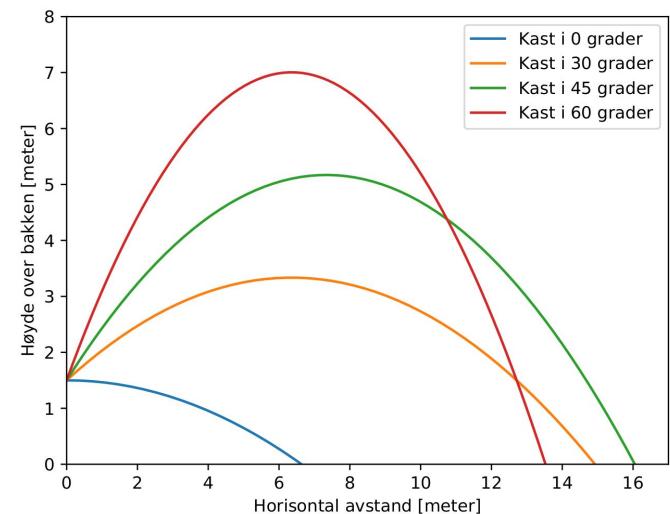


Programmeringskurs for Realfag



simula
kodeskolen

```
1 import numpy as np
2 n = 12*50    #antall tidsintervaller
3 y0 = 100      #antall byttedyr når vi starter
4 x0 = 50       #antall rovdyr når vi starter
5 index_set = range(n+1)
6
7 x = np.zeros(len(index_set))
8 y = np.zeros(len(index_set))
9
10
11 a = 0.05    # dødsrate gauper
12 b = 0.0003  # reproduksjonsrat
13
14 c = 0.02    # vekstrate harer
15 d = 0.0001  # dødsrate harer
16
17
18 y[0] = y0
19 x[0] = x0
20 for k in index_set[:-1]:
21     #print y[k]
22     y[k+1] = y[k] + c*y[k] -
23     x[k+1] = x[k] - a*x[k] +
```



Praktisk Info



Velkommen til
simula

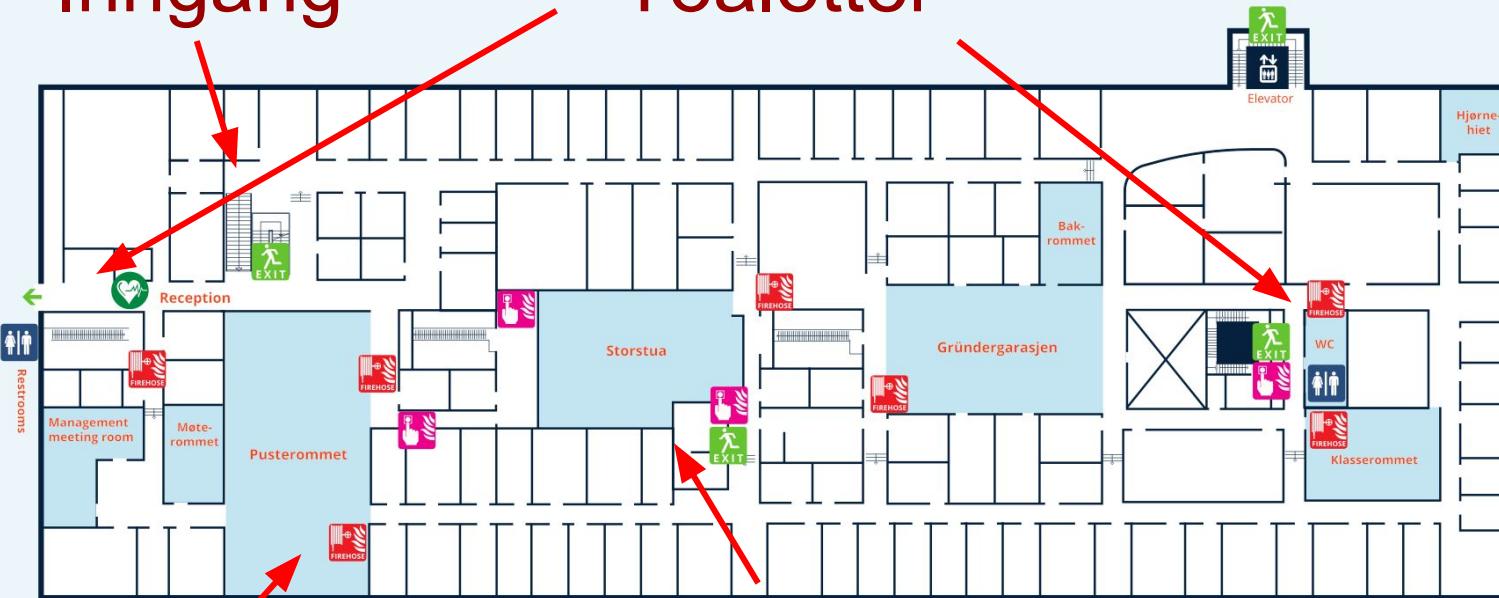
Hvorfor programmering?



