



Det Biovidenskabelige Fakultet

Fagpakke i Machine Learning og Algoritmer

Programmering

Arinbjörn Brandsson

Benjamin Rotendahl

Andrea Charlie Stender Cordes

Christopher Mulvad Groot





Program for idag

1. time

- Introduktion til algoritmer.
- Algoritme design og metoder.
- Hvordan man kan sammenligne forskellige løsninger.
- Øvelser i algoritmer.

2. time

- Introduktion til Programmering.
- Programmerings øvelser.

3. time

- Databehandling med Machine Learning.
- Øvelser i dataanalyse.



Programmering

Hvorfor skal vi kode?

- 1 Vi kan bruge at computeren kan lave 10.240 millioner instruktioner per sekund.



Programmering

Hvorfor skal vi kode?

- 1 Vi kan bruge at computeren kan lave 10.240 millioner instruktioner per sekund.
- 2 Vi slipper for at håndkøre algoritmer



Programmering

Hvorfor skal vi kode?

- ① Vi kan bruge at computeren kan lave 10.240 millioner instruktioner per sekund.
- ② Vi slipper for at håndkøre algoritmer
- ③ Algoritmer kan testes og udføres med computere effektivt



Programmering

Hvorfor skal vi kode?

- ① Vi kan bruge at computeren kan lave 10.240 millioner instruktioner per sekund.
- ② Vi slipper for at håndkøre algoritmer
- ③ Algoritmer kan testes og udføres med computere effektivt
- ④ Vi kan arbejde med data som mennesker ikke kan overskue



Programmering

Hvorfor skal vi kode?

- ① Vi kan bruge at computeren kan lave 10.240 millioner instruktioner per sekund.
- ② Vi slipper for at håndkøre algoritmer
- ③ Algoritmer kan testes og udføres med computere effektivt
- ④ Vi kan arbejde med data som mennesker ikke kan overskue
- ⑤ Computeren har allerede vist sig over menneskelig i visse situationer.





Hvorfor python?

Høj niveau sprog Man skal ikke tænke på hvordan maskinen fortolker det.



Hvorfor python?

Høj niveau sprog Man skal ikke tænke på hvordan maskinen fortolker det.

Alsidigt Bliver brugt mange steder, fra webudvikling til kræftforskning



Hvorfor python?

Høj niveau sprog Man skal ikke tænke på hvordan maskinen fortolker det.

Alsidigt Bliver brugt mange steder, fra webudvikling til kræftforskning

Nemt at lære Det har en simpel pæn syntax og er meget tilgivende. Minder meget om pseudokode



Hvorfor python?

Høj niveau sprog Man skal ikke tænke på hvordan maskinen fortolker det.

Alsidigt Bliver brugt mange steder, fra webudvikling til kræftforskning

Nemt at lære Det har en simpel pæn syntax og er meget tilgivende. Minder meget om pseudokode

Minimums algoritmen

```
nums = [42, 314, 1337, 69, 13, 7, 3]
mini = nums[0]
for x in nums:
    if(x < mini):
        mini = x
print mini
```



Matematik og variabler

Matematik

- $3 + 14$



Matematik og variabler

Matematik

- $3 + 14$
- $69 / 2$



Matematik og variabler

Matematik

- $3 + 14$
- $69 / 2$
- $21 * 2$



Matematik og variabler

Matematik

- $3 + 14$
- $69 / 2$
- $21 * 2$
- $2 ** 2$



Matematik og variabler

Matematik

- $3 + 14$
- $69 / 2$
- $21 * 2$
- $2 ** 2$

variabler

- $\pi = 3.14$



Matematik og variabler

Matematik

- $3 + 14$
- $69 / 2$
- $21 * 2$
- $2 ** 2$

variabler

- $\text{pi} = 3.14$
- $\text{liv} = 40 + 2$



Matematik og variabler

Matematik

- $3 + 14$
- $69 / 2$
- $21 * 2$
- $2 ** 2$

variabler

- $\text{pi} = 3.14$
- $\text{liv} = 40 + 2$
- $\text{pi2} = \text{pi} * 2$



Matematik og variabler

Matematik

- $3 + 14$
- $69 / 2$
- $21 * 2$
- $2 ** 2$

variabler

- $\text{pi} = 3.14$
- $\text{liv} = 40 + 2$
- $\text{pi2} = \text{pi} * 2$
- $\text{fornavn} = \text{"Benjamin"}$



Matematik og variabler

Matematik

- $3 + 14$
- $69 / 2$
- $21 * 2$
- $2 ** 2$

variabler

- $\text{pi} = 3.14$
- $\text{liv} = 40 + 2$
- $\text{pi2} = \text{pi} * 2$
- $\text{fornavn} = \text{"Benjamin"}$
- $\text{efternavn} = \text{"Rotendahl"}$



