Python as a second language

Introduction to Python

Thank You

- Geert Jan Bex
  - https://github.com/gjbex/training-material/tree/master/Python
  - https://creativecommons.org/licenses/by/4.0/deed.ast#
- Slides + demo-files
  - https://franklbvp.github.io/python_intro/

What will be covered?

Content

- Environment
    - User environment
    - Command line
    - Spyder
    - Notebook
    - *Slides: Python_intro-userEnvironments*

Content

- Basic syntax
  - *Slides: Python_intro-programming-syntax*
- Variables
- Operators
  - *Slides: Python_intro-programming-variables_operators*
- Built-in atomic Data types
  - *Slides: Python_intro-programming-builtin_datatypes*
- Data structures
  - Lists
  - Tuples
  - Sets
  - Dictionaries
  - *Slides: Python_intro-programming-datastructures*

Content

- Flow control
    - Basic flow control commands
        - *Slides: Python_intro-programming-control_flow*
    - Functions
        - *Slides: Python_intro-programming-functions*
- IO
    - *Slides: Python_intro-programming-io*

Using the demo files

- Using the demo notebooks
- Download the zip file
- Unzip the file a local disk (Windows ex. in c:\Temp)
- Notebook
  - Start Anaconda powershell
    - ‘cd’ to the folder with ipynb files
    - `jupyter notebook`
- `py` source files
  - ‘cd’ to the folder with py files
  - Start editor / IDE
  - Run from command line

What will be covered in this session?

Scope

- Teach programming in Python
  - prerequisite: you should know how to program in some other language, if not consider first completing
    - CodeAcademy
      http://www.codecademy.com/tracks/python
    - LearnPython
      http://www.learnpython.org/
- *Taken from GJ Bex*

These sessions won't teach you how to program, how to find algorithms, that's beyond the scope

Running Python I

- Running Python from the command line
  - goals: run Python scripts in a shell
  - prerequisites: none
- Interactive Python
  - goals: using Python for explorative programming using iPython & Jupyter notebooks
  - prerequisites: none

Basic Python programming

- Core Python programming
  - goals: Python syntax & semantics, control flow, data types, functions
  - prerequisites: experience in some programming language

More in other courses

- Check https://github.com/gjbex/training-material/tree/master/Python
- Object oriented programming
  - goals: creating Python classes, inheritance
- Functional programming
  - goals: creating Python classes, inheritance
- Code organization
  - goals: organizing code of a non-trivial software project
- Documentation
  - goals: how to document Python code?

More in other courses

- Testing
  - goals: tests are integral part of software development
- Error handling
  - goals: catch & handle runtime errors
- Debugging
  - goals: using the Python debugger
- Profiling
  - goals: using the Python profiler to identify optimization opportunities

More in other courses

- Command line arguments & configuration files
  - goals: handling options, flags specified on command line, reading configuration files
- Logging
  - goals: writing application events to log files, using log levels
- Interacting with the operating system
  - goals: file system operations, executing external commands
- Web applications
  - goals: basic concepts of web application development

More in other courses

- Text-based formats
  - goals: reading & writing text-based file formats
- Scientific file formats
  - goals: reading & writing HDF5
- Linear algebra, numerical analysis
  - goals: various numerical analysis algorithms
- Scientific visualization
  - goals: creating 2D and 3D plots from Python