Praktisk Info

Inngang

Toaletter



Lunsj

Kursrom

- Defibrillator
- Alternative exits
- Fire hose
- Fire alarm
- Restrooms

9.00-12.00 Formiddagsøkt

12.00-13.00 Lunsjpause

13.00-16.00 Ettermiddagsøkt

Dag 1 - Innføring til Python

Dag 2 - Innføring til Python

Dag 3 - Faglige Opplegg

Dag 4 - Faglige Opplegg

Dag 5 - Lage eget opplegg + Presentasjoner

Wifi: SRL Kodeskolen (eller eduroam)

**Dere må registrere dere med navn og mobilnr
Passord sendes på sms**

Kursmateriale deles på

github.com/kodeskolen/vgs

Materialer legges ut etterhvert som kurset går

Møt de kursansvarlige



Marie



Jonas



Eina

Møt de kursansvarlige



Marie



Jonas



Eina



Freyja



Håkon

Det kan bli tatt bilder i løpet av kurset

**Freyja vil samle inn
samtykke fra dere**

**Det er selvfølgelig lov å
reservere seg**



**Velkommen
til
Simula**

Simula er et forskningsinstitutt som driver med grunnleggende, langsiktig forskning

Fornebu



simula

Oslo



Bergen



Simula driver med forskning innenfor fem ulike felt, alle relatert til informatikk og programmering



Communication
Systems



Scientific
Computing



Machine Learning



Cryptography



Software
Engineering

Datterselskapet Simulaskolen fokuserer på utdannelse av forskere gjennom master og doktorgradsstudenter



Antall uteksaminerte:

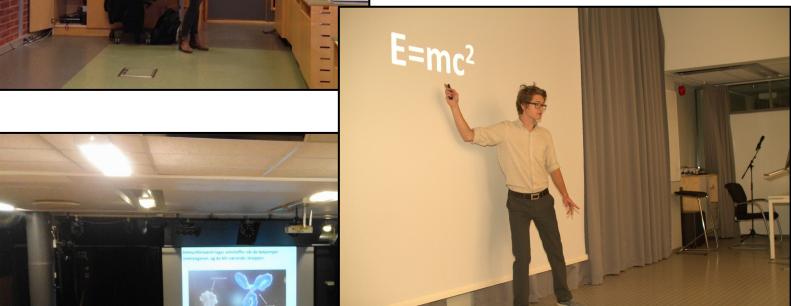
Mastergrad	408
Doktorgrad	119

September 2018

Vi jobber også på realfagsrekruttering blant barn og unge



Skolebesøk



Realfagsambassadører

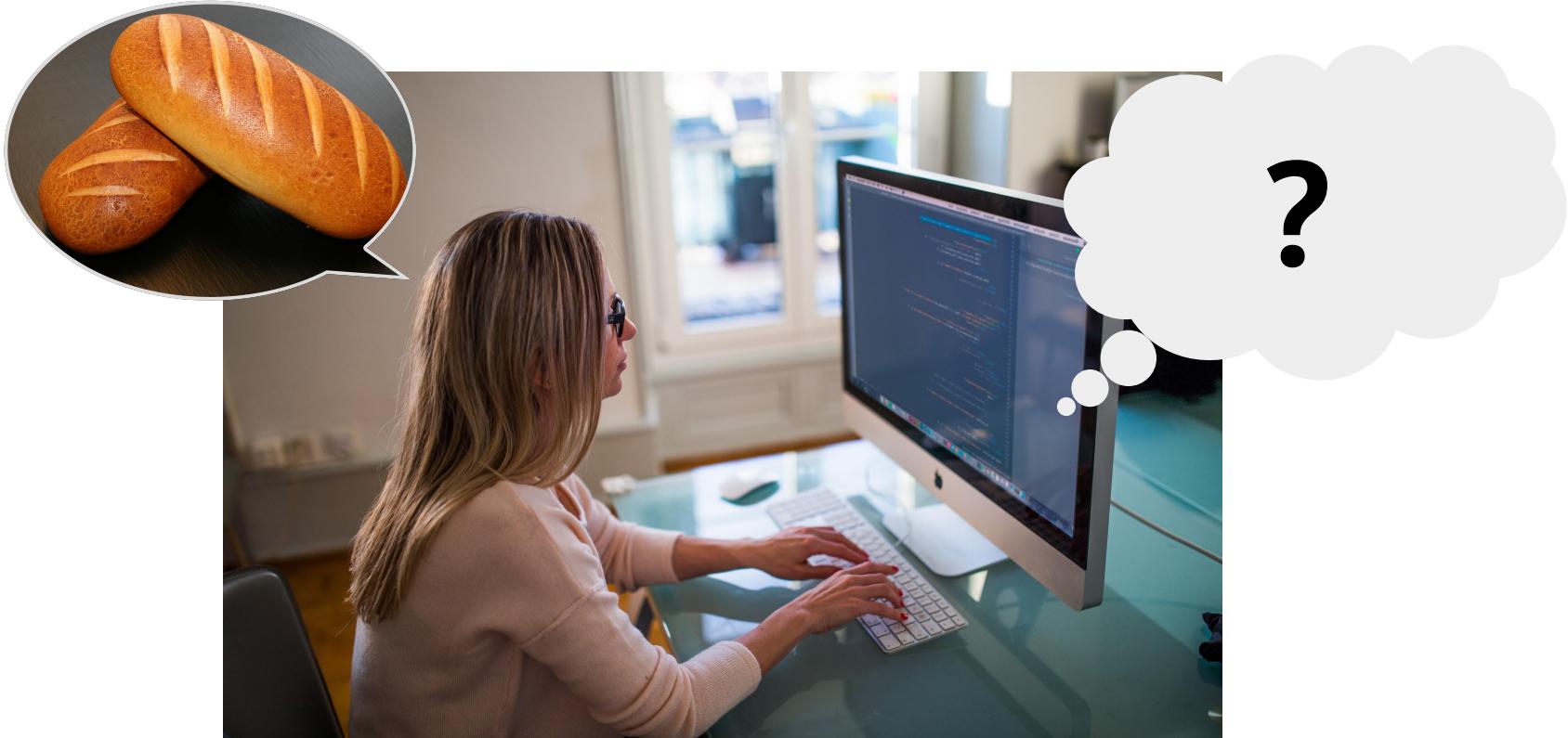
Hva er programmering?
-og hvorfor skal vi lære det?

Programmering handler om å instruere en datamaskin eller andre digitale enheter til å utføre en oppgave eller løse et problem

