

TDT4110 Informasjonsteknologi, grunnkurs

Uke 35

Introduksjon til programmering i Python

Professor Guttorm Sindre
Institutt for datateknikk og informasjonsvitenskap

Læringsmål og pensum



- Mål
 - Vite hva et program er
 - Ha kjennskap til hvordan programmer designes
 - Kunne bruke Python
 - programmeringsomgivelse, skript, prompt, variabler
- Pensum for dagens forelesning
 - Starting out with Python, chapter 1, 2.1, 2.7

Hva er et program?

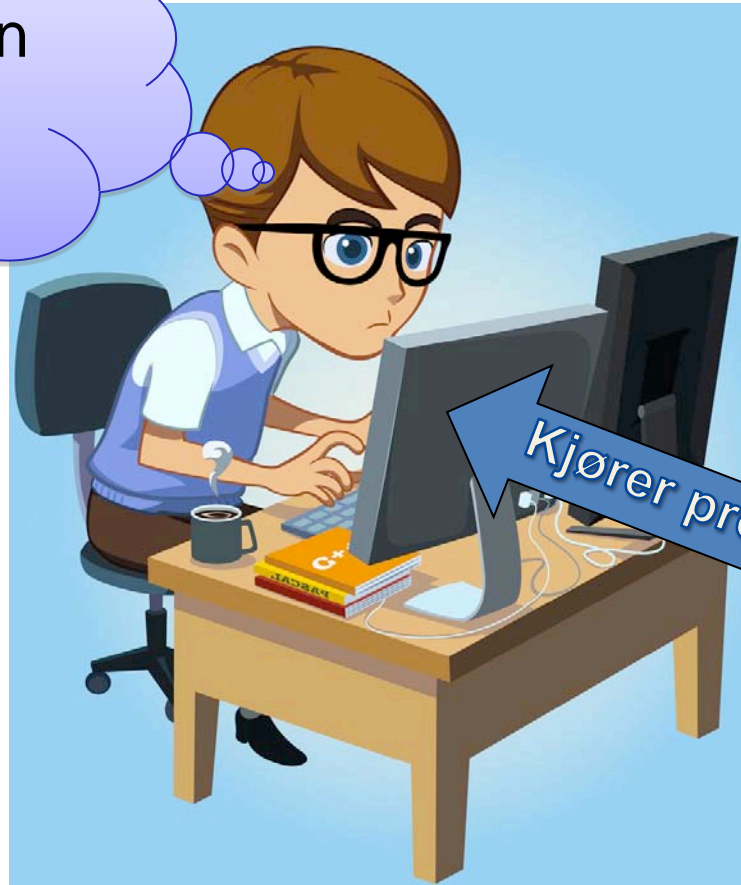


- Program: en oppskrift
 - instruksjoner som forteller en datamaskin hva den skal gjøre
- Et program kan bestå av instruksjoner som:
 - Oppretter (deklarerer) og gir verdier til (tilordner) variable
 - Evaluerer og regner på variable
 - Gjør valg basert på betingelser
 - Repeterer samme handling mange ganger
 - Utfører ulike operasjoner
 - f.eks. viser grafikk, spiller av lyd
 - tar imot data fra brukeren
 - tar imot data fra sensorer
- Før en datamaskin kan gjøre et program, må det oversettes til maskinkode

Hva er et program?

Skriv program

Skriv summen
av 123 + 321
på skjermen



```
x = 123  
y = 321  
sum = x + y  
print(sum)
```

Python

↙
Oversetter

```
0100101010101010  
1010101001111010  
1010101010101010  
1010101010001010  
101010101110101  
010101011010
```

kode
Maskin

↘
Kjører prog

Hva er programmering?



- Programmering == problemløsning
 - Arbeidsprosessen man gjennomgår når man skriver et program
- Forstå problemet
 - Hva slags resultat trenger brukeren? (output)
 - Hva slags data har vi tilgjengelig i utgangspunktet?
 - Hvordan kan vi komme fra input til output?
- Finne en løsning, uttrykke den i Python
 - Noen problemer kan ha mange mulige løsninger som alle virker
 - Noen løsninger kan likevel være bedre enn andre
 - bruker kortere tid eller mindre strøm
 - kortere / enklere å skrive, eller enklere å forstå
 - Mer fleksible for fremtidige endringer
 - Mindre sårbare for hacking

Installering av Python

- Python kan lastes ned fra www.python.org/download
- Velg versjon **3.5.2** av Python og riktig operativsystem (Windows, Mac OS X eller Linux)
 - Gjør en standard installasjon
- Start Python ved å klikke på IDLE i Python katalogen
 - Windows fra Start-menyen
 - Mac fra Application/Programmer
- Kan også kjøre Python fra et unix/dos-skall ved å skrive kommandoen “python”
 - Sjekk at du kjører Python 3.5, ikke Python 2.7!

