Programmering i Python

SDU IT-Camp for Piger 20. marts 2021

Rolf Fagerberg
Institut for Matematik og Datalogi
SDU

Hvad er programmering?

- Få maskinen til at gøre ønskede ting gennem simple instruktioner
- Programmeringssprog minder lidt om simpel engelsk. Bliver automatisk oversat til maskinens kommandoer.
- Nogle programmeringssprog: Python, Java, C, C#, C++, Javascript, Pascal, Delphi, Basic, Fortran, Cobol, Ada, Perl, Ruby,....
- Alle ovenstående bygger på samme principper.

Dagens mål

- Møde programmering
- Lære noget og *gøre* noget
- Forvent ikke at blive programmør på 2 timer :-)
- Forvent at være forvirret undervejs. Helt OK!

Programmering er både spændende og (i starten) lidt forvirrende

Forvirrende:

- Lære et kunstigt sprog
- Vi skal opfinde og beskrive handlinger, som løser en opgave

Spændende:

- Programmer løser opgaver i den virkelige verden
- Programmer er hurtige og effektive (data kan være store)

Hvad er et program?

- Blot et stykke tekst med simple instruktioner.
- Instruktionerne er skrevet i et programmeringssprog, som computeren kan forstå.

Program Eksempel

```
a=3
b=a+2
print(b)
```

Hvad gør man når man har skrevet sit program?

- Programmer skal køres, før der sker noget.
- Et python program køres af en *python fortolker*.

Hvor er python fortolkeren?

- Man kan installere den på sin computer.
- Findes online, så man kan bruge den uden at installere noget.

I dag: Google Colaboratory

- Du behøver ikke at installere noget, alt foregår i browseren.
- Din programmer gemmes i dit Google Drive.
- https://colab.research.google.com/

Grundlæggende elementer i programmering

- Data:
  - 1. Typer
  - 2. Variable
- Kommandoer til at ændre data:
  - 3. Operatorer
  - 4. Tildeling
- Rækkefølge af udførsel af kommandoer:
  - 5. Sekvens
  - 6. Betingelse udførsel
  - 7. Løkker
  - 8. Metoder

1: Typer

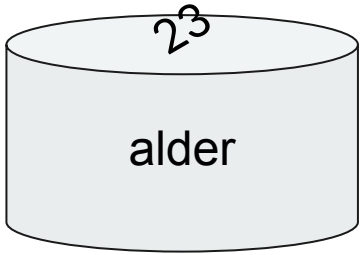
Alt data i et program har en *type*. De vigtigste typer er:

int	heltal	5
float	kommatal	3.22
str	Streng af tekst	“Hej med dig”
bool	Sand eller falsk	5>3
list	Liste	[1,4,10]

2: Variable

En variabel er en beholder, der indeholder noget data. Den har et navn.

```
alder=23
print(alder)
```



3. Operatorer: nye data fra gamle

Operatorer:

- Sædvanlige matematik-operatorer +, -, *, /
- Matematik sammenlignere: <, >, <=, >=
- == lig med
- != ikke lig med
- and, or

Eksempler:

- 2+5, a+5, b*b
- 3<5, price>=30
- a == 7
- a != 6
- username==”jhmed” and password==”Sol334”

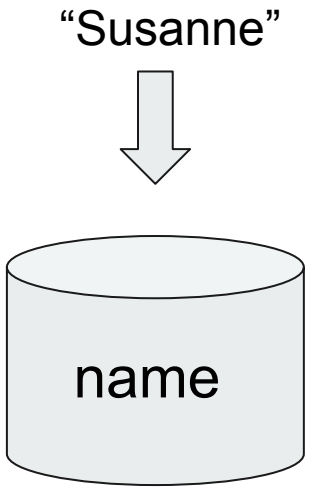
4: Tildeling

Variable tildeles værdier med lighedstegn (=):

```
a=5

name=”Susanne”

b = a+2
```

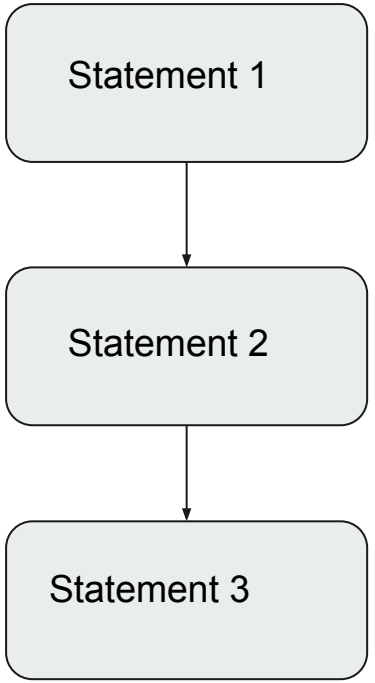


Et variabelnavn på højre side henter værdien gemt i variablen, mens det på venstre side gemmer en ny værdi i variablen.

5: Sekvens

Som udgangspunkt udføres kommandoer én efter én i rækkefølge oppefra og ned

```
a=3
b=a+2
print(b)
```



6: Betinget udførsel: vælg mellem kode

Noget kode udføres kun, hvis betingelse er opfyldt.

```
if name==”Bo”:
    print(“Velkommen Bo”)
else:
    print(“Skrid med dig”)
```

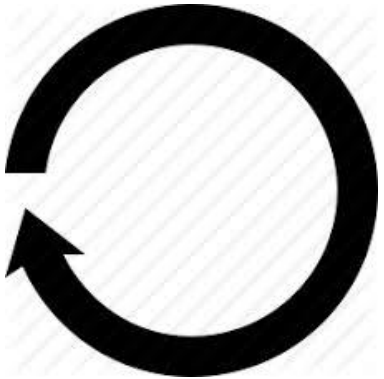
Bemærk indrykningen. Den er vigtig i Python.

7.1: Løkker: gentag kode

En for-løkke: kode gentages et forudbestemt antal gange

```
for count in [1,2,3]:
    print(count)

print("nu!")
```

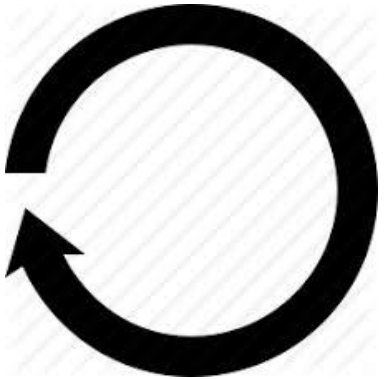


7.2: Løkker: gentag kode

En while-løkke: kode gentages indtil en betingelse ikke længere er sand

```
count=1
while(count<=100):
        print(count)
        count=count+1

print(“Stop!”)
```



8. Metoder (= funktioner): fleksibel genbrug af kode

```
def bowlingBanePris(antalTimer):
    opstart = 100
    timepris = 200
    return opstart + antalTimer * timepris

print(bowlingBanePris(1))
300
print(bowlingBanePris(3))
700
```