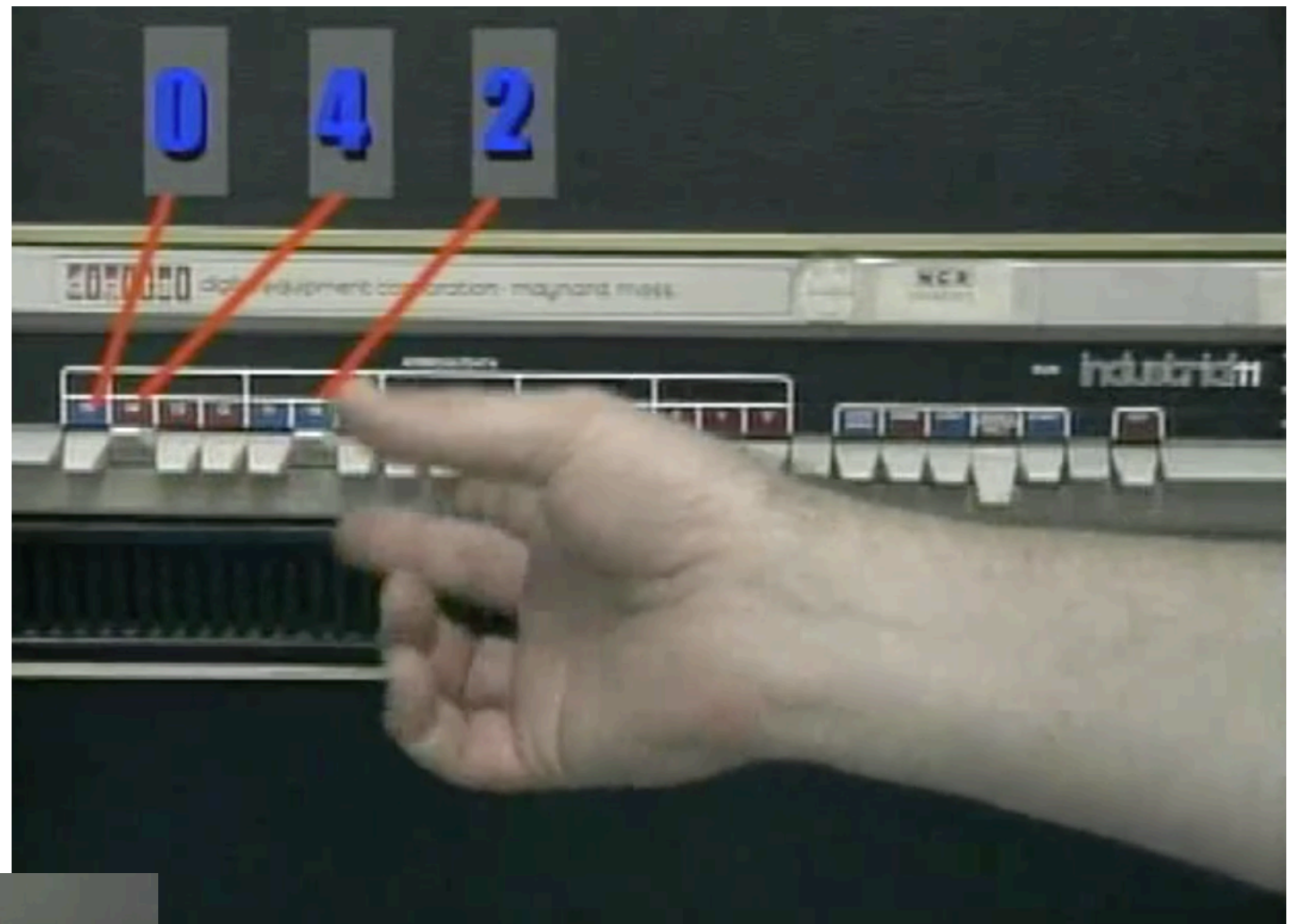
- **Verktøy: editor, filhandtering, kjøring**
- Språk: regler, strukturer
- Design: omsetning av problemstilling; gode/dårlige løsninger
- Biblioteker man kan bruke