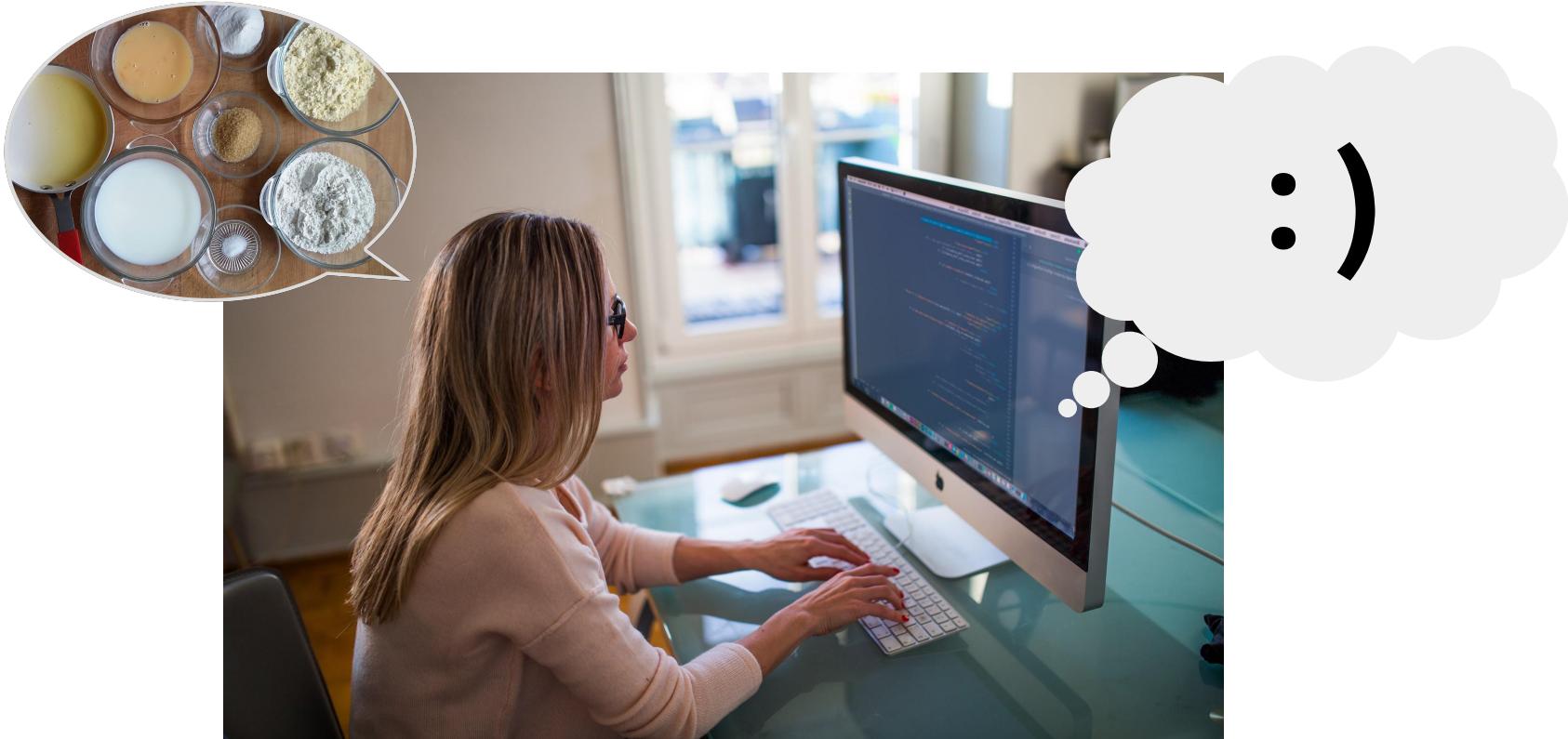


```
1 import numpy as np
2 n = 12*50      #antall tidsintervaller
3 y0 = 100        #antall byttedyr når vi starter
4 x0 = 50         #antall rovdyr når vi starter
5 index_set = range(n+1)
6
7 x = np.zeros(len(index_set))
8 y = np.zeros(len(index_set))
9
10
11 a = 0.05       # dødsrate gauper
12 b = 0.0003     # reproduksjonsrate gauper
13
14 c = 0.02       # vekstrate harer
15 d = 0.0001     # dødsrate harer
16
17
18 y[0] = y0
19 x[0] = x0
20 for k in index_set[:-1]:
21     #print y[k]
22     y[k+1] = y[k] + c*y[k] - d*y[k]*x[k]
23     x[k+1] = x[k] - a*x[k] + b*x[k]*y[k]
```

Datamaskiner er egentlig ganske dumme,
og trenger derfor enkle og nøyaktige instrukser

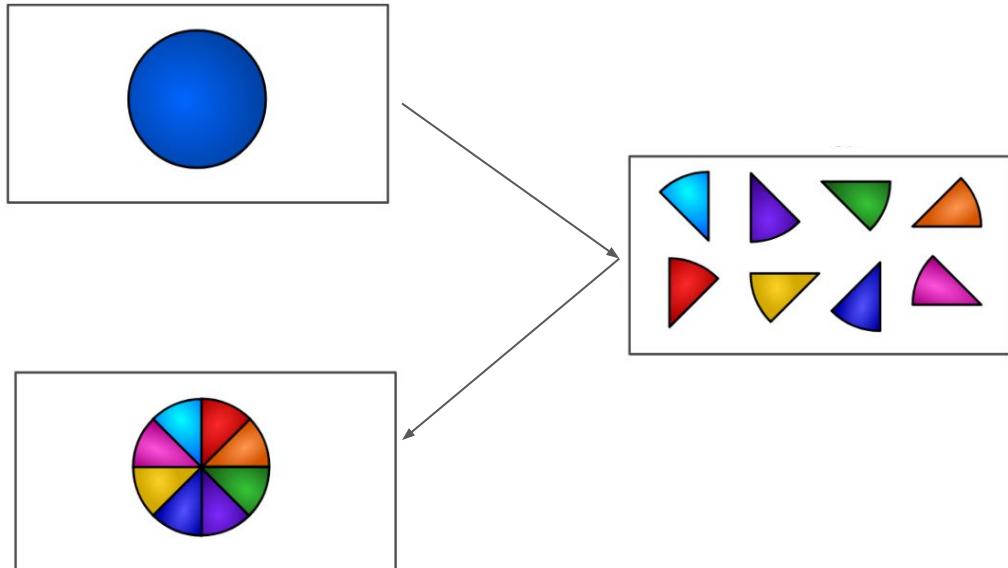


Datamaskiner er egentlig ganske dumme,
og trenger derfor enkle og nøyaktige instrukser



Programmering handler om å bryte opp kompliserte problemer i små biter, og løse dem steg for steg

Dette kalles gjerne for
algoritmisk tankegang



**“Everybody in this country should learn to program
a computer... because it teaches you how to think”**

**“Everybody in this country should learn to program
a computer... because it teaches you how to think”**



-Steve Jobs

“Vi vil at forståelse for koding og teknologi skal inn i læreplanene allerede fra barneskolen.”



Samfunnet og yrkeslivet blir i større og større grad avhengig av digitale løsninger og verktøy

“Blir det ikke som å lære alle som skal kjøre bil å bli bilmekanikere?”



Programmering i realfag

Studier viser at forskere innen naturfagene bruker typisk minst 30% av tiden sin på programmering

...men 90% av forskere er hovedsakelig selvlærte

How Do Scientists Develop and Use Scientific Software?

Jo Erskine Hannay
Dept. of Software Engineering
Simula Research Laboratory
Dept. of Informatics, Univ. of Oslo
johannay@simula.no

Carolyn MacLeod
Dept. of Computer Science
University of Toronto
cmacleod@cs.utoronto.ca

Janice Singer
Software Engineering Group
National Research Council of Canada
janice.singer@nrc-cnrc.gc.ca

Hans Petter Langtangen
Center for Biomedical Computing
Simula Research Laboratory
Dept. of Informatics, Univ. of Oslo
hpl@simula.no

Dietmar Pfahl
Dept. of Software Engineering
Simula Research Laboratory
Dept. of Informatics, Univ. of Oslo
dietmarp@simula.no

Greg Wilson
Dept. of Computer Science
University of Toronto
gvwilson@cs.utoronto.ca

Abstract

New knowledge in science and engineering relies increasingly on results produced by scientific software. Therefore, knowing how scientists develop and use software in their research is critical to assessing the necessity for improving current development practices and to making decisions about the future of scientific software. This paper identifies

1. Motivation

There is probably not a single scientist who has not, at some point in time, used a software system to analyze, visualize, or simulate processes or data. Many scientists use such software daily, while others develop it for their own use or for a wider community.

Det foregår en dugnad innenfor vitenskapene, der programmering og datakunnskaper undervises som nødvendige verktøy



DATA CARPENTRY

BASED ON THE UNTOLD TRUE STORY

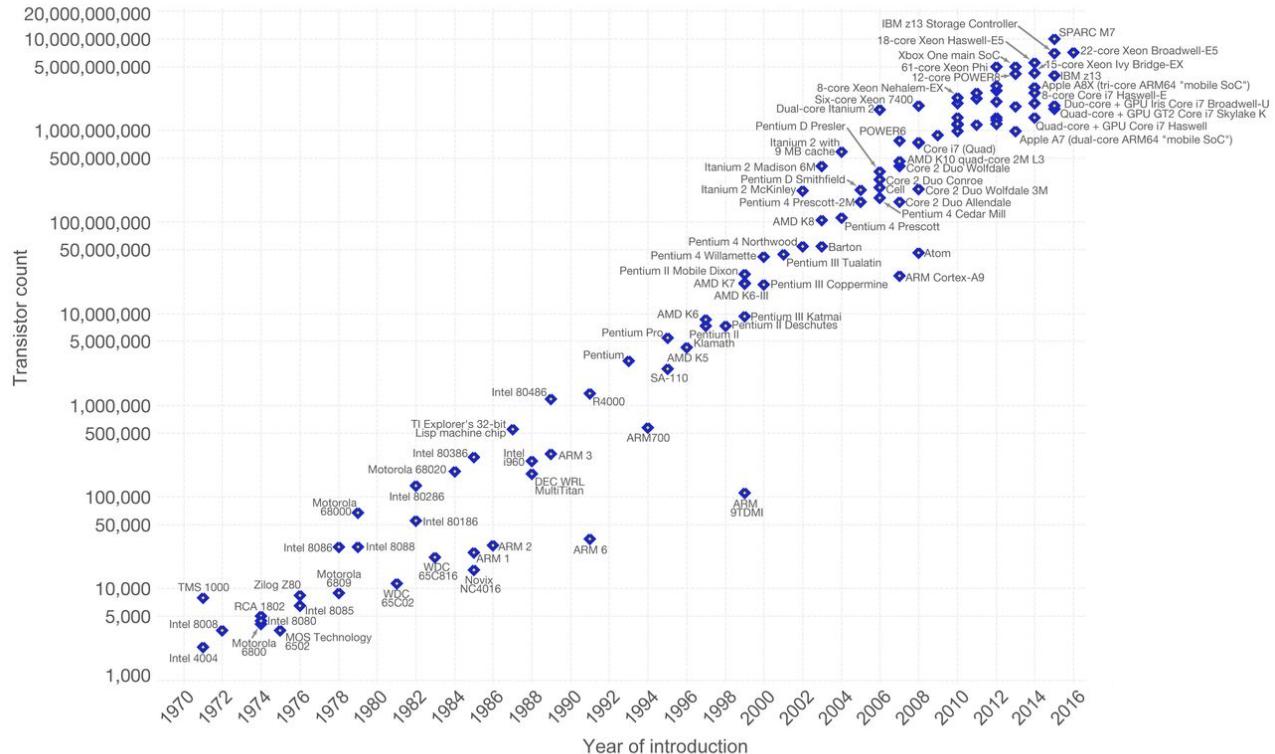
HIDDEN FIGURES



Beregningskraften har vokst eksponentielt siden slutten av 60-tallet

Moore's Law – The number of transistors on integrated circuit chips (1971-2016) OurWorld in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.

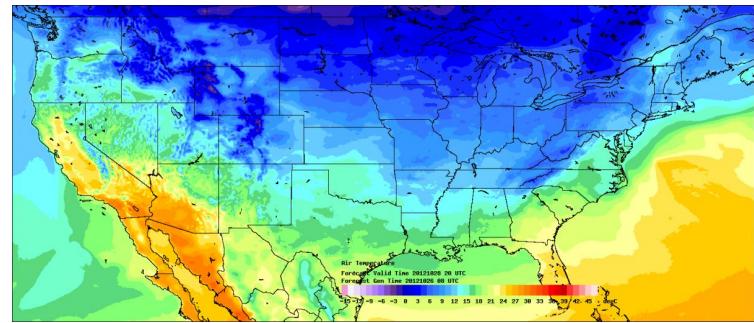
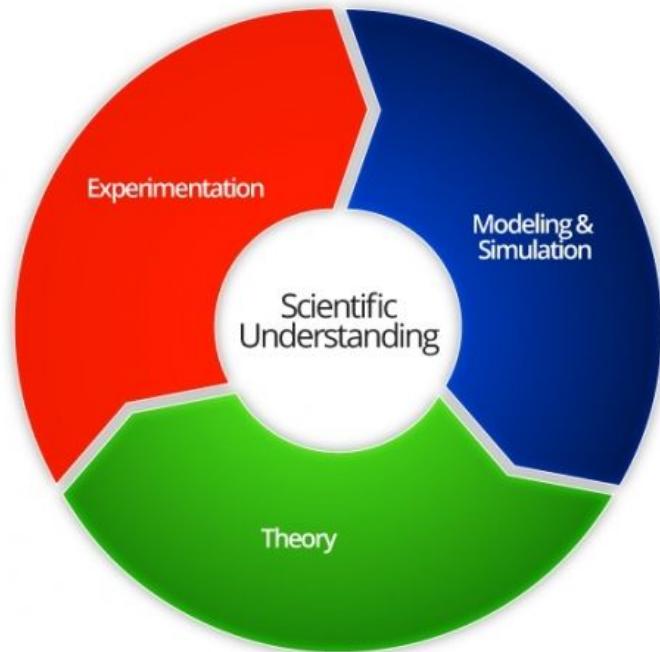


Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)

The data visualization is available at OurWorldInData.org. There you find more visualizations and research on this topic.

Licensed under CC-BY-SA by the author Max Roser.

Numeriske metoder lar oss løse ligninger og problemstillinger som rett og slett er umulig å løse med penn og papir



Programmering har gjort sitt inntog i universitetsutdannelsen, ved UiO lærer samtlige realfagsstudenter programmering første semester

6. semester	Se studieretning	Se studieretning	Se studieretning
5. semester	Se studieretning	Se studieretning	Se studieretning
4. semester	Se studieretning	Se studieretning	Se studieretning
3. semester	MAT1120 – Lineær algebra	Se studieretning	IN1910 – Programmering for naturvitenskapelige anvendelser
2. semester	MAT1110 – Kalkulus og lineær algebra	MEK1100 – Feltteori og vektoranalyse	STK1100 – Sannsynlighetsregning og statistisk modellering
1. semester	MAT1100 – Kalkulus og HMS-emner	MAT-INF1100 – Modellering og beregninger	IN1900 – Introduksjon til programmering for naturvitenskapelige anvendelser
	10 studiepoeng	10 studiepoeng	10 studiepoeng

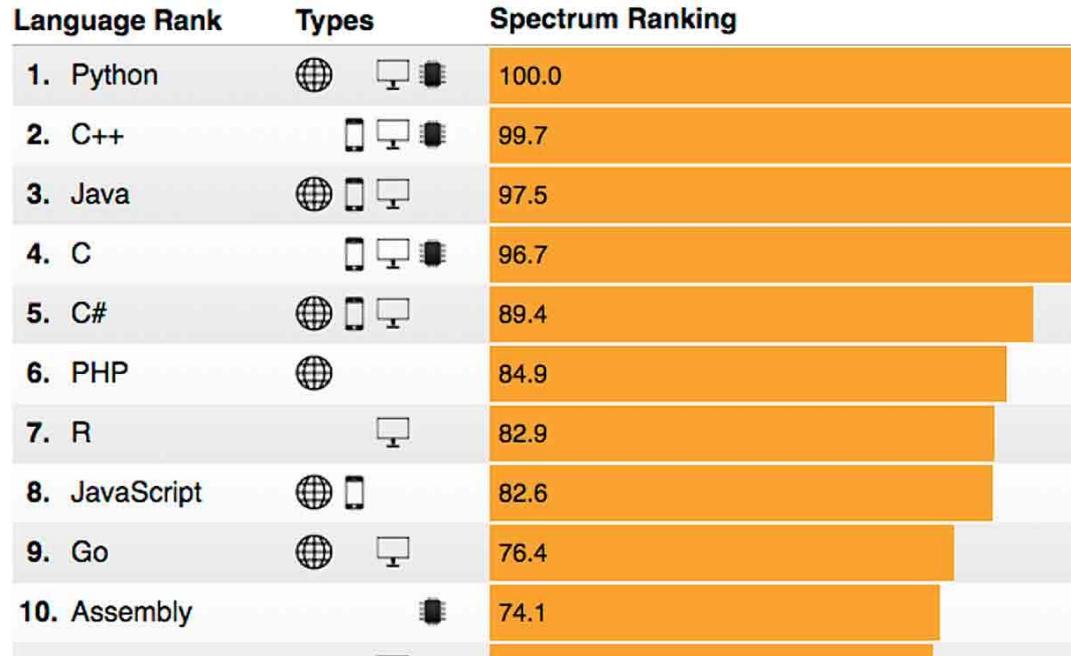


Oppbygningen av en bachelorgrad i matematikk ved UiO



Hvorfor Python?

Python er et av de aller mest brukte programmeringsspråk i dag, på tvers av fagfelt



Python er også blitt det mest populære språket å undervise i første semesters kurs ved høyskoler og universiteter

**Python er plattformuavhengig og lett å installere,
det er også utviklet open-source**



**Anaconda er den letteste måten å installere
på Windows, Mac, eller Linux og er gratis**

Uavhengig av hvilke verktøy/språk vi velger, er det viktig å heve blikket og se på de underliggende konseptene

- Algoritmisk tankegang
- Problemløsning
- Begrepsforståelse
- Tverrfaglig forståelse