

# Python-kurs for NTNU Konstruksjonsteknikk

05. November 2018

Teodor Heggelund, Teknologisk leder Pure Logic AS

# Hei!

- Teodor Heggelund
- **2015** Uteksaminert KT/Beregningsmekanikk, Odd Sture, tøyningsgradientplastisitet
- **2015-2017** Styrkeanalyser Norconsult. Stål, betong, Abaqus, Robot
- **2018** Ansvar for utvikling (team og kode) i [Pure Logic AS](#)

# Agenda

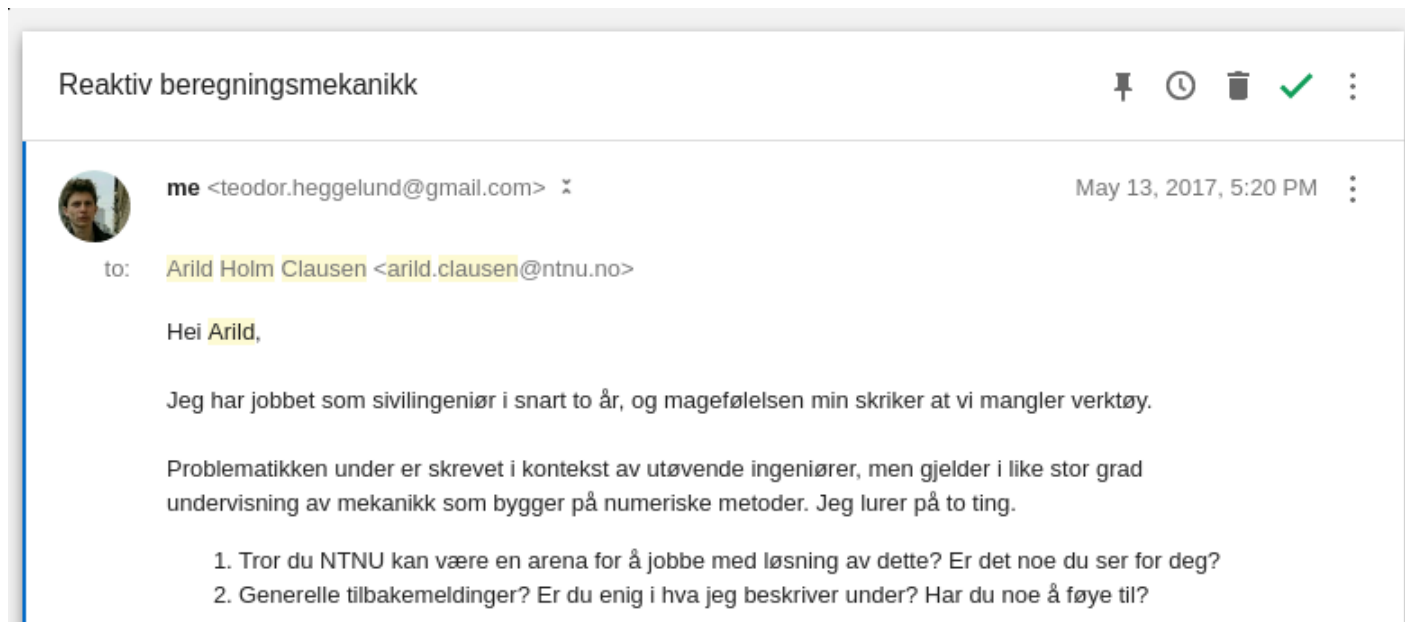
- Introduksjon
- Del 1: Python-programmering med Jupyter Notebooks
- Del 2: Vitenskaplig programmering med Python
- Del 3: Samhandling med Jupyter Notebooks
- Finale

Intro. Hvorfor er vi her?

# Arild!

- Oppfølging
- Praktisk arbeid
- Fornuftig styring / veiledning.

# Hva har skjedd? (1)



# Hva har skjedd? (2)

- Seminar utdanning ved NTNU Konstruksjonsteknikk  
16. April 2018
- Industrirepresentasjon med bakgrunn fra Aas Jacobsen, Multiconsult, Olav Olsen, Norconsult, Pure Logic og Statoil
- "Hvordan forholde seg til digitalisering?"
  - "På et fornuftig vis slik at det støtter opp under Konstruksjonsteknikk, og ikke ødelegger det som allerede fungerer bra"

# Hva har skjedd? (3)

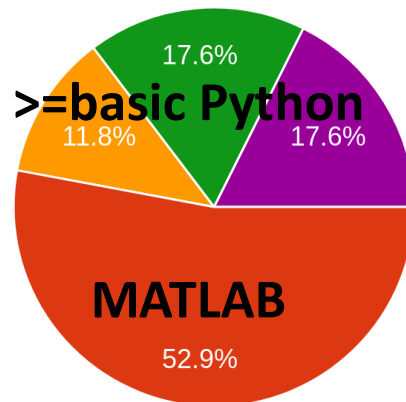
- Python i IKT Grunnkurs fra 2018
- Større spørsmål: Hvordan passer data inn i konstruksjonsteknikk?
- Seminar med NTNU KT og industripartnere sommeren 2018, arrangert av Arild
- Innsats på tampen av 2018 der jeg er involvert:
  - Python-kurs for forelesere
  - Proof-of-concept; "Nytte av Python i forelesning"



# Hvem er vi nå?

Har du programmert i Python fra før?

17 responses

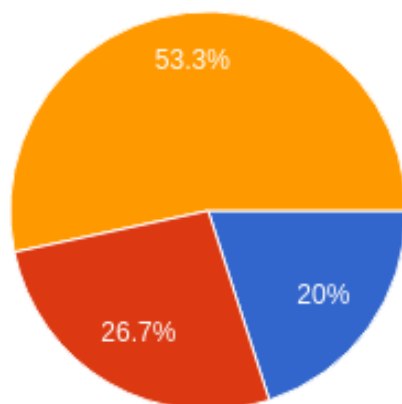


- Jeg har ikke erfaring med programmering
- Ikke Python, men jeg har programmert i Matlab eller andre programmeringsspråk
- Jeg kjenner til grunnleggende Python
- Jeg kan bruke lister, løkker og dictionaries i Python
- Jeg kan i tillegg lage mine egne moduler og klasser i Python

# Hvem er vi nå?

Er du interessert i å bruke Python i forelesninger?

15 responses

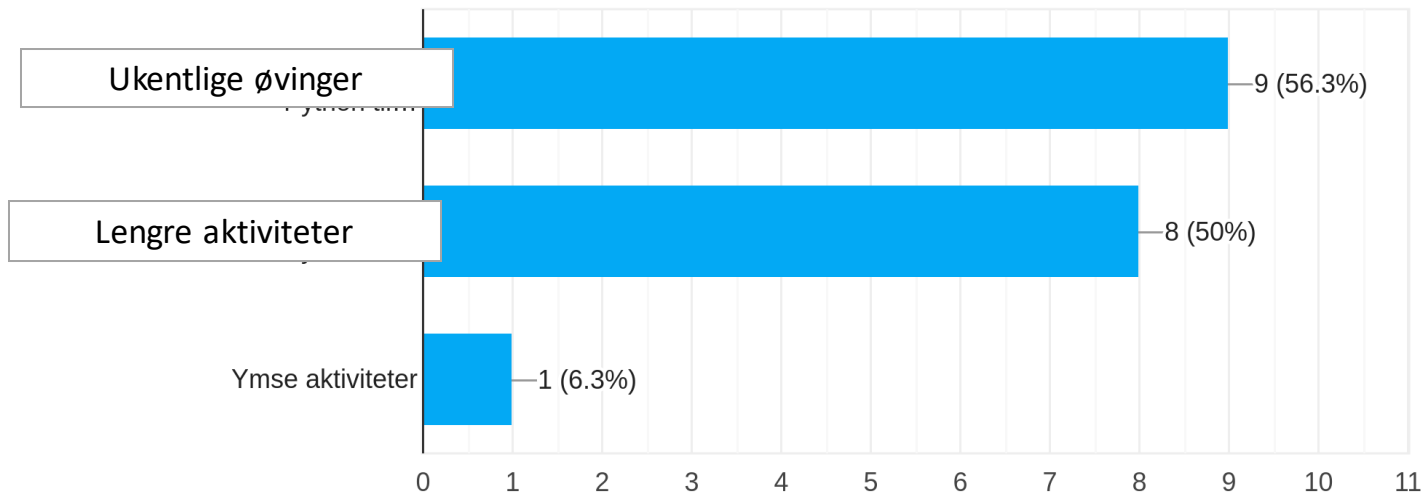


- Nei.
- Kun dersom jeg kan bruke <5 minutter av forelesningstid
- Kun dersom jeg kan bruke <20 minutter av forelesningstid

# Hvem er vi nå?

Er du interessert i å bruke Python utenfor forelesninger?

16 responses



# Hvem er vi nå?

Jeg ønsker å bruke Python slik (alternativt la stå tom):

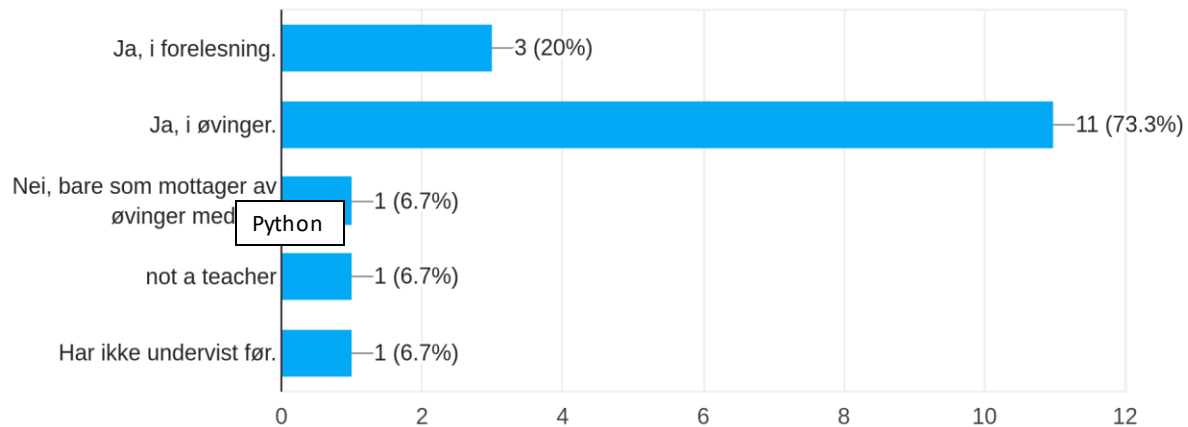
1 response

Grunnleggende ting som å plote funksjoner, matrisealgebra som f.eks. å finne egenverdier og egenretninger, samt å løse enkle ligninger.

# Hvem er vi nå?

Har du tidligere brukt programmering som verktøy i undervisning?

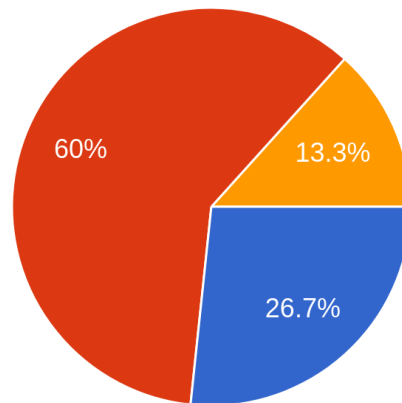
15 responses



# Hvem er vi nå?

Følgende er viktigst for at Python skal fungere godt i min undervisning:

15 responses



- Enkelt å bruke uten at noe henger!
- God mulighet til å visualisere data relevant for mine kurs
- Mulighet til å involvere studenter interaktivt

# Hvem er vi nå?

Jeg har andre kommentarer eller anbefalinger!

3 responses

Av egen interesse er jeg interessert i bruk av grafikk som endrer seg ved parameter input, eller click-and-drag

I would attend if the course were in English

is it possible to give this lecture in English?

# Hvem er vi nå?

## Mine fag (valgfri)

3 responses

TKT4108 (Advanced Dynamics)

Mekanikk 2

Mekanikk



# Oppsummering behov

- 70 % ønsker Python i forelesninger
- 100 % ønsker Python i øvinger eller lengre prosjekter
- Må være enkelt å bruke
- Må kunne visualisere data godt
- Nyttige anvendelser innen lineær algebra

# Plan for i dag

- Variabel Python-kompetanse
- Mange kommer til Python fra annet språk, og trenger en introduksjon (del 1)
- Mange etterspør vitenskaplig bruk (del 2)
  - Datavisualisering
  - Likningsløsning
- Hvordan bør vi bruke Python sammen? (del 3)

