



in1000 uke 1 dypdykk

viktig forståelse fra forrige ukes pensum

læremål



0

kodeflyt

1

tilordninger

2

beslutninger

-

diverse

skrive til/lese fra terminal
lese og forstå feilmeldinger



0

kodeflyt

bygge opp en god mental modell
av python-programmer

kodeforståelse

problem-nivå

hvordan programmet mitt er en modell
av noe ute i den virkelige verden

kode-nivå

programmets syntaks og struktur
(hvert programmeringsspråk gjør dette forskjellig)

maskin-nivå

hva skjer inne i maskinen når jeg kjører
programmet mitt?



problem-nivå

et dataprogram er en modell av et problem ute i den virkelige verden som vi forsøker å løse



```
1 # Hvor mye av hver ting?
2 antall_epler = 2
3 antall_pærer = 1
4
5 # Hva koster hver ting?
6 pris_epler = 5
7 pris_pærer = 6
8
9 # Regn ut hva vi betaler
10 sum_epler = antall_epler*pris_epler
11 sum_pærer = antall_pærer*pris_pærer
12
13 # Skriv ut totalt beløp
14 sum = sum_epler + sum_pærer
15 print(sum)
```

- hva slags problem er dette programmet laget for å løse?
- **variabelnavn** er viktige: «hva betyr disse verdiene?»
- **kommentarer** er viktige: gir oss mer informasjon enn vi klarer å lese ut fra koden alene
- sammenlign med et annet program:

hva er likt?

hva er forskjellig?

frukt.py > ...

```
1 # Hvor mye av hver ting?
2 antall_epler = 2
3 antall_pærer = 1
4
5 # Hva koster hver ting?
6 pris_epler = 5
7 pris_pærer = 6
8
9 # Regn ut hva vi betaler
10 sum_epler = antall_epler*pris_epler
11 sum_pærer = antall_pærer*pris_pærer
12
13 # Skriv ut totalt beløp
14 sum = sum_epler + sum_pærer
15 print(sum)
```

energi.py > ...

```
1 # Hvor mange timer i bruk?
2 timer_elbil = 2
3 timer_dusj = 1
4
5 # Effekt (i kW)
6 kW_elbil = 5
7 kW_dusj = 6
8
9 # Regn ut forbruk (i kWh)
10 kWh_elbil = timer_elbil*kW_elbil
11 kWh_dusj = timer_dusj*kW_dusj
12
13 # Skriv ut totalt forbruk (i kWh)
14 kWh_sum = kWh_elbil + kWh_dusj
15 print(kWh_sum)
```

kode-nivå

et dataprogram består av variabler, verdier, kommentarer, prosedyrer osv. som må skrives på en bestemt måte for at maskinen skal kunne tolke det riktig



frukt.py > ...

```
1 # Hvor mye av hver ting?
```

kommentar

```
2 antall_epler = 2
```

```
3 antall_pærer = 1
```

tilordning

```
4  
5 # Hva koster hver ting?
```

```
6 pris_epler = 5
```

```
7 pris_pærer = 6
```

```
8
```

luft («whitespace»)

```
9 # Regn ut hva vi betaler
```

```
10 sum_epler = antall_epler*pris_epler
```

```
11 sum_pærer = antall_pærer*pris_pærer
```

tilordning

```
12  
13 # Skriv ut totalt beløp
```

```
14 sum = sum_epler + sum_pærer
```

```
15 print(sum)
```

funksjon (prosedyre)