Programmkjøring

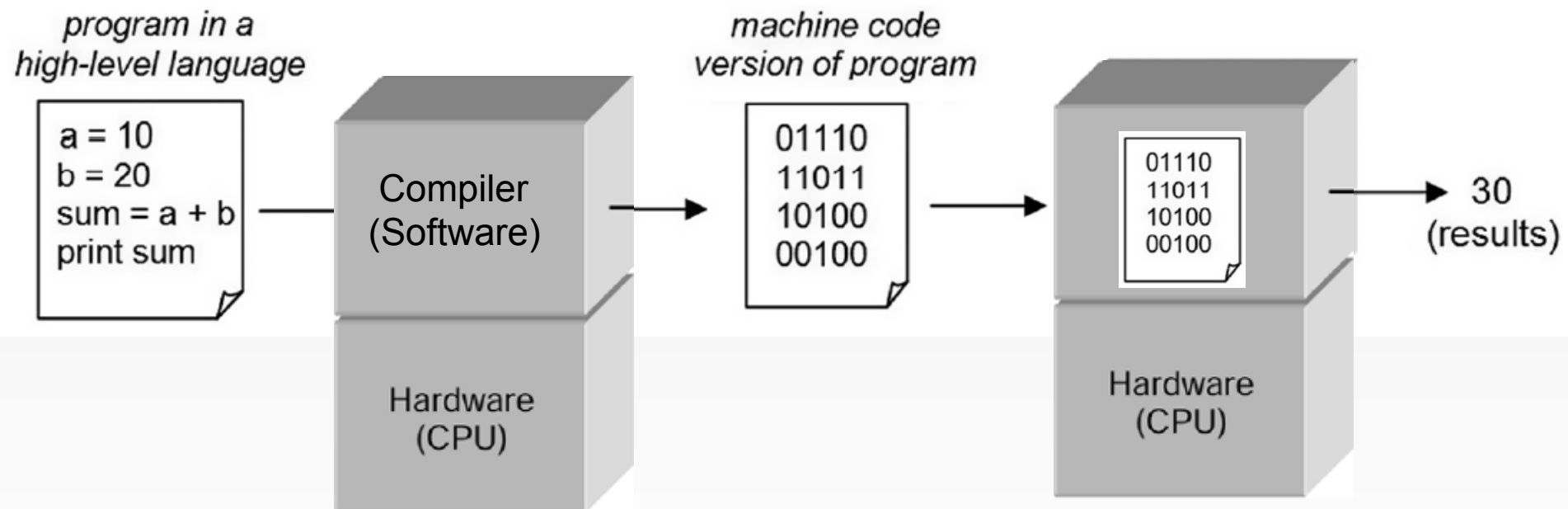
En **central processing unit** (CPU) er designet for å utføre et spesifikt sett med instruksjoner representert i binær form (dvs. 1-ere og 0-ere) kalt **maskinkode**. Bare programmer i maskinkode kan kjøres av en CPU.



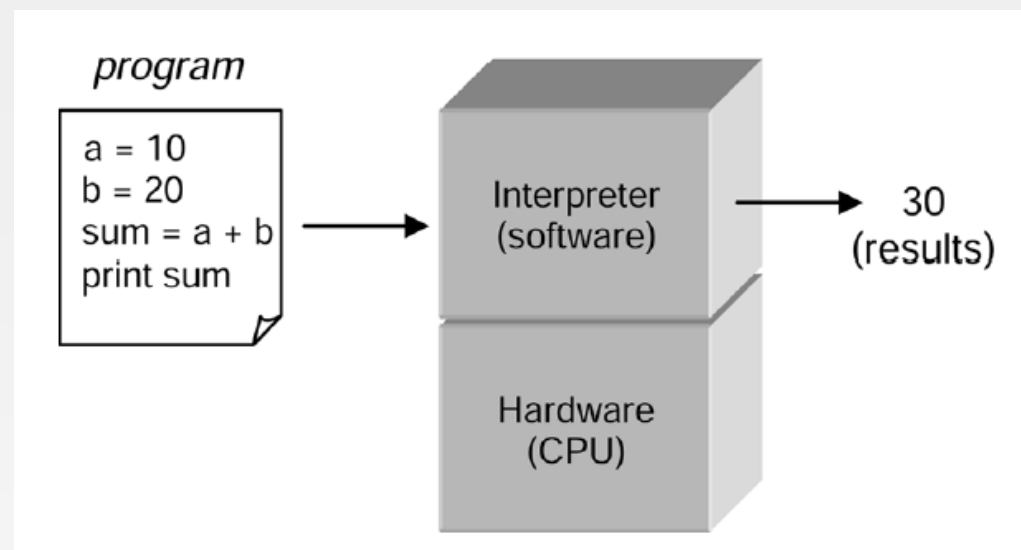


Compiler

Compiler er en programvare som **oversetter programmer** til **maskinkode** som skal kjøres av CPU.



