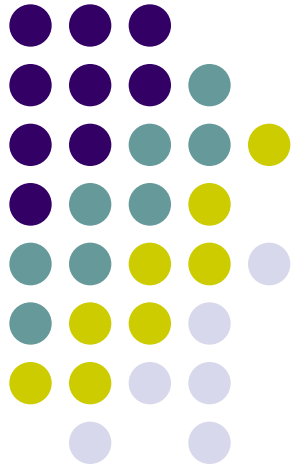
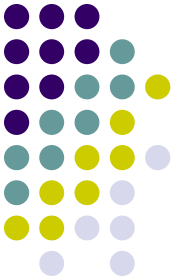
Programming in Python (NPRG065)

Ladislav Peška

Department of Software Engineering,
Charles University in Prague,
Czech Republic

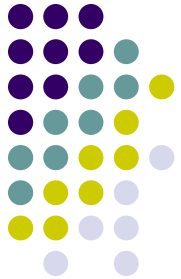


About NPRG065



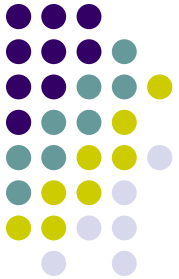
- Basics of Python
  - Focus on machine learning / data mining / data analysis etc.
  - Std. Python + common libraries
  - Numpy, Pandas, SciPy, Matplotlib, TensorFlow,…
- Rather „practical guide“ to what can be done and how
  - Not too much „language oriented“

About NPRG065



- Instructions to pass the subject:
  - Reasonable attendence + cooperation during lectures
    - No newspaper readers allowed☺
  - Homework: apply some machine learning on a defined problem (details later)
  - *Alternatively: create a lecture about a usage of Python for some particular task (upon agreement), e.g., Python&web, Python for linguistics…*
  - More complex homework for not attending students

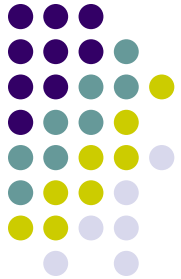
Disclaimer



- This is the first year of the subject
  - There will be bugs and errors, inevitably
  - Third-party slides will be used (do not re-invent the wheel)

- Rather an experienced user than an expert on both Python and ML

Scheduele



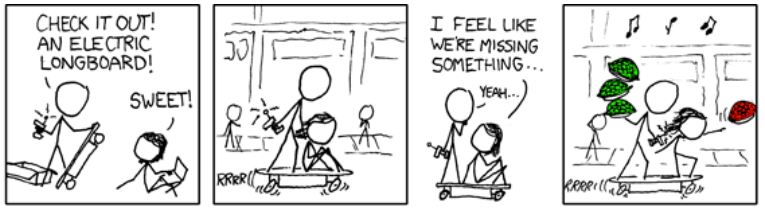
- Basics of Python & light intro into Machine Learning (2-3 lectures)
- NumPy, Pandas and I/O (2 lectures)
  - Matrices, Data Frames & effective data manipulation
- SciPy, Scikit-learn & some ML tasks and algorithms (2 lectures)
  - Learning algorithms, clustering, evaluation, metrics…
- Matplotlib & data visualization
- TensorFlow
  - Deep learning & „symbolic“ evaluation
- *Homework consultation + additional topics*

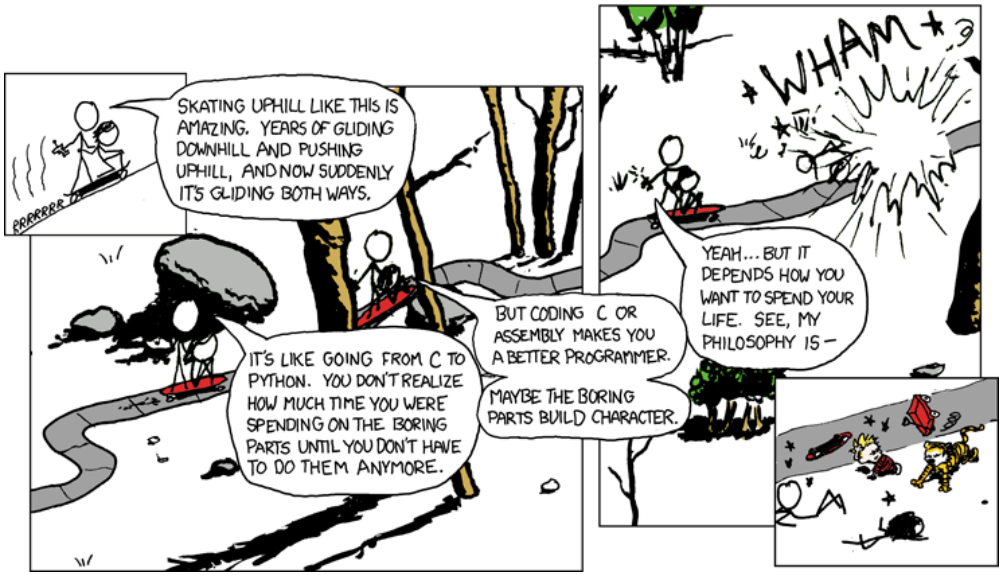
- Materials available from:

  www.ksi.mff.cuni.cz/~peska/vyuka/nprg065/

About Python

- Interpreted, object orientedlanguage
  - High-level, multi-platform
  - *Rapid development and prototyping*
  - Dynamically typed
  - Easy data manipulation
  - Rich bulit-in collection types
    - Lists
    - Tuples
    - Dictionaries
    - Sets





About Python

- Negatives
  - Native Python is slow (up to 100x compared to C)
    - There are simple workarounds
  - There are too many ways to handle the same problem
    - You can solve the problem, sure – but is this the best solution?
    - *It will "always" be a foreign language for you*
  - Type errors on runtime (due to dynamic typing)

Specifics of Python

- Intendation instead of braces

```
If a==0:                    If a==0:
    if b == 0:                  if b == 0:
        print (1)      !=           print (1)
    else:                       else:
        print (0)                   print (0)
```

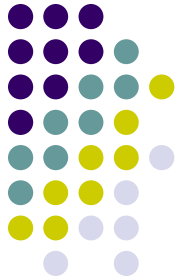
- Variables are references to an object
  - A.k.a passing values by reference in functions

- Many functions support data broadcasting

Versions of Python

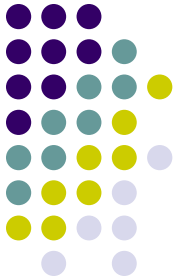
- Python 2.7
  - Strings are ASCII by default
  - !!!Integer divisions!!!
    - Int / int = int
    - 5 / 2 = 2
  - Older stuff (prior 2010) typically written in Python 2
- Python 3.6
  - Strings are unicode (čeština)
  - Sporadic function name changes (xrange -> rande)
  - Print syntax change
  - Some newer libraries do not support Python 2 (TensorFlow)

Install Python



- Several options – it is up to you
  - Anaconda 3
    - https://www.anaconda.com/download/
    - Contains Python & common scientific libraries
    - JuPyter Notebook (on-line interpreter, R-style development)
  - PyCharm IDE
    - https://www.jetbrains.com/pycharm/download
    - Code completion, navigation, inspector, debug, diff,… you name it☺

A Bit More Complicated Hello World



- Pepíček si řekl, že bude spořit. Založil si proto spořící účet u SuperBanky, která slibovala **úročení** vložené částky **každý den o 0.05%** a vložil na něj všechny své úspory: **250 korun**.
  Každý 30. den Pepíčkovi přišla výplata, ze které **přidal** na účet dalších **5 korun**.
  Takto si Pepíček spořil přesně 180 dní. Pak si ovšem zlomil nohu a aby mohl jít k doktorovi, musel si půjčit peníze u lichváře, pana Morella.
  Pan Morello dal Pepíčkovi velmi nevýhodné podmínky: každý měsíc (30 dní) **musí zaplatit 50 korun** po dobu příštích 12 měsíců. Pokud nezaplatí, tak mu prý pan Morello osobně věnuje betonové boty a pošle ho na dovolenou k moři. PS: se zlomenou nohou Pepíček nevydělává.

Napište program, který simuluje výše popsanou situaci.

- Vypište zůstatek na Pepíčkově účtu ke každému 30. dnu
- Pokud se Pepíčkovi nepodaří zaplatit, vypište *"Pepíček umřel x-tý den od úrazu"*
- Pokud se Pepíčkovi podaří zaplatit, vypište *"Pepíček splatil své dluhy a na účtu mu zbylo y korun"*