maskin-nivå

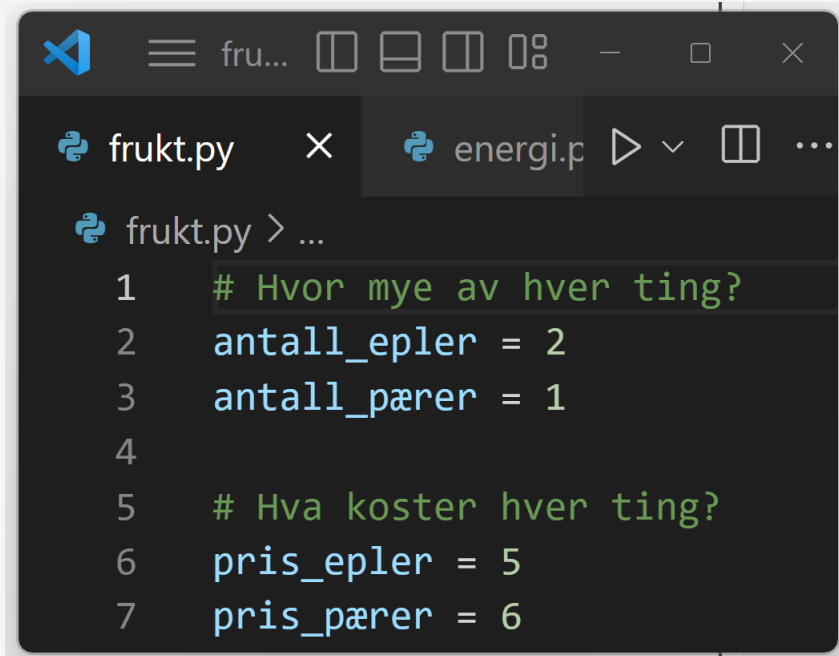
for å kunne lese og forstå et dataprogram må vi ha en mental modell av hvordan det kjører i maskinen og hvilke tilstander minnet på maskinen er i



sentrale verktøy

editor (VS Code)

her skriver vi programkoden



The image shows a screenshot of the Visual Studio Code (VS Code) editor interface. The window title bar at the top displays the VS Code logo, a hamburger menu, and the text 'fru...'. Below the title bar, there are two open files: 'frukt.py' and 'energi.p'. The 'frukt.py' file is active, and its content is visible in the editor area. The code is written in Python and includes comments in Norwegian. The code defines variables for the number of apples and pears, and their respective prices. The line numbers 1 through 7 are visible on the left side of the editor.

```
frukt.py > ...
1  # Hvor mye av hver ting?
2  antall_epler = 2
3  antall_pærer = 1
4
5  # Hva koster hver ting?
6  pris_epler = 5
7  pris_pærer = 6
```

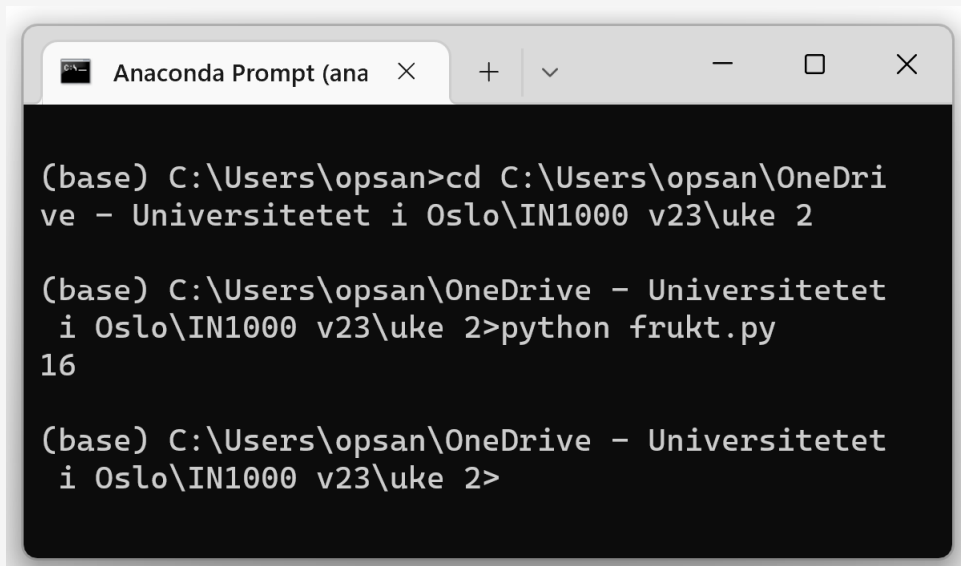
sentrale verktøy

editor (VS Code)

her skriver vi programkoden

terminal

her kjører vi programkoden



```
Anaconda Prompt (ana) x + v - □ x
(base) C:\Users\opsan>cd C:\Users\opsan\OneDrive - Universitetet i Oslo\IN1000 v23\uke 2
(base) C:\Users\opsan\OneDrive - Universitetet i Oslo\IN1000 v23\uke 2>python frukt.py
16
(base) C:\Users\opsan\OneDrive - Universitetet i Oslo\IN1000 v23\uke 2>
```

sentrale verktøy

editor (VS Code)

her skriver vi programkoden

terminal

her kjører vi programkoden

debugger ([python tutor](#))

her går vi gjennom programmet steg for steg slik at vi kan forstå hva som skjer

kan også brukes til å finne feil (“bugs”)