Interpreter



Interpreter er en programvare som **kjører programmer** i stedet for CPU.

En interpreter kan umiddelbart utføre instruksjoner etter hvert som de legges inn. Dette kalles **interaktiv modus**.

Python, som vi skal se, utføres av en interpreter.



Verktøy

- Python 3.10
- VSCode editor
- Installasjon:
<https://inf100.ii.uib.no/notat/installere/>
- browserbasert, uten påmelding for enkle programm:
f.eks. <https://replit.com/lm/python3>

hello.py — code

EXPLORER

CODE

hello.py

Get Started

hello.py

1 print("Hello, World")

2

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

JUPYTER

Python

\$ /opt/local/bin/python /Users/dg/INF100/22H-DIGI111/code/hello.py

Hello, World

\$

OUTLINE

TIMELINE

0 0

Ln 2, Col 1 Spaces: 4 UTF-8 LF Python 3.10.6 64-bit

Programmering

- Verktøy: editor, filhandtering, kjøring
- **Språk: regler, strukturer**
- Design: omsetning av problemstilling; gode/dårlige løsninger
- Biblioteker man kan bruke

Hvorfor Python?

- enkel syntaks, lett å lære
- raskt fremskritt fra skisse til brukbart program
- enkelt å visualisere data
- mest brukt språket i forskning i mange ulike felt
- stort utvalg av nyttige biblioteker

Programming

- Verktøy: editor, filhandtering, kjøring
- Språk: regler, strukturer
- **Design: omsetning av problemstilling; gode/dårlige løsninger**
- Biblioteker man kan bruke

Programmering

- Verktøy: editor, filhandtering, kjøring
- Språk: regler, strukturer
- Design: omsetning av problemstilling; gode/dårlige løsninger
- **Biblioteker man kan bruke**

Første eksempler

- Last ned **digi111-dag-1.zip** fra mitt.uib -> Moduler
- Pakk ut zip-filen i en ny mappe
- åpne mappen i VSCode
- les og kjør filene 01_... til 07_...
- lek med dem, endre på ting, se hva som skjer!

Programmering

- Verktøy: editor, filhandtering, kjøring
- **Språk: regler, strukturer**
- Design: omsetning av problemstilling; gode/dårlige løsninger
- Biblioteker man kan bruke

UNIVERSITY OF BERGEN

Syntaks og semantikk

Slikt som “**naturlige språk**” som norsk og engelsk, er **syntaks** og **semantikk** viktige begreper som finnes i alle programmeringsspråk.



Syntaks

- **Syntaksen** til et språk er et **sett med tegn og de akseptable sekvensene (arrangementene)** av disse tegnene.
- Norsk inkluderer for eksempel bokstavene i alfabetet, tegnsetting og riktig stavede ord og riktige setninger. **Følgende er en syntaktisk korrekt setning på norsk,**
“Hei, hvordan går det med deg?”
- **Det følgende er imidlertid ikke syntaktisk korrekt,**
“Hei, hwortan går det med deg?”
- Bokstavsekvensen “**hwortan**” er ikke et ord på norsk.
- `print("hello ")` ✓ `print("hello` ✗



Semantikk

- **Semantikken** til et språk er **betydningen** knyttet til hver syntaktisk korrekt sekvens av tegn.
- Tenk på følgende setning:
“De hyggelige trærne leser fryktløst.”
- Denne setningen er syntaktisk korrekt, men har ingen betydning.
Dermed er det *semantisk feil*.
- $(\text{num1} + \text{num2} + \text{num3}) / 3 \times 0$
- $(\text{num1} + \text{num2} + \text{num3}) / 2.0$



Setning
(Statement)

a = 7 - 5

Variabel
(navn kan velges fritt)

tilordning
(assignment)

Uttrykk (Expression)
beregnes først

Det finnes andre setninger enn
tilordninger:
if-setninger, for-løkker, funksjoner, ...