Matematik og variabler

Matematik

- $3 + 14$
- $69 / 2$
- $21 * 2$
- $2 ** 2$

variabler

- $\text{pi} = 3.14$
- $\text{liv} = 40 + 2$
- $\text{pi2} = \text{pi} * 2$
- $\text{fornavn} = \text{"Benjamin"}$
- $\text{efternavn} = \text{"Rotendahl"}$
- $\text{mig} = \text{fornavn} + \text{efternavn}$



Matematik og variabler

Matematik

- $3 + 14$
- $69 / 2$
- $21 * 2$
- $2 ** 2$

variabler

- $\text{pi} = 3.14$
- $\text{liv} = 40 + 2$
- $\text{pi2} = \text{pi} * 2$
- $\text{fornavn} = \text{"Benjamin"}$
- $\text{efternavn} = \text{"Rotendahl"}$
- $\text{mig} = \text{fornavn} + \text{efternavn}$
- $\text{harKage} = \text{True}$



Funktioner og lister

lister

lst = [1,2,3,4,5] variabelen “lst” er nu en liste med tal.



Funktioner og lister

lister

lst = [1,2,3,4,5] variabelen “lst” er nu en liste med tal.

nr = lst[0] variabelen nr er nu lig med det første tal i listen.



Funktioner og lister

lister

lst = [1,2,3,4,5] variabelen “lst” er nu en liste med tal.

nr = lst[0] variabelen nr er nu lig med det første tal i listen.

lst.append (6) listen er nu blevet en længere og det sidste element er 6.



Funktioner og lister

lister

lst = [1,2,3,4,5] variabelen “lst” er nu en liste med tal.

nr = lst[0] variabelen nr er nu lig med det første tal i listen.

lst.append (6) listen er nu blevet en længere og det sidste element er 6.

funktioner

- **mindst = min(lst)**



Funktioner og lister

lister

lst = [1,2,3,4,5] variablen “lst” er nu en liste med tal.

nr = lst[0] variablen nr er nu lig med det første tal i listen.

lst.append(6) listen er nu blevet en længere og det sidste element er 6.

funktioner

- **mindst = min(lst)**
- **forksel = max(lst) - min(lst)**



Funktioner og lister

lister

lst = [1,2,3,4,5] variablen "lst" er nu en liste med tal.

nr = lst[0] variablen nr er nu lig med det første tal i listen.

lst.append(6) listen er nu blevet en længere og det sidste element er 6.

funktioner

- **mindst = min(lst)**
- **forksel = max(lst) - min(lst)**
- **print "hejsa"**



Funktioner og lister

lister

lst = [1,2,3,4,5] variablen “lst” er nu en liste med tal.

nr = lst[0] variablen nr er nu lig med det første tal i listen.

lst.append (6) listen er nu blevet en længere og det sidste element er 6.

funktioner

- `mindst = min(lst)`
- `forksel = max(lst) - min(lst)`
- `print "hejsa"`
- `prikprod(v, u)`



kontrol udtryk

If statements

Hvis a så gør b .

Hvis a så gør b ellers gør c .

Løkker

- `for i in range(0,10):`



kontrol udtryk

If statements

Hvis a så gør b .

Hvis a så gør b ellers gør c .

Løkker

- for i in range(0,10):
- for i in lst:



kontrol udtryk

If statements

Hvis a så gør b .

Hvis a så gør b ellers gør c .

Løkker

- for i in range(0,10):
- for i in lst:
- while(point < 100):



Funktions definition

prikprodukt

For vektoren v og vektoren u er prikproduktet defineret som

$$v_1 \cdot u_1 + v_2 \cdot u_2$$

En funktion er en måde at genbruge sin kode på.



Funktions definition

prikprodukt

For vektoren v og vektoren u er prikproduktet defineret som

$$v_1 \cdot u_1 + v_2 \cdot u_2$$

En funktion er en måde at genbruge sin kode på.

Demo

Så er der live demo!



Funktions definition

Fibonacci tal

Det n 'te fibonacci tal er defineret som

$$fib(n-1) + fib(n-2)$$



Funktions definition

Fibonacci tal

Det n 'te fibonacci tal er defineret som

$$fib(n-1) + fib(n-2)$$

Demo

Den koder vi!



Funktions definition

Fakultets funktionen

Fakultetfunktionens af n (typisk skrevet $n!$) er defineret som

$$n * (n - 1) * (n - 2) * \dots * 1$$

Demo

Den koder vi også!



Hvad med et gætte spil?

Vi ønsker at kode et lille spil hvor brugeren skal gætte det tal computeren har valgt.

Demo

Den koder vi også!



Øvelsestid

Kode øvelser i python!