Python 3.6
[known limitations](#)

```
1 # Hvor mye av hver ting?
2 antall_epler = 2
3 antall_pærer = 1
4
5 # Hva koster hver ting?
6 pris_epler = 5
7 pris_pærer = 6
8
9 # Regn ut hva vi betaler
10 sum_epler = antall_epler*pris_epler
11 sum_pærer = antall_pærer*pris_pærer
12
13 # Skriv ut totalt beløp
14 sum = sum_epler + sum_pærer
15 print(sum)
```

Print output (drag lower right)

Frames

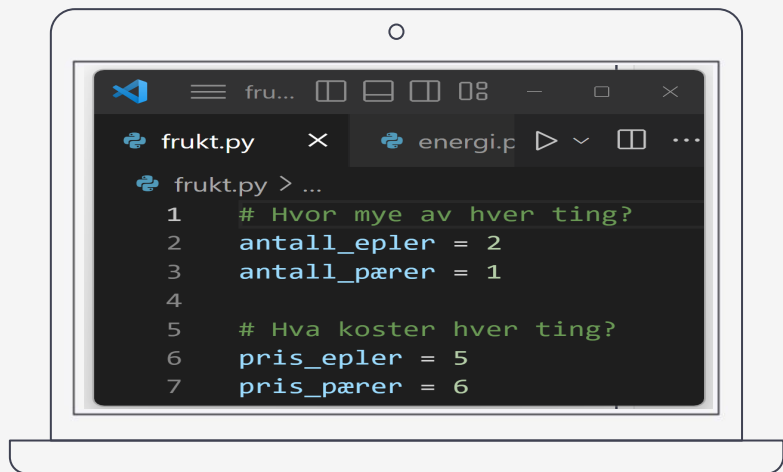
Global frame	
antall_epler	2
antall_pærer	1
pris_epler	5
pris_pærer	6
sum_epler	10
sum_pærer	6
sum	16



1

tilordninger

forstå hvordan vi tar vare på
verdier med variabler



live-koding

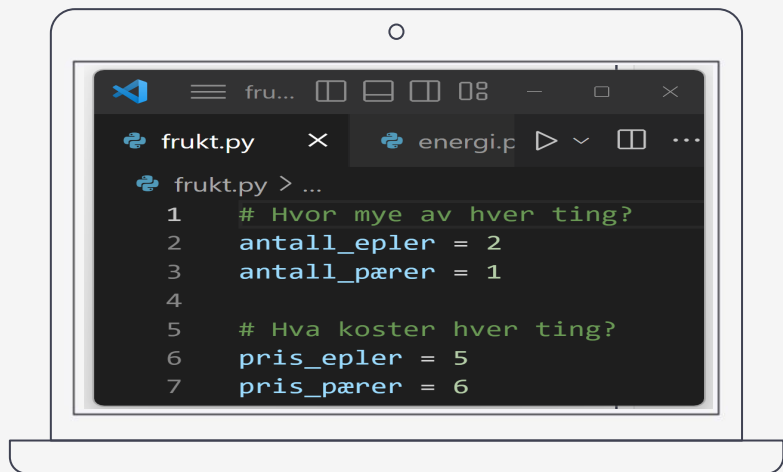
livecode-1.py
(se timeplan på emnesiden)



2

beslutninger

kunne bruke if/elif/else til å avgjøre
hvilke kodelinjer som kjøres



live-koding

livecode-2.py
(se timeplan på emnesiden)

elif / else kjører **bare** hvis
tidligere test(er) **feiler**



sammenlign kodesnuttene: hva gjør er likt/forskjellig?

```
alder_tekst = input("Din alder :")

# konverterer til heltall
alder = int(alder_tekst)

if (alder < 18) or (alder > 68):
    print("Rabatt")
else:
    print("Ikke rabatt")
```

```
alder_tekst = input("Din alder :")

# konverterer til heltall
alder = int(alder_tekst)

if (alder > 18) and (alder < 68):
    print("Ikke rabatt")
else:
    print("Rabatt")
```

mentimeter:

[menti.com](https://www.menti.com)

kode: 6988 1433





in1000

uke 2

nytt lærestoff

datatyper, uttrykk, kodeflyt, prosedyrer

læremål



0

kodeflyt

1

evaluering av uttrykk

2

datatyper

3

prosedyrer



0

kodeflyt

vi fortsetter å se på kodeflyt for
det som er nytt i dag



1 uttrykk

forstå, lese og skrive uttrykk for å
evaluere verdier

frukt.py > ...

```
1 # Hvor mye av hver ting?
2 antall_epler = 2
3 antall_pærer = 1
4
5 # Hva koster hver ting?
6 pris_epler = 5
7 pris_pærer = 6
8
9 # Regn ut hva vi betaler
10 sum_epler = antall_epler*pris_epler
11 sum_pærer = antall_pærer*pris_pærer
12
13 # Skriv ut totalt beløp
14 sum = sum_epler + sum_pærer
15 print(sum)
```



Hva er venstre side av = ?
Hva er høyre side av = ?

frukt.py > ...

```
1 # Hvor mye av hver ting?
2 antall_epler = 2
3 antall_pærer = 1
4
5 # Hva koster hver ting?
6 pris_epler = 5
7 pris_pærer = 6
8
9 # Regn ut hva vi betaler
10 sum_epler = antall_epler*pris_epler
11 sum_pærer =
12
13 # Skriv ut totalt beløp
14 sum = sum_epler + sum_pærer
15 print(sum)
```

Venstre side: Kun 1 variabel
Denne får nå en (ny) verdi

frukt.py > ...

```
1 # Hvor mye av hver ting?
2 antall_epler = 2
3 antall_pærer = 1
4
5 # Hva koster hver ting?
6 pris_epler = 5
7 pris_pærer = 6
8
9 # Regn ut hva vi betaler
10 sum_epler = antall_epler*pris_epler
11 sum_pærer = antall_pærer*pris_pærer
12
13 # Skriv ut totalt beløp
14 sum = sum_epler + sum_pærer
15 print(sum)
```



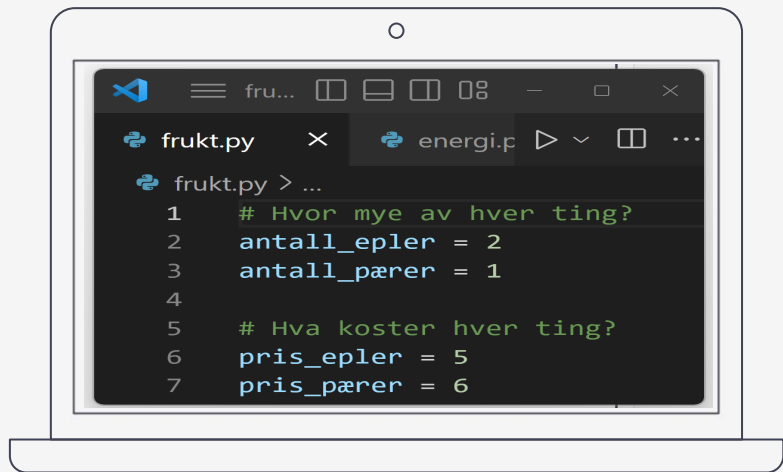
Høyre side: En eller flere verdier (akkurat her kommer begge fra andre variabler) til utregning av ny verdi for variabelen på venstre side



2

datatyper

forstå, lese og skrive uttrykk for å
evaluere verdier



live-koding

(utsatt til repetisjonstimen mandag!)



3

prosedyrer

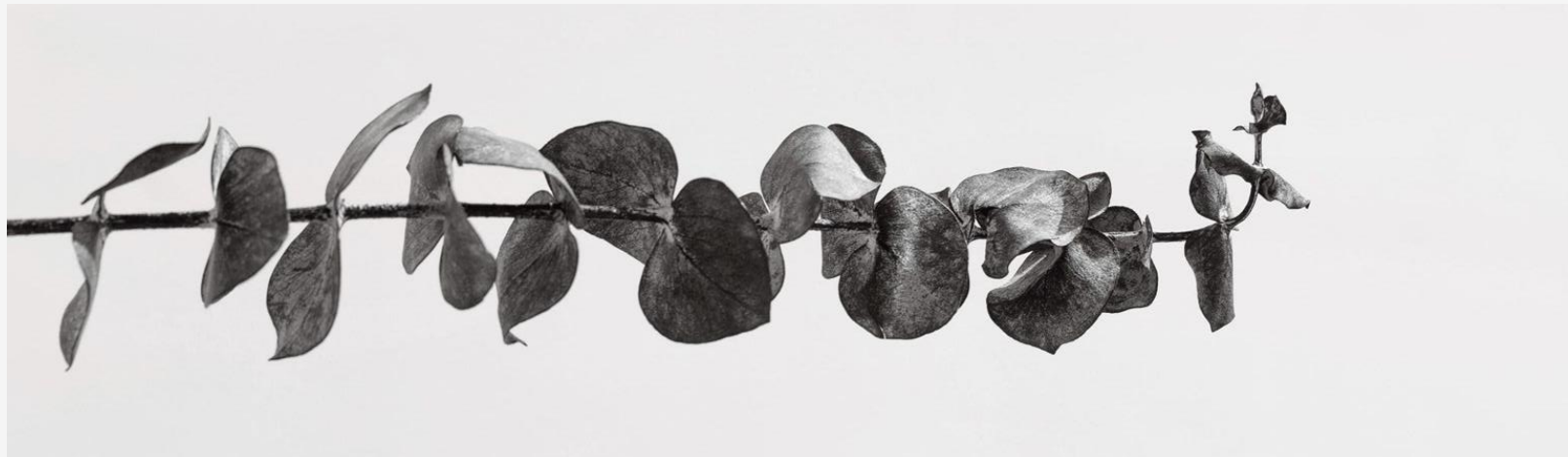
forstå og bruke prosedyrer til å
organisere programmer



537142195813



191419452023



191419452023



høyt abstraksjonsnivå

se på flere linjer kode som en
helhet med en bestemt oppgave

lavt abstraksjonsnivå

se på **detaljene**
i hver enkelt linje med kode



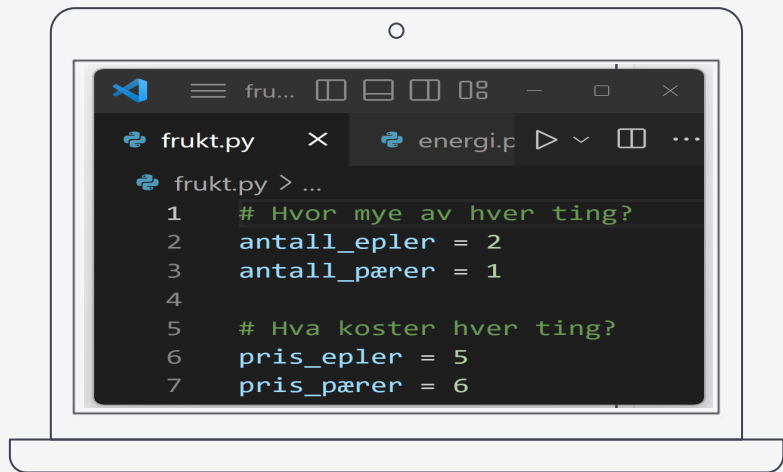
sammenlign kodesnuttene:

```
print("Her kommer en liten hilsen:")

time = int(input("Hva er klokka? "))
if time < 6:
    print("Sov godt!")
elif time < 10:
    print("God morgen!")
elif time < 12:
    print("God formiddag!")
elif time < 19:
    print("God kveld!")
else:
    print("God natt!")

print("Husk å ta godt vare på deg selv!")
```

```
print("Her kommer en liten hilsen:")
hilsen()
print("Husk å ta godt vare på deg selv!")
```



live-koding

(utsatt til repetisjonstimen mandag!)