Expression / Uttryk	Value / Verdi
12	12
3.56	3.56
"Hei"	"Hei"
13.2 + 17.2 - 2.4	28.0
5**3 - 5**2	100
11 % 3	2
"ABC" + "xyz"	"ABCxyz"

print / input / len

Viktig: gode variabelnavn

```
length = 5.0  
width = 2.0  
height = 1.2  
volume = length * width * height
```

```
a = 5.0  
b = 2.0  
c = 1.2  
d = a * b * c
```

Viktig: gode variabelnavn

```
length = 5.0  
width = 2.0  
height = 1.2  
volume = length * width * height
```

```
a = 5.0  
b = 2.0  
c = 1.2  
d = a * b * c
```

```
bird = 5.0  
yellow = 2.0  
water = 1.2  
duck = yellow * water * bird
```



Betingelser / Conditionals

if elif else

```
n = input("A number: ")
n = int(n)

if n > 100:
    print('Yay!')
    print('You know big numbers')

print('Bye bye')
```


Betingelser / Conditionals

if elif else

```
n = input("A number: ")
n = int(n)

if n > 100:
    print('Yay!')
    print('You know big numbers')

print('Bye bye')
```

```
A number: 7
Bye bye
```

Betingelser / Conditionals

if elif else

```
n = input("A number: ")
n = int(n)

if n > 100:
    print('Yay!')
    print('You know big numbers')

print('Bye bye')
```

```
A number: 7
Bye bye
```

```
A number: 16435
Yay!
You know big numbers
Bye bye
```

Betingelser / Conditionals

if elif else

alle uttryk som har
bool verdi (True/False)
kan stå her

if

True / False

:

4 →

kun hvis True

4 →

kun hvis True

4 →

kun hvis True

resten av programmet

resten av programmet

if

True / False

:

4 →

kun hvis True

4 →

kun hvis True

4 →

kun hvis True

else:

4 →

kun hvis False

4 →

kun hvis False

4 →

kun hvis False

resten av programmet

resten av programmet

if

True / False

■
■

```
4 n = input("A number: ")
  n = int(n)
4
4 if n > 100:
4     print('Yay!')
    print('You know big numbers')
else:
    print("That's quite small!")
4 print('Bye bye')
```

4 → kun hvis False

4 → kun hvis False

resten av programmet

resten av programmet

if

True / False

■
■

```
4 n = input("A number: ")
  n = int(n)
4
  if n > 100:
4     print('Yay!')
      print('You know big numbers')
  else:
      print("That's quite small!")
4 print('Bye bye')
```

```
4
4 A number: 7
  That's quite small!
  Bye bye
```

resten av programmer

if

True / False

1

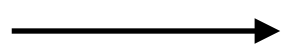
⋮

4



kun hvis True₁

4



kun hvis True₁

elif

True / False

2

⋮

4



kun hvis False₁ True₂

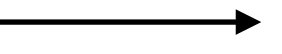
elif

True / False

3

⋮

4



kun hvis False_{1,2} True₃

else:

4



kun hvis alle False

```
tekst = 'Det var en gang...'
```

```
if 'a' in tekst:  
    print('Aaaah!')
```

```
elif ' ' in tekst:  
    print('Space')
```

```
elif 'z' in tekst:  
    print('Zzz Zzz')
```

```
elif '.' in tekst:  
    print('Dots...')
```

```
tekst = 'Det var en gang...'
```

```
if 'a' in tekst:  
    print('Aaaah!')
```

```
if ' ' in tekst:  
    print('Space')
```

```
if 'z' in tekst:  
    print('Zzz Zzz')
```

```
if '.' in tekst:  
    print('Dots...')
```

```
tekst = 'Det var en gang...'
```

```
if 'a' in tekst:  
    print('Aaaah!')
```

```
elif ' ' in tekst:  
    print('Space')
```

```
elif 'z' in tekst:  
    print('Zzz Zzz')
```

```
elif '.' in tekst:  
    print('Dots...')
```

Aaaah!

```
tekst = 'Det var en gang...'
```

```
if 'a' in tekst:  
    print('Aaaah!')
```

```
if ' ' in tekst:  
    print('Space')
```

```
if 'z' in tekst:  
    print('Zzz Zzz')
```

```
if '.' in tekst:  
    print('Dots...')
```

Aaaah!
Space
Dots...