Programmering i Python



- To måter å programmere på
 - Interaktiv modus:
 - Skriv en og en instruks i en tolker i Python-programmet
 - Eget til kjapp utprøving av nye ting
 - Uegnet ellers siden man må skrive koden om igjen
 - Lagre programmer på fil
 - Skriv tekstfil med Python-kommandoer
 - Bruk Python-editoren (f.eks IDLE) til å
 - oversette ditt program til maskinkode
 - kjøre programmet ditt
 - Eget til større programmer, nyttige programmer
 - Kjapp utprøving blir litt mindre kjapp...

Interaktiv modus



- Startes ved å åpne IDLE eller starte “python”
 - Programmeringen foregår i et såkalt “shell” (skall)
- Skriv inn en programlinje og få direkte respons
 - `>>>` viser hvor du kan skrive kommandoer
- Får ut
 - resultat av beregning hvis det gikk bra
 - feilmelding hvis du har skrevet noe feil

Aritmetiske uttrykk i Python

Forskjeller fra matematisk notasjon

- Desimalpunktum: `5.82` (ikke `5,82`)
- De fire regneartene : `+` `-` `*` `/` (ikke `.` `:`)
 - `*` må alltid skrives... `5*a + 4*b + a*b` (ikke `5a + 4b + ab`)
- Heltallsdivisjon `//`, rest (modulo) `%`
- Eksponentiering (potens): `**`, eks 5^4 angis `5**4`
- Som i matte: parenteser gir rekkefølge på utregning
 - Hvis vi ønsker annen rekkefølge enn normal presedens
 - Men kun `()` –parenteser, ikke `{.....[...(...)(...)]}`
 - Kan trenge mer parenteser enn i matte
 - F.eks. hvor matematisk uttrykk har brøkstreker, potenser, rottegn

Tilordning og variable i Python

Forskjell fra matematiske likninger og regneark



- I matte kan symboler brukes helt abstrakt
 - F.eks. $(a+b)^2 = a^2 + 2ab + b^2$
 - Ikke i Python, variable må ha en definert verdi før de brukes
 - Variabel: Huske en verdi til senere bruk
- = i Python betyr ikke matematisk likhet men tilordning
 - Det er en ordre (imperativ setning): "Gi variabelen på venstre side verdien av uttrykket på høyre side"
 - Matematisk likning er derimot en påstand (fortellende setning)
- = i Python ligner mer på = i regneark
 - Men Python utføres linje for linje, ikke en permanent relasjon
- Likhet i Python testes med ==
 - Men dette er også en ordre: "Sjekk om venstre side er lik høyre side", ikke en påstand om at det nødvendigvis er tilfelle

Lage Python programmer



- Skrive et program i en teksteditor
 - Men ikke Word e.l., bruk en editor ment for programmering
- Lagre dokumentet med filtype .py
- Kjør programmet ved hjelp av python-tolker
- I IDLE er det en teksteditor for Python-programmer
 - Velg File / New Window
- For å kjøre programmet i IDLE:
 - Velg Run / Run Module

Kort intro til Python



- Vanlig: en programsetning per linje
 - Hvis setninga er så lang at flere linjer trengs:
 - Bruk () rundt kommandoen, eller \ der linja må brekkes
- Variable tar vare på informasjon: `x = 5.4`
- Kan lagre tekst: `tekst = 'Hei'`
- Kan teste logiske uttrykk: `x > 3.14`
- Har innebygde funksjoner, f.eks.
 - Absoluttverdi `abs(x)`
 - Avrunding `round(x)`

Hvordan et program fungerer



- Programmer utføres linje for linje
 - altså ei linje av gangen
 - ...og linjer utføres operasjon for operasjon
 - viktig å forstå etter hvert som vi lanserer avanserte programmer
- ”programpeker” flytter seg nedover linje for linje
 - Det som står øverst skjer først!
- Det fins også parallelle programmer
 - Utfører flere programsetninger samtidig
 - Ikke tema for dette kurset, kan læres senere
 - TDT4200 Parallelle beregninger

Illustrasjon av programkjøring



```
pi = 3.14
r = 5.7
O = 2 * pi * r
A_sirkel = pi * r ** 2
h = 25.1
A_sylinder = O * h + 2 * A_sirkel
print('Areal av sylinderen:', A_sylinder)
```

Illustrasjon av programkjøring



```
pi = 3.14
r = 5.7
O = 2 * pi * r
A_sirkel = pi * r ** 2
h = 25.1
A_sylinder = O * h + 2 * A_sirkel
print('Areal av sylinderen:', A_sylinder)
```

Illustrasjon av programkjøring



```
pi = 3.14
r = 5.7
O = 2 * pi * r
A_sirkel = pi * r ** 2
h = 25.1
A_sylinder = O * h + 2 * A_sirkel
print('Areal av sylinderen:', A_sylinder)
```


Illustrasjon av programkjøring



```
pi = 3.14
r = 5.
O = 2 * pi * r
A_sirkel = pi * r ** 2
h = 25.1
A_sylinder = O * h + 2 * A_sirkel
print('Areal av sylinderen:', A_sylinder)
```




Illustrasjon av programkjøring



```
pi = 3.14
r = 5.7
O = 2 * pi * r
A_si 6.28 = pi * r ** 2
h = 25.1
A_sylinder = O * h + 2 * A_sirkel
print('Areal av sylinderen:', A_sylinder)
```



Illustrasjon av programkjøring



```
pi = 3.14
r = 5.7
O = 2 * pi * r
A_sirkel = 35.796 * pi * r ** 2
h = 25.1
A_sylinder = O * h + 2 * A_sirkel
print('Areal av sylinderen:', A_sylinder)
```

The code block is highlighted in yellow. A green arrow points down to the variable `pi` in the first line. A blue oval highlights the expression `2 * pi * r` in the third line, with a blue arrow pointing from the oval to the value `35.796` in the fourth line. The value `35.796` is also highlighted with a white background.

Illustrasjon av programkjøring



```
pi = 3.14
r = 5.7
O = 2 * pi * r
A_sirkel = pi * r ** 2
h = 25.1
A_sylinder = O * h + 2 * A_sirkel
print('Areal av sylinderen:', A_sylinder)
```

Diagram illustrating the execution order of the code. Three green arrows point downwards from the variables `pi`, `r`, and `O` to the expression `pi * r ** 2` in the line `A_sirkel = pi * r ** 2`. The arrows are numbered 3, 2, and 1 from left to right, indicating the order of evaluation: `pi` (3), `r` (2), and `r ** 2` (1).

Illustrasjon av programkjøring



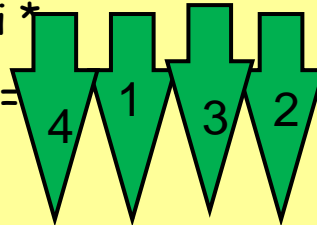
```
pi = 3.14  
r = 5.7  
O = 2 * pi * r  
A_sirkel = pi * r ** 2  
h = 25.1  
A_sylinder = O * h + 2 * A_sirkel  
print('Areal av sylinderen:', A_sylinder)
```



Illustrasjon av programkjøring



```
pi = 3.14  
r = 5.7  
O = 2 * pi *  
A_sirkel =  
h = 25.1  
A_sylinder = O * h + 2 * A_sirkel  
print('Areal av sylinderen:', A_sylinder)
```

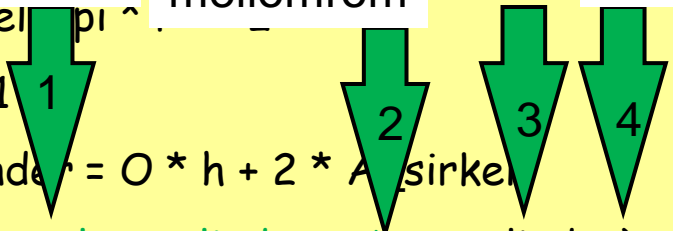


Illustrasjon av programkjøring



```
pi = 3.14
r = 5.7
O = pi * r ** 2
A_sirkel = pi * r ** 2
h = 25.1
A_sylinder = O * h + 2 * A_sirkel
print('Areal av sylindere:', A_sylinder)
```

teksten mellomrom tallet linjeskift



Oppsummering

- Et program forteller hva datamaskinen skal gjøre
- Vi skal bruke programmeringsspråket Python til å programmere datamaskinen
- Før PC-en kan gjøre noe med programmet må det oversettes til maskinvare ved hjelp av et program
- Python kan programmeres interaktivt eller ved å skrive et program i en teksteditor, som IDLE eller PyCharm
- Det er viktige forskjeller på programsetninger og matematiske likninger (og program vs regneark)
- **NESTE TIME: Øvingsforelesning fredag 8-10**
 - Orakeltjenesten, nyttige tips om NTNUs datasystem, installering,
...