Programming

Introduction

Luna Pianesi

Faculty of Technology, Bielefeld University

Who am I?

- M.Sc. Luna Pianesi

  email: lpianesi@cebitec.uni-bielefeld.de
  office: UHG U10-121

- PhD student in the research group “Genome Data Science” headed by Prof. Dr. Alexander Schönhuth

  `https://gds.techfak.uni-bielefeld.de`

Organizational matters

What is Programming?

Overview of Python

Python Basics

Organizational matters

- Course prerequisites: *none*

Organizational matters

- Course prerequisites: *none*
- Coursework
  - Weekly exercises
    - Submission in groups of 2-3
    - Upload to corresponding assignment in the Moodle (former “LernraumPlus”)
    - Submission deadline is every ***Wednesday 14:00***.
  - Written exam on ***TBA***
    - Admitted: everyone exceeding 50% of total exercise points

Organizational matters

- Course prerequisites: *none*
- Coursework
  - Weekly exercises
    - Submission in groups of 2-3
    - Upload to corresponding assignment in the Moodle (former “LernraumPlus”)
    - Submission deadline is every ***Wednesday 14:00***.
  - Written exam on ***TBA***
    - Admitted: everyone exceeding 50% of total exercise points
- Lecture part of module *39-Inf-Pro “Programming”*, study program *Data Science*

Course material

- … available on course website: `https://gds.techfak.uni-bielefeld.de/teaching/2023winter/prog`
  - Slides and pointers to literature
  - Exercise sheets
- Moodle (former LernraumPlus):
  - e-Learning Videos
  - Exercise sheets
  - Pointers to literature
  - Forum
  - ***Weekly submission of exercise solutions***

Theory lectures

- Every ***Wednesday, 14:00-16:00***
- ZOOM meeting: `https://uni-bielefeld.zoom-x.de/j/67696699112?pwd=WVNSODhpTEIxVnB1elVUYU5UWWpWUT09` (passcode: 359913)
- Presentation of theoretical topics
- Discussion of questions and doubts

Tutorials

- Every ***Thursday, 10:00-12:00***
- Discussion of exercise solutions
- You will present solutions to your classmates

Literature

- VanderPlas, Jake. (2016). *Python data science handbook*. Beijing; Boston; Farnham; Sebastopol; Tokyo: O'Reilly.
- Toomey, Dan. (2017). *Jupyter for data science.* Birmingham; Mumbai: Packt.
- Ana Bell, Eric Grimson, John Guttag (2016) *MIT 6.0001 Introduction to Computer Science and Programming in Python*: `http://ocw.mit.edu/6-0001F16`
- Eric Grimson, John Guttag, Ana Bell (2016) *MIT 6.0002 Introduction to Computational Thinking and Data Science*: `http://ocw.mit.edu/6-0002F16`

Course syllabus

Part 1

- Programming basics and terminology
- Introduction to Python

Part 2

- Scientific Programming
- Data Science with Python

Organizational matters

What is Programming?

Overview of Python

Python Basics

What is a programming language?

- Natural vs. programming language
- Human-readable vs. machine-readable

Syntax and semantics

Syntax

Symbols, words, sentences

Semantics

Meaning behind symbols, words, and sentences

Syntax and semantics

Syntax

Symbols, words, sentences

e.g. English:

- Words: `He`, `She`, `It`, `Program`, …
- Sentence grammar rule:
  Subject + Verb + Object

`She loves Python` ✓

`The house table the cup` ✗

`The table reads the cup` ✓

Semantics

Meaning behind symbols, words, and sentences

Syntax and semantics

Syntax

Symbols, words, sentences

e.g. English:

- Words: `He`, `She`, `It`, `Program`, …
- Sentence grammar rule:
  Subject + Verb + Object

`She loves Python` ✓

`The house table the cup` ✗

`The table reads the cup` ✓

Semantics

Meaning behind symbols, words, and sentences

`She loves Python` ✓

`The table reads the cup` ✗

Programming languages …

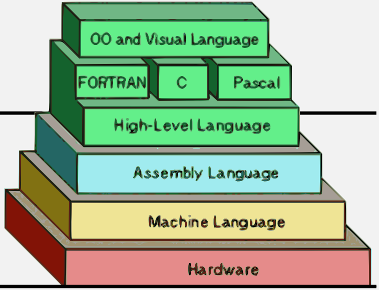
- are formal languages with unambiguous context-free grammars,

  syntactic ambiguity:
  “Tom hit the man with a stick.”

- offer different levels of abstraction,

High level language
- Easy to understand

Low level language
- Machine code
  (processor instructions
  and binary code)



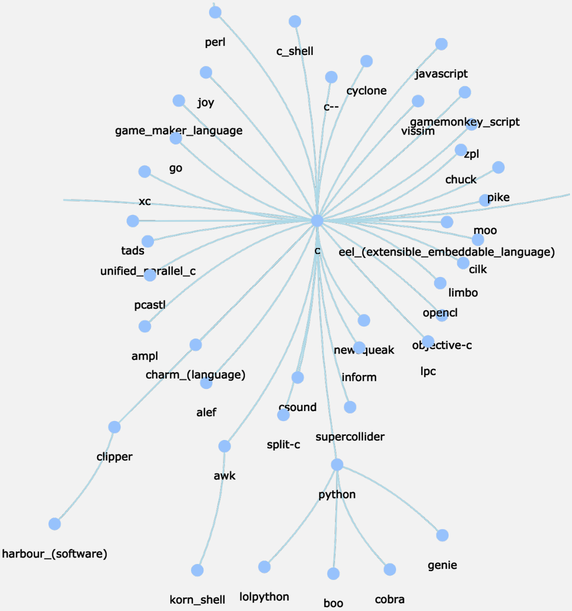
Source: `https://thebittheories.com`

Programming languages …

- are formal languages with unambiguous context-free grammars,

  syntactic ambiguity:
  “Tom hit the man with a stick.”

- offer different levels of abstraction,
- change over time,
- inspire new generations of languages.



Source: `http://svalver.github.io/Proglang/paradigms.html`

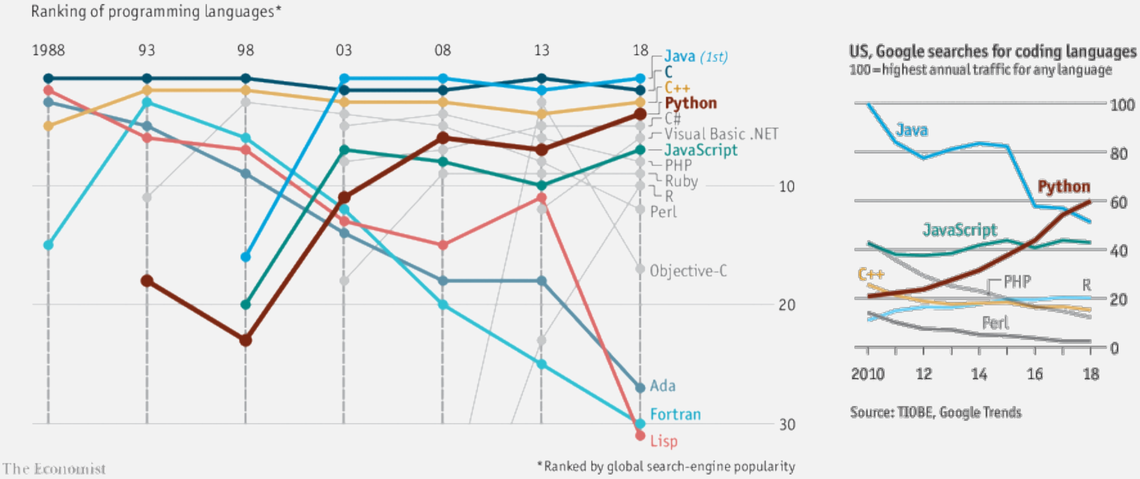
Programming paradigms

Many different general paradigms (notable excerpts):

- Imperative – *Do this, then do that!*
  - Procedural (C)
  - Object-oriented (C++, C#, Java)
- Declarative – *I want this, I want that!*
  - Logic (Prolog)
  - Functional (Haskell, Lisp)
- Mixed (Python, R)

There are also special-purpose languages (not necessarily considered “programming” languages), e.g. *LaTeX*, *HTML*, *XML*.

Language popularity



Source: Python is becoming the world's most popular coding language - The Economist (2018)

Quiz

- *Syntactic* or *semantic* ambiguity?
  - “Milk drinkers are turning to powder.”
  - “Stolen painting found by tree.”
  - “She went to her house, and so did Jane.”
- *True* or *false*?
  - “All context-free grammars are unambiguous.”
  - “Assembly language is a low level language.”
  - “Functional programming is a form of imperative programming.”

Quiz

- *Syntactic* or *semantic* ambiguity?

“Milk drinkers are turning to powder.”	syntactic
“Stolen painting found by tree.”	syntactic
“She went to her house, and so did Jane.”	semantic

- *True* or *false*?

“All context-free grammars are unambiguous.”	false
“Assembly language is a low level language.”	true
“Functional programming is a form of imperative programming.”	false

Organizational matters

What is Programming?

Overview of Python

Python Basics

Python

Originally developed by Guido van Rossum in the late 1980s.

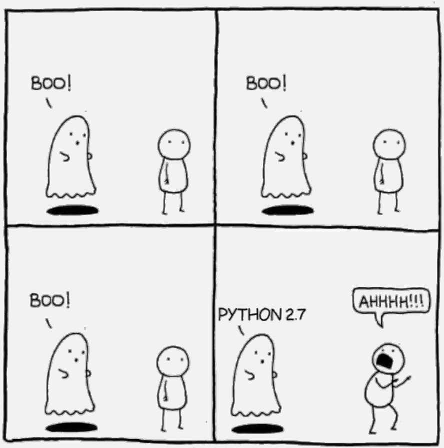
- Open-source and actively maintained
- Applicable to a wide range of applications
- Extremely popular in the data science community

But: There are alternative programming languages. Make sure to use the right one for the task.

Guido van Rossum, source: `https://gvanrossum.github.io`, ©Michael Cavotta, license: CC BY-NC-ND 4.0

Which Python version?

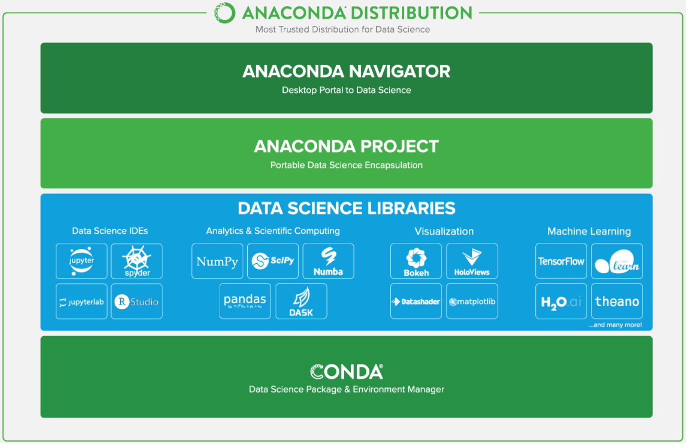
- Python 2: still common, although no longer maintained
- Python 3: modernized, *backwards-incompatible* version of the language



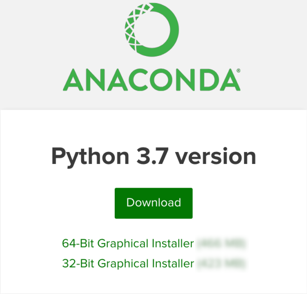
source: `https://www.reddit.com/r/ProgrammerHumor/comments/91vtas/python_27/`

Development environment: Anaconda

Python Data Science Distribution



Download Anaconda



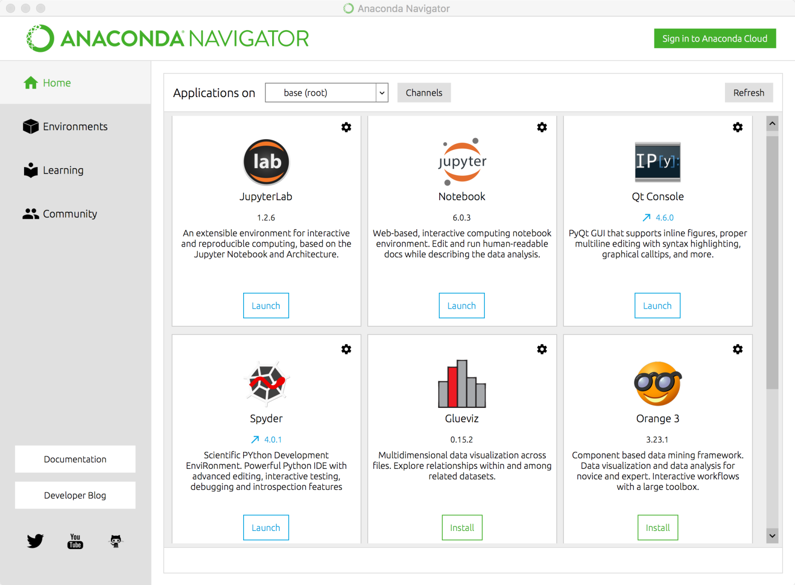
Python 3.7 version

64-Bit Graphical Installer [illegible]

32-Bit Graphical Installer [illegible]

`https://www.anaconda.com/distribution#download-section`

Anaconda navigator



Anaconda Navigator

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Home

Environments

Learning

Community

Documentation

Developer Blog

Applications on base (root) Channels Refresh

JupyterLab

1.2.6

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Launch

Notebook

6.0.3

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Launch

Qt Console

↗ 4.6.0

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Launch

Spyder

↗ 4.0.1

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Launch

Glueviz

0.15.2

Multidimensional data visualization across files. Explore relationships within and among related datasets.

Install

Orange 3

3.23.1

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Install

Quiz

- *True* or *false*?
  - “Python has been developed for data science analysis.”
  - “Python is the only language used in data science analysis.”
  - “The university has bought Python licenses for this course.”

Quiz

- *True* or *false*?
  - “Python has been developed for data science analysis.” false
  - “Python is the only language used in data science analysis.” false
  - “The university has bought Python licenses for this course.” false

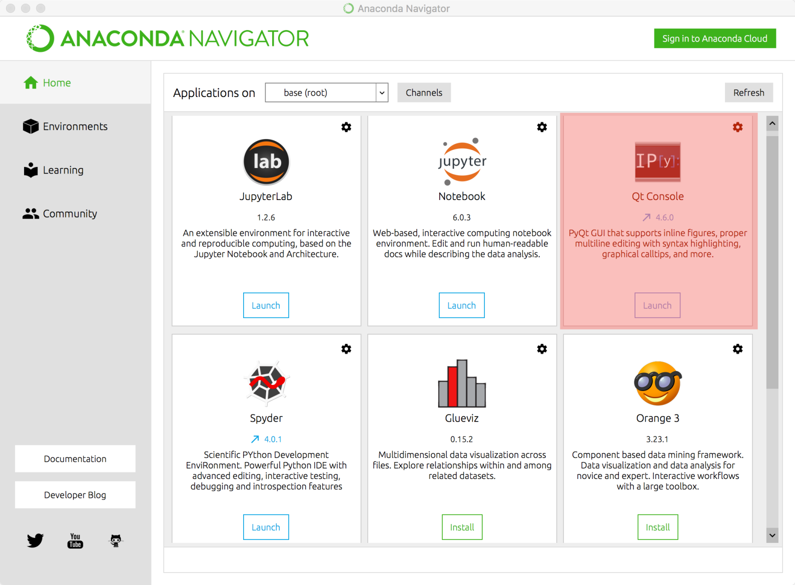
Organizational matters

What is Programming?

Overview of Python

Python Basics

Launching Python



Anaconda Navigator

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Home

Environments

Learning

Community

Documentation

Developer Blog

Applications on base (root)

Channels

Refresh

JupyterLab

1.2.6

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Launch

Notebook

6.0.3

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Launch

Qt Console

↗ 4.6.0

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Launch

Spyder

↗ 4.0.1

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Launch

Glueviz

0.15.2

Multidimensional data visualization across files. Explore relationships within and among related datasets.

Install

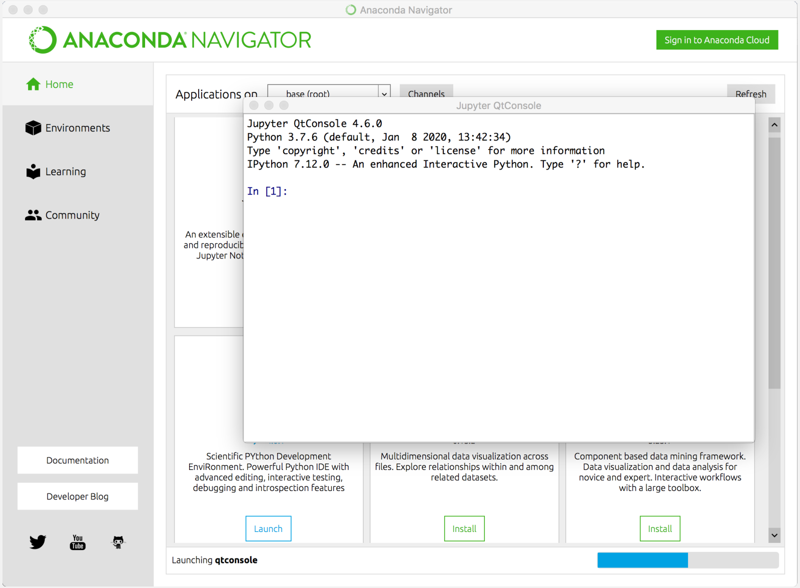
Orange 3

3.23.1

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Install

Launching Python



```
Jupyter QtConsole 4.6.0
Python 3.7.6 (default, Jan  8 2020, 13:42:34)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.12.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]:
```

An extensible and reproducib Jupyter Not

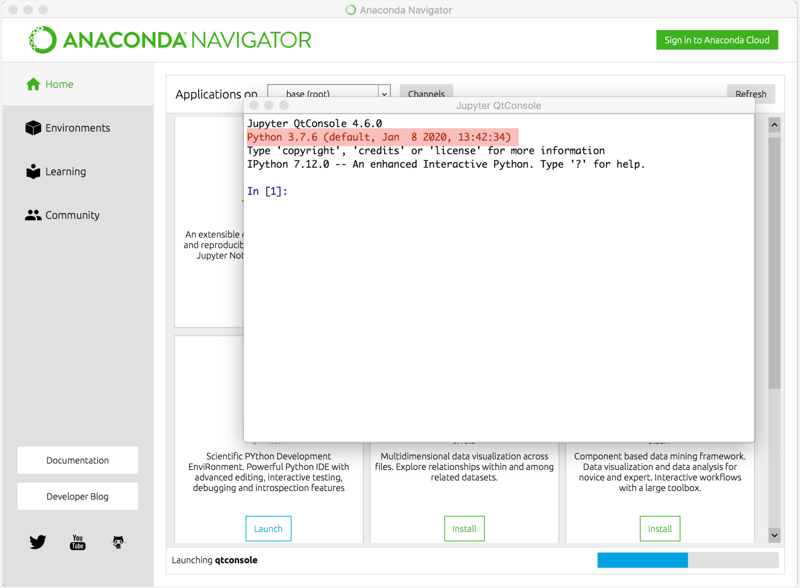
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Multidimensional data visualization across files. Explore relationships within and among related datasets.

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Launching **qtconsole**

Launching Python

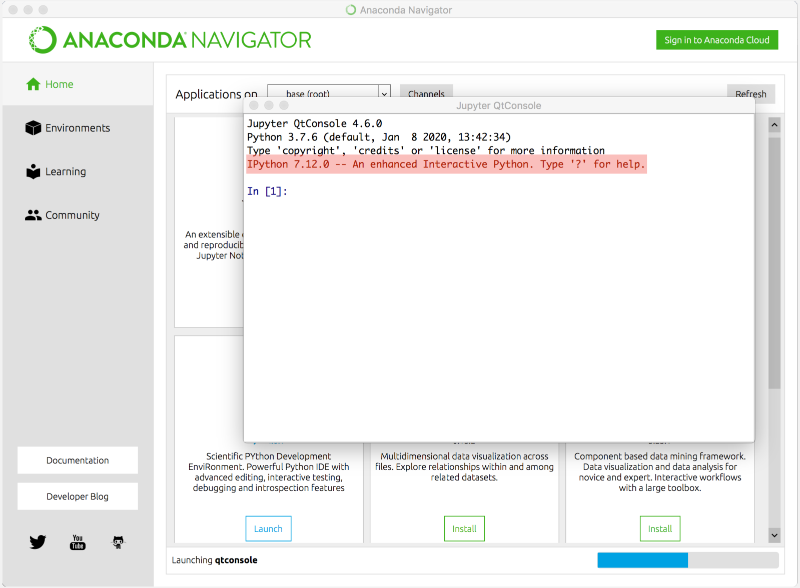


Anaconda Navigator
ANACONDA NAVIGATOR
Sign in to Anaconda Cloud
Home
Environments
Learning
Community
Documentation
Developer Blog
Applications on
base (root)
Channels
Refresh
Jupyter QtConsole
An extensible
and reproducib
Jupyter Not
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.
Launch
Install
Install
Launching qtconsole

```
Jupyter QtConsole 4.6.0
Python 3.7.6 (default, Jan  8 2020, 13:42:34)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.12.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]:
```

Launching Python



```
Jupyter QtConsole 4.6.0
Python 3.7.6 (default, Jan  8 2020, 13:42:34)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.12.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]:
```

An extensible
and reproducib
Jupyter Not

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Multidimensional data visualization across files. Explore relationships within and among related datasets.

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Launching **qtconsole**

Arithmetic in Python

Numeric types:

- Integer: `int` 42
- Real valued numbers: `float` 42.0
- Complex numbers: `complex` 42+0j

Operators

- Addition and subtraction `+ -`
- Multiplication and division `* / // %`
- Exponentiation `**`

Variables

Variable assignment

- `a = 42`
- `b = a - 6.0`

`type(«name of the variable»)`: returns type of variable

Quiz

- What are types numeric types of the following calculations?
  - `type(42 / 3)`
  - `type(42 // 3)`
  - `type(3.14 + 2.71+8j)`
  - `type(42 // 3.14)`

  -
    ```
    a = 1
    a - 10 * 1.0
    type(a)
    ```

Quiz

- What are types numeric types of the following calculations?
  - `type(42 / 3)` **float**
  - `type(42 // 3)` **int**
  - `type(3.14 + 2.71+8j)` **complex**
  - `type(42 // 3.14)` **float**

  -
    ```
    a = 1
    a - 10 * 1.0
    type(a)
    ```
    int

Recap

Summary

- Course logistics
- Introduction to Programming
- First steps in Python

What comes next?

- Go to the course website (`https://gds.techfak.uni-bielefeld.de/teaching/2023winter/prog`) or Moodle (`https://moodle.uni-bielefeld.de/course/view.php?id=3230`)and download this week's exercise sheet
- Group yourselves into pairs or (less recommended) triples
- Due date for this week's exercises is ***Wednesday, October 25, 2pm, 2023***.

Next lecture: Programming & Python basics continued ...