



Introduktion til Python & SageMath



pythonTM



Tomas H. Andersen

Lad os tage installationen først:



- 1) Gå til min github:
<https://github.com/Yuiyjg09/Introduktion-til-Python-og-SageMath>
- 2) Åbn "installations_guide.txt" og følg den

Alternative IDE'er:

- Eclipse (Med PyDev plugin)
- Pycharm
- Spyder



Hvad er Python og Sagemath?



- Python: Et “*General-Purpose*” *Interpreted* Programmeringssprog, hvis popularitet er stigende.
- SageMath: En “*Distribution*” af Python version 2.7, der har ekstra indbyggede funktioner og libraries til matematik.

Hvad er Jupyter Notebook?

- Det er lidt ligesom Rstudio bare med Python/SageMath.
- Kan også bruge “R”, Javascript, Markdown osv.
- Færdigheder er eftersprugt i erhvervslivet



Python i Data Science

- Bruges til mange af det samme som “R”, mest “*Scripting*”
- Mest anvendte libraries:
 - Pandas (data import og håndtering)
 - Matplotlib (plotting a la ggplot2)
 - Numpy (Numerisk Lineær Algebra)
 - Scipy (Udvidet Numpy)
 - (SymPy: Symbolske beregninger)



Python Datatyper og - strukturer



- Datatyper:
 - Numbers, eks: 1, 2.87
 - Strings, eks: "Hello, World!", 'Some text'
 - Bools, eks: True, False, 0, 1
- Datastrukturer:
 - Lists (ArrayList/Vectors): [1, 2, 3], ["En", "To", "Tre"]
 - Dictionaries (Maps): {1 : "En", 2 : "To", 3 : "Tre"}
 - Tuples (Arrays): (1, 2, 3), ("En", "To", "Tre")

Operationer



- Numbers:
 - $+$, $-$, $*$, $/$, $\%$, $**$ (Sage: $^$)
- Logisk:
 - and, or, not

Control flow



- If-else statements
 - `if`
 - `elif`
 - `else`
- Loops
 - `while` (*condition*) :
 - `for` (*condition*)

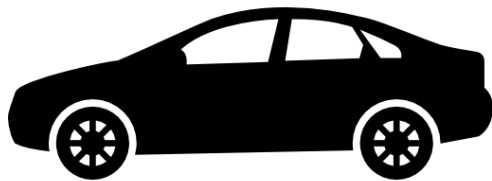
Metoder / Funktioner

A yellow pencil and a pink eraser are positioned in the top right corner of the slide, appearing as if they are on a piece of paper.

- Metoder (funktioner) tager parameter/argumenter som input
- Returner et output som enten en datatype, datastruktur eller objekt

Objekt-Orienteret Programmering

- Classes:
 - Et koncept eller en kategori af “ting”
- Objects:
 - En “instans” af en Class
- Eksempel:



Objekt-Orienteret Programmering 2



- En Class har følgende:
 - Attributes/Fields
 - Methods/Functions (Accessor/Mutator)
 - Constructor(s) (Destructor)
 - (Superclass)
- Skab et objekt:
 - `obj = myClass(arg1, arg2)`

Input/Output



- Læs fra konsolen:
 - `var = input()`
- Indlæs fil:
 - `f = open('myFile.txt', 'w')`
 - `f.read()`
 - `f.readline()`
 - `f.write()`
 - `f.writeline()`
- N.B., husk at lukke filer efter endt brug:
 - `f.close()`

Exceptions



- try:
 - Blok med med kode, som måske kan “kaste” en Exception/
Fejl
- except:
 - Fanger og behandler Exceptions/Fejl
- raise Error():
 - Smider en Exception/Fejl
- finally:
 - Blok med kode, der altid køres uanset fejl

“Kort” om Sortering og Søgning

A yellow pencil is positioned diagonally across the top right corner of the slide, pointing towards the bottom left. A pink eraser is placed below the pencil, also in the top right corner.

- Sortering:
 - Bubble Sort
 - Merge Sort
- Søgning:
 - Linear Search
 - Binary Search

SageMath



- SageMath benyttes som kode-baseret CAS (Computer Algebra System)
- Emulerer funktionalitet fra værktøjer som MatLab/GNU Octave, Maple, TI Nspire, osv.
- <https://doc.sagemath.org/html/en/tutorial/>

Numpy



- Matricer:

- `A = numpy.array([[2,1],[1,3]])`

Sympy



- Initialiser symbolsk variabel:
 - Python: `x = sympy.symbol('x')`
 - SageMath: `x = sym('x')`
- Lav symbolske funktioner:
 - $f(x, y) = x^2 + x*y + y^2$
- Evaluer symbolske funktioner numerisk:
 - `N(f(2, 3))`
 - `f.eval()`
-

Test og Fejlfinding



- Unit-test
 - Tester enkelte funktioner/metoder:
<https://docs.python.org/3/library/unittest.html>
- System-test
 - Tester systemet som helhed

Lidt om Software-Udvikling



Emner, som man kan gå videre med

- GUI: Tkinter, PyQt5
- Avancerede datastrukturer, eks: Linked-lists
- Socket programmering
- Django til webudvikling



django

Tak, skal I have!

- (Klap, klap, klap...)



Opgaver:



- Kod en basal lommeregner
- Tag en ny/gammel afleveringsopgave og se om du kan bruge Python/SageMath til at gøre arbejdet lettere
- Set dig ind i PyQt5 og lav en grafisk lommeregner med knapper: <http://zetcode.com/gui/pyqt5/>
- Se opgaver på datacamp:
<https://www.datacamp.com/courses/intro-to-python-for-data-science>



This work is licensed under a Creative Commons
Attribution-ShareAlike 3.0 Unported License.
It makes use of the works of Mateus Machado Luna.