# Plan for i dag

***Vise hva som er mulig, hvordan ting henger sammen  
uten å bruke alt for mye tid på teori.***

# Del 1

Python-programmering med Jupyter Notebooks

# Starte Python

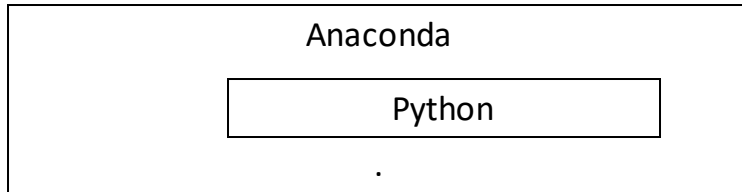
- Kommandolinje

- *python* – skriv setninger interaktivt
- *python script.py* – kjør script.py

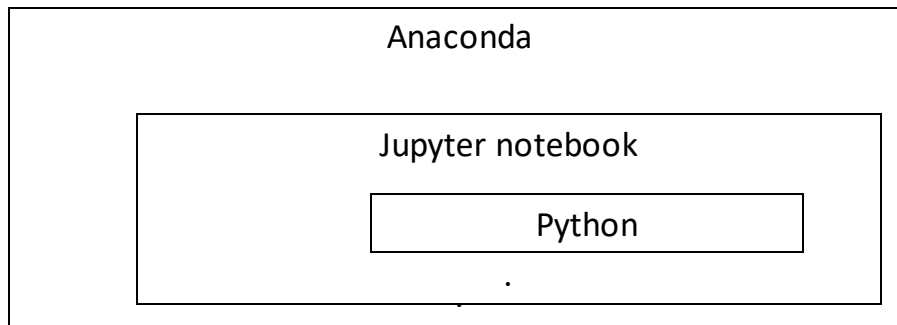
- Jupyter Notebook

- *jupyter notebook* – start lokal notebook med tilgang til mappen du starter fra
  - Starter du fra kommandolinjen, ser du typisk C:\Users\username\, /Users/username/ eller /home/username/
- *jupyter lab* – start "litt bedre" miljø som du lettere kan jobbe med flere notebooks og vanlige Python-moduler

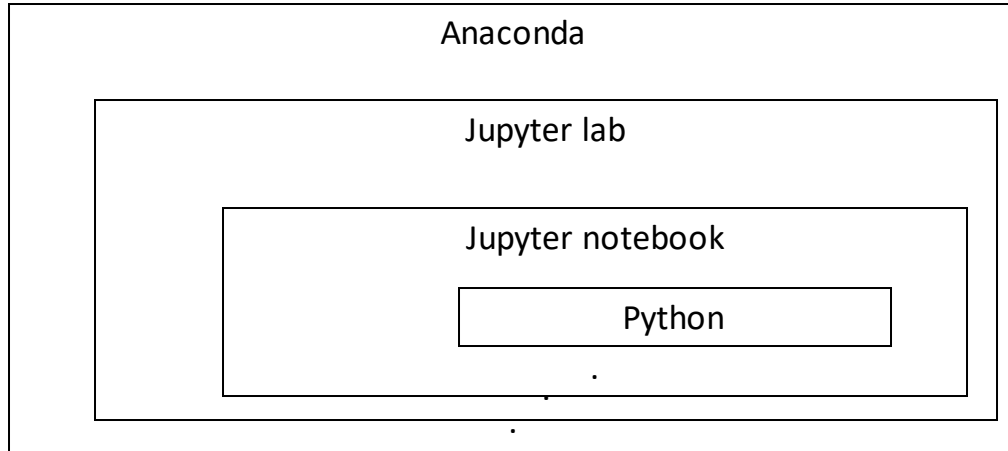
# python script.py



# jupyter notebook



# jupyter lab





# Hvordan funker x

- *help(x)*

# Python-primitiver

- Variabler
- Tekst
- Tall
- "Truthy"
- If
- While

# Python er dynamisk typet

- Python lar deg kombinere typer som du vil
- Ingen kompilering
- Krasj om du prøver å behandle tall som tekst
- Bruk *type(value)* for å se typen til value

# Funksjoner, "navngitt transformasjon"

- *def* for å navngi en funksjon
- Funksjoner er førsteklasses verdier
  - `def f(x): return 2 * x + 3`
  - `def integrate(f, from): # ...`
  - `F = integrate(f, 0)`

# Sekvenser, "ting etter hverandre"

- Tupler
  - Konstant lengde og verdier med forskjellige typer
  - ("Teodor", 2015, "teodor@teodorheggelund.com")
  - collections.namedtuple: tuppel der verdiene navngis
  - dict (senere): navngitte verdier, og variabel lengde
- Lister
  - Variabel lengde og verdier av samme type
  - ["Arild", "Odd Sture", "Anders"]
- (dette er konvensjon, ikke lov; "Python er dynamisk typet")

# Kompositter, "typer inni typer"

- Lister kan inneholde andre lister
- Lister kan inneholde tupler
- Tupler kan inneholde lister av tupler av funksjoner
- You name it.
- `[("Trondheim", ["ntnu", "hist"]), ("oslo", ["uio", "MET"])]`
- (Der Matlab ofte "foretrekker flate matriser")

# Mapping, "fra X-er til Y-er"

- Dictionaries
  - Variabel lengde, navngitte verdier, ingen rekkefølge
  - `d = {"teodor": 2015, "arild": 1991}`
  - Oppslag: `d["arild"] # => 1991`
  - Endring: `d["odd sture"] = ...`
  - Gå gjennom alt:  

```
for name, year in d.items():  
    print(name, "finished in", year)
```
- "Er du i tvil på hvordan du skal modellere, start med en dict"

# Modellering av data + prosedyre

- Klasser – objektorientert programmering (OO)
  - dict, list, tuple, til og med function er klasser.
  - Superfleksibelt
  - Subjektiv vurdering:
    - Felles elementløser kan på god måte løses med et objektorientert API (Application Programming Interface, programmeringsgrensesnitt)
    - Studenter bør ikke undervises i programvaredesign med OO på NTNU Konstruksjonsteknikk
    - Finnes allerede TDT4100 Objektorientert Programmering



# Del 2

Vitenskaplig programmering med Python

Still gjerne spørsmål!

# Anaconda bundler med et utvalg gode pakker

- Numpy: numerisk programmering
  - `numpy.array` (n-dimensjonal), `numpy.linalg`, `numpy.matlib` (matriser)
- Scipy: samling vitenskaplig programmering
  - `scipy.integrate` (integrasjon, diff. likn.), `scipy.interpolate`, `scipy.linalg`, `scipy.sparse`
- Sympy: symbolsk matematikk
  - *Sympy.sqrt(24)  $\Rightarrow$  2\*sqrt(6)*
- Matplotlib: Visualisering, *stort*, "funksjoner for alt"
- Altair: Visualisering, "visualiseringsspråk"

# Eksempel: analyse av sinus

- Sinus og cosinus
- Derivasjon med Sympy
- Evaluering med Numpy
- Plotting med Altair

# Del 3

Samhandling med Jupyter Notebooks

# Typer samhandling

- Eksempel: Notebook (.ipynb)
- Delt fundament: Python-pakke (.py)

# Finale

Hvor går veien videre?

# Bør vi ha et felles bibliotek?

- Hvordan bør vi samhandle?
- Hvordan sikrer vi integritet?
- Hvem bør ansvaret ligge hos?



# Hvordan bør vi jobbe med Elementmetoden?

- Kan vi la studentene bygge videre på kompetanse fra fag til fag?
- Hva er "felles"?
- Hva er forskjellig?
- Hvilke fag er affisert?

Spørsmål?

# Jupyter i undervisning (min subjektive)

- Dele notebook, elever fylle inn
- Veilede elever i Python-verdenen: mange løsninger
- Støtte til de som trenger hjelp til å komme i gang, fleksibilitet til de som ønsker å trekke det lenger

# Delt kompetanse mellom fag (forslag)

- Visualisering
- Elementmetode
  - Tar mye tid med nytt opplegg for hvert fag
- Felles bibliotek?
  - Github: <https://github.com/ntnu-kt/ntnu.kt>
  - Pypi: <https://pypi.org/project/ntnu.kt/>
  - Fagspesifikt: ntnu.kt.course.tkt4122
  - Felles: ntnu.kt.vis, ntnu.kt.fem
  - Ansvar?