

Hydroinformatik II - SoSe 2025

UW-BHW-625-E: Python und Jupyter

Prof. Dr.-Ing. habil. Olaf Kolditz

¹Helmholtz Centre for Environmental Research – UFZ, Leipzig

²Technische Universität Dresden – TUD, Dresden

³Center for Advanced Water Research – CAWR

⁴TUBAF-UFZ Center for Environmental Geosciences – C-EGS, Freiberg / Leipzig

Dresden, 16.05.2025

0 Objekt-Orientierung und OOP

1 Python

2 Jupyter notebooks

3 Übungen

4 Ausblick auf die nächste Veranstaltung: Modellierung und Digitalisierung

Werkzeuge

- Editor: Notepad++
- Compiler: MinGW (C++)
- Python: Simulation und (einfache) Grafik
- Jupyter-Notebook: Workflows
- Visualisierung: ParaView
- Repository: Git

- Editor: Notepad++
- Compiler: MinGW (C++)
- **Python: Simulation und (einfache) Grafik**
- **Jupyter-Notebook: Workflows**
- Visualisierung: ParaView
- Repository: Git



Python

- "Python is a programming language that lets you work more quickly and integrate your systems more effectively."
- Webseite: <https://www.python.org>
- Vorteil: funktioniert auf allen Rechnern (>> Demo)



Python: Installation



The screenshot shows the Python.org website with a dark blue header and a lighter blue navigation bar. The header contains the Python logo, a search bar, and a 'Donate' button. The navigation bar includes links for 'About', 'Downloads', 'Documentation', 'Community', 'Success Stories', 'News', and 'Events'. The main content area features a large yellow button labeled 'Download Python 3.8.3' and a section titled 'Download the latest version for Windows'. Below this, there are links for other operating systems and pre-releases, and a note about Python 2.7.

Python

PSF

Docs

PyPI

Jobs

Community

python™

Donate

Search

GO

Socialize

About

Downloads

Documentation

Community

Success Stories

News

Events

Download the latest version for Windows

Download Python 3.8.3

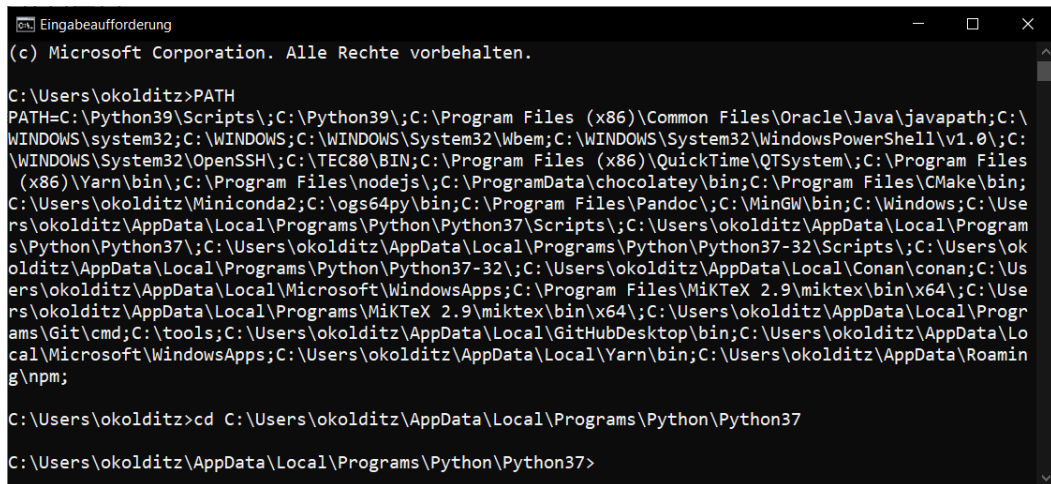
Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Prereleases](#), [Docker images](#)

Looking for Python 2.7? See below for specific releases



PATH



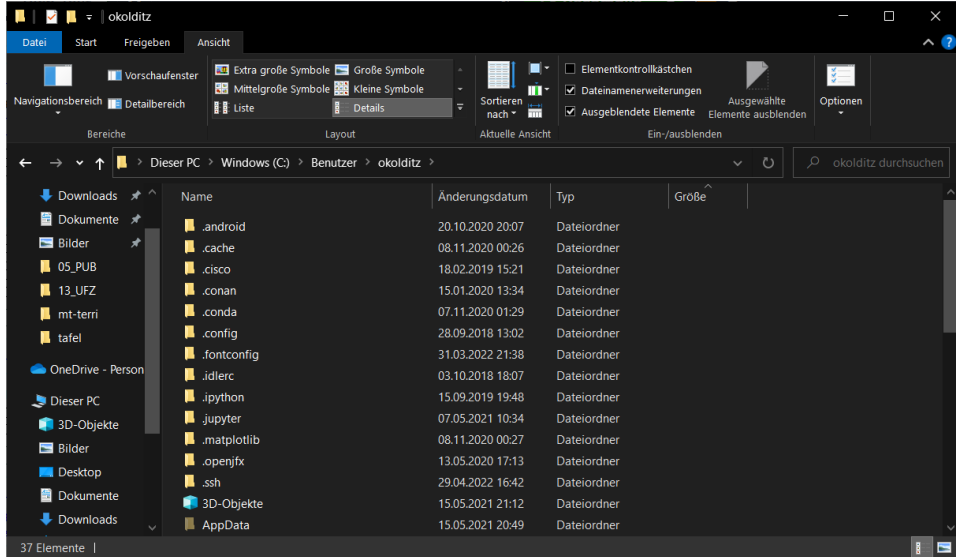
```
Eingabeaufforderung
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\okolditz>PATH
PATH=C:\Python39\Scripts\;C:\Python39\;C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\WINDOWS\System32\OpenSSH\;C:\TEC80\BIN;C:\Program Files (x86)\QuickTime\QTSystem\;C:\Program Files (x86)\Yarn\bin\;C:\Program Files\nodejs\;C:\ProgramData\chocolatey\bin;C:\Program Files\CMake\bin;C:\Users\okolditz\Miniconda2;C:\ogs64py\bin;C:\Program Files\Pandoc\;C:\MinGW\bin;C:\Windows;C:\Users\okolditz\AppData\Local\Programs\Python\Python37\Scripts\;C:\Users\okolditz\AppData\Local\Programs\Python\Python37\;C:\Users\okolditz\AppData\Local\Programs\Python\Python37-32\Scripts\;C:\Users\okolditz\AppData\Local\Programs\Python\Python37-32\;C:\Users\okolditz\AppData\Local\Conan\conan;C:\Users\okolditz\AppData\Local\Microsoft\WindowsApps;C:\Program Files\MiKTeX 2.9\miktex\bin\x64\;C:\Users\okolditz\AppData\Local\Programs\MiKTeX 2.9\miktex\bin\x64\;C:\Users\okolditz\AppData\Local\Programs\Git\cmd;C:\tools;C:\Users\okolditz\AppData\Local\GitHubDesktop\bin;C:\Users\okolditz\AppData\Local\Microsoft\WindowsApps;C:\Users\okolditz\AppData\Local\Yarn\bin;C:\Users\okolditz\AppData\Roaming\npm;

C:\Users\okolditz>cd C:\Users\okolditz\AppData\Local\Programs\Python\Python37

C:\Users\okolditz\AppData\Local\Programs\Python\Python37>
```

Python: Installation

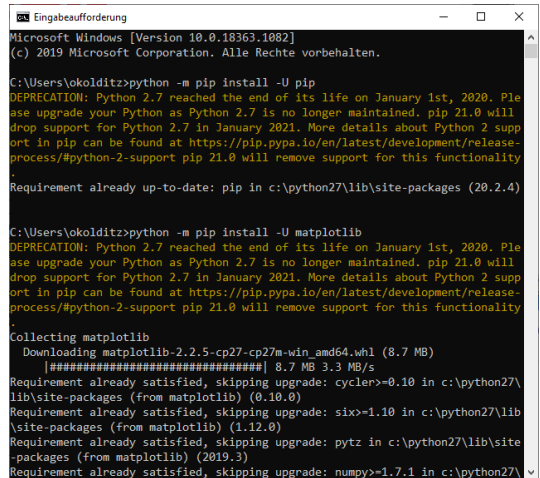


Python: Module installieren

matplotlib

- 1 `python -m pip install -U pip`
- 2 `python -m pip install -U matplotlib`

Listing: Installieren von der Konsole



```
Eingabeaufforderung
Microsoft Windows [Version 10.0.18363.1082]
(c) 2019 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\okolditz>python -m pip install -U pip
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 is no longer maintained. pip 21.0 will drop support for Python 2.7 in January 2021. More details about Python 2 support in pip can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support pip 21.0 will remove support for this functionality
Requirement already up-to-date: pip in c:\python27\lib\site-packages (20.2.4)

C:\Users\okolditz>python -m pip install -U matplotlib
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 is no longer maintained. pip 21.0 will drop support for Python 2.7 in January 2021. More details about Python 2 support in pip can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support pip 21.0 will remove support for this functionality
Collecting matplotlib
  Downloading matplotlib-2.2.5-cp27-cp27m-win_amd64.whl (8.7 MB)
    [#####] 8.7 MB 3.3 MB/s
Requirement already satisfied, skipping upgrade: cycler>=0.10 in c:\python27\lib\site-packages (from matplotlib) (0.10.0)
Requirement already satisfied, skipping upgrade: six>=1.10 in c:\python27\lib\site-packages (from matplotlib) (1.12.0)
Requirement already satisfied, skipping upgrade: pytz in c:\python27\lib\site-packages (from matplotlib) (2019.3)
Requirement already satisfied, skipping upgrade: numpy>=1.7.1 in c:\python27\lib\site-packages (from matplotlib) (1.17.0)
```

Python: Basics

Synthax

Import

Rechnen

Plotten

vgl. C++ Programme

Beispiel

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 #-----
4 year = np.arange(11)
5 numbers = [1,1,7,4,8,7,6,7,3,2,3]
6 #-----
7 fig, ax = plt.subplots()
8 ax.set_title('Hydroinformatik I 2019 - Notenspiegel')
9 ax.set_ylabel('Anzahl von Noten')
10 plt.bar(year, numbers)
11 plt.xticks(year, ('1.0', '1.3', '1.7', '...', '5.0'))
12 plt.grid(True)
13 plt.show()
```

Listing: Python example for bar charts: Hydroinformatik-I-Noten.py

Datentypen: EX04-data-types.py

Python-Basics

```
1 #boolean
2 d = True
3 print(d)
4 type(d)
5 print(type(d))
6 print(d.__sizeof__())
7 #integer
8 a = 3
9 print(a)
10 type(a)
11 print(type(a))
12 print(a.__sizeof__())
13 #float
14 b = 3.1415
15 print(b)
16 type(b)
17 print(type(b))
18 print(b.__sizeof__())
19 #string
20 c = 'Hydroinformatik'
21 print(c)
22 type(c)
23 print(type(c))
24 id(c)
25 print(id(c))
26 print(c.__sizeof__())
```

Listing: Exercise: Data types: Python

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout << "E04-data-types: Size of data types\n";
6     // system("pwd");
7     cout << "Type\tNumber of bytes\n";
8     cout << "-----\n";
9     cout << "bool\t\t" << sizeof(bool) << endl;
10    cout << "char\t\t" << sizeof(char) << "\n";
11    cout << "short\t\t" << sizeof(short) << endl;
12    cout << "int\t\t" << sizeof(int) << endl;
13    cout << "long\t\t" << sizeof(long) << endl;
14    cout << "float\t\t" << sizeof(float) << endl;
15    cout << "double\t\t" << sizeof(double) << endl;
16    cout << "long double\t" << sizeof(long double) <<
        endl;
17    return 0;
18 }
```

Listing: Exercise: Data types: C++

Felder anlegen

```
1 import numpy as np
2
3 #dynamische Felder
4 x = [] #leeres Feld
5 y = [] #leeres Feld
6
7 #statische Felder
8 year = np.arange(11) #x
9 numbers = [1,1,7,4,8,7,6,7,3,2,3] #y
10
11 #mehrdimensionale Felder
12 dim = 20
13 xarray = np.arange(-dim,dim)
14 yarray = np.arange(-dim,dim)
```

Listing: Python: Arrays

Felder füllen

```
1 for i in range(0,n):
2     x.append(float(i)/float(n))
```

Listing: Python: Arrays

```
1 #feste Schleife
2 t = [0.1,0.5,1.0,2.0]
3 for i in t:
4     x.append(i*i)
5
6 #dynamische Schleife
7 for i in range(0,n):
8     x.append(float(i)/float(n))
```

...

Listing: Python: Loops

```
1 class Student:
2     name = "Maxi Musterfrau"
3     matrikel = 123456789
4     def printnamen(self):
5         print("Name " + name)
6     def printmatrikel(self):
7         print("Matrikel " + str(123456789))
```

Listing: Python: Klassen

```
1 #Instanzen
2 Suse = Student()
3 Suse.name(Suse)
4 Suse.matrikel(192837465)
5 Suse.print()
6
7 Max = Student()
8 Max.name(Max)
9 Max.matrikel(987654321)
10 Max.print()
```

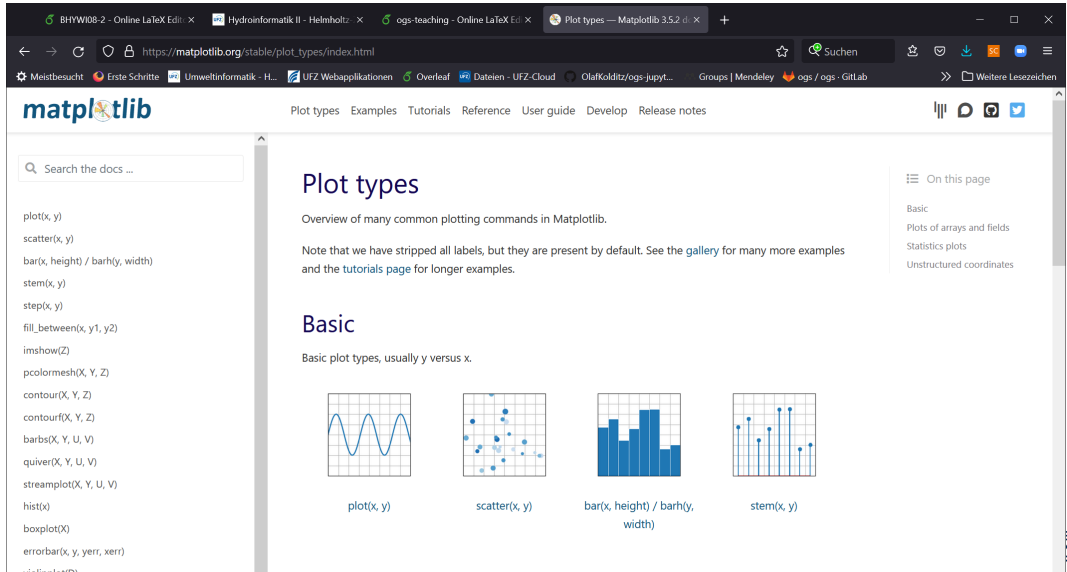
Listing: Python: Klassen

```
1 import matplotlib.pyplot as plt
2
3 fig, ax = plt.subplots()
4 ax.set_title('Hydroinformatik I 2019 - Notenspiegel')
5 ax.set_ylabel('Anzahl von Noten')
6
7 plt.bar(marks, numbers)
8 plt.xticks(marks, ('1.0', '1.3', '1.7', '2.0', '2.3', ..., '5.0'))
9 plt.grid(True)
10 plt.show()
```

...

Listing: Python: Plotten

Python: Grafik



The screenshot shows the Matplotlib website's 'Plot types' page. The browser's address bar displays https://matplotlib.org/stable/plot_types/index.html. The page features a navigation bar with links to 'Plot types', 'Examples', 'Tutorials', 'Reference', 'User guide', 'Develop', and 'Release notes'. A search bar is located on the left side of the page. The main content area is titled 'Plot types' and provides an overview of common plotting commands. It includes a note about default labels and links to a 'gallery' and 'tutorials' page. Below this, the 'Basic' section is introduced, showing four example plots: a line plot, a scatter plot, a bar chart, and a stem plot. The left sidebar lists various plot types, including `plot(x, y)`, `scatter(x, y)`, `bar(x, height) / barh(y, width)`, `stem(x, y)`, `step(x, y)`, `fill_between(x, y1, y2)`, `imshow(Z)`, `pcolormesh(X, Y, Z)`, `contour(X, Y, Z)`, `contourf(X, Y, Z)`, `barbs(X, Y, U, V)`, `quiver(X, Y, U, V)`, `streamplot(X, Y, U, V)`, `hist(x)`, `boxplot(X)`, and `errorbar(x, y, yerr, xerr)`.

Search the docs ...

plot(x, y)
scatter(x, y)
bar(x, height) / barh(y, width)
stem(x, y)
step(x, y)
fill_between(x, y1, y2)
imshow(Z)
pcolormesh(X, Y, Z)
contour(X, Y, Z)
contourf(X, Y, Z)
barbs(X, Y, U, V)
quiver(X, Y, U, V)
streamplot(X, Y, U, V)
hist(x)
boxplot(X)
errorbar(x, y, yerr, xerr)

Plot types

Overview of many common plotting commands in Matplotlib.

Note that we have stripped all labels, but they are present by default. See the [gallery](#) for many more examples and the [tutorials page](#) for longer examples.

Basic

Basic plot types, usually y versus x.

plot(x, y)

scatter(x, y)

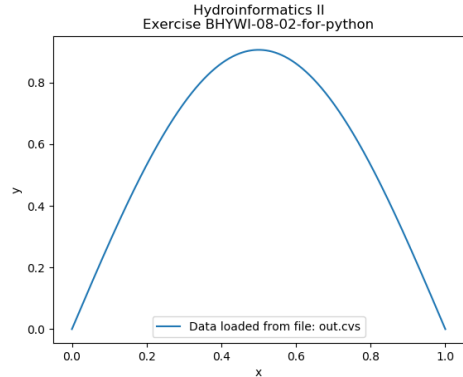
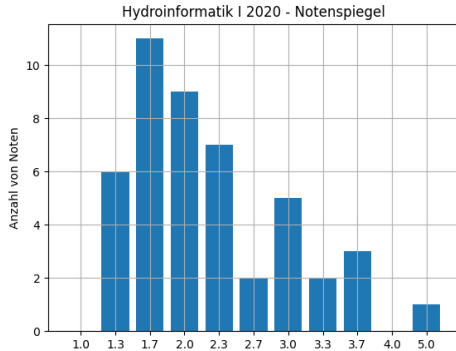
bar(x, height) / barh(y, width)

stem(x, y)

On this page

- Basic
- Plots of arrays and fields
- Statistics plots
- Unstructured coordinates

Python: Plotting (matplotlib)



Python: Plotting bar charts

Übung: Hydroinformatik-I-Noten.py

```
1 from matplotlib.ticker import FuncFormatter
2 import matplotlib.pyplot as plt
3 import numpy as np
4 #-----
5 year = np.arange(11)
6 numbers = [1,1,7,4,8,7,6,7,3,2,3]
7 #-----
8 fig, ax = plt.subplots()
9 ax.set_title('Hydroinformatik I 2019 - Notenspiegel')
10 ax.set_ylabel('Anzahl von Noten')
11 plt.bar(year, numbers)
12 plt.xticks(year, ('1.0', '1.3', '1.7', '2.0', '2.3', '2.7', '3.0', '3.3', '3.7',
13                    '4.0', '5.0'))
13 plt.grid(True)
14 plt.show()
```

Listing: Python example for bar charts: Hydroinformatik-I-Noten.py

Diffusionsgleichung: Analytische Lösung Tutorial: Abschnitt 2.2

- Repo: BHYWI-08-02-E-Script
- C++ Variante
- C++ / Python Variante
- Python Variante

MinGW + Python: Function plotter (analytical solution) #1

Übung: BHYWI-08-02-E-Script: Skript

```
1 echo Compilation
2 g++ main.cpp
3 echo Execution
4 a.exe
5 echo Ploting
6 data_from_file.py
7 echo End
```

Listing: Script file for entire workflow

MinGW + Python: Function plotter (analytical solution) #2

Übung: BHYWI-08-02-E-Script: C++

```
1 #include <cmath>
2 #include <fstream>
3 #define PI 3.14159265358979323846
4 int main(int argc, char *argv[])
5 {
6     //1-Definitionen
7     int numPoints = 1000;
8     double x,y,alpha=1.,t=0.01;
9     std::ofstream out_file;
10    out_file.open("out.csv");
11    //2-Berechnung
12    for (int i = 0; i < numPoints+1; ++i)
13    {
14        x = double(i)/double(numPoints);
15        y=sin(PI*x) * exp(-alpha*PI*PI*t);
16        out_file << x << "," << y << std::endl;
17    }
18    //3-Ausgabe
19 }
```

Listing: C++ program for analytical solution

MinGW + Python: Function plotter (analytical solution) #3

Übung: BHYWI-08-02-E-Script: File reading and data plotting

```
1 import matplotlib.pyplot as plt
2 import csv
3
4 x = []
5 y = []
6 with open('out.csv','r') as csvfile:
7     plots = csv.reader(csvfile, delimiter=',')
8     for row in plots:
9         x.append(float(row[0]))
10        y.append(float(row[1]))
11
12 plt.plot(x,y, label='Data loaded from file: out.csv')
13 plt.xlabel('x')
14 plt.ylabel('y')
15 plt.title('Hydroinformatics II\nExercise BHYWI-08-02-for-python')
16 plt.legend()
17 plt.savefig("diffusion-equation.png.png")
18 plt.show()
```

Listing: File reading and data plotting

```
1 import math
2 import matplotlib.pyplot as plt
3 PI = 3.14159265358979323846
4 numPoints = 10
5 alpha = 1.0
6 t = [0.1,0.5,1.0,2.0]
7 x = []
8 y = []
9 for n in t:
10     for i in range(0,numPoints+1):
11         x.append(float(i)/float(numPoints))
12         y.append(math.sin(PI*x[i]) * math.exp(-alpha*n*n))
13     plt.plot(x,y,color='red',marker="o")
14     x = []
15     y = []
16 plt.xlabel('x')
17 plt.ylabel('u')
18 plt.savefig("diffusion-equation.png")
19 plt.show()
```

Listing: Analytical solution

ChatGPT (OpenAI)

ChatGPT (Generative Pre-trained Transformer) is a large language model chatbot developed by OpenAI that can hold conversations and generate human-like text. It uses natural language processing and machine learning to understand and respond to user prompts in a variety of formats. ChatGPT can be used for tasks like answering questions, writing different types of content, translating languages, and even interacting with computers.

<https://chatgpt.com>

DeepSeek (open source AI)

DeepSeek is a Chinese AI startup (Liang Wenfeng) that has gained significant attention for developing competitive AI models at a fraction of the cost of established players like OpenAI. They have released open-source models like DeepSeek-R1, demonstrating their capabilities while also pushing the boundaries of AI development cost-effectiveness.

>> cost and energy efficiency

<https://chat.deepseek.com/>

- ▶ data types
- ▶ arrays
- ▶ ...

...

??

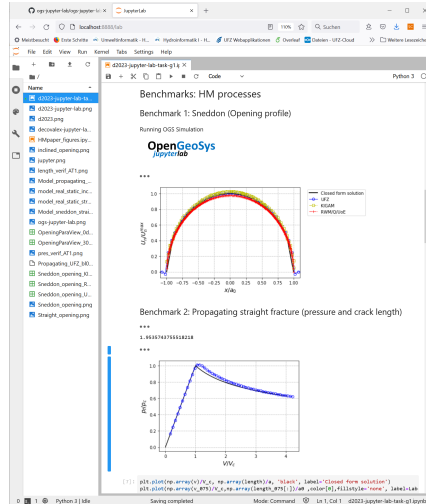
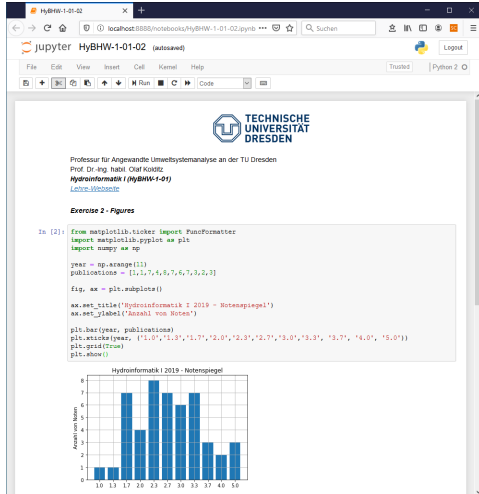
Jupyter

- Jupyter Notebook
- Jupyter Lab
- Browser-basiert

- "The Jupyter Notebook · The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, ..."
- Webseite: <https://jupyter.org/>
- Vorteil: funktioniert auf allen Rechnern
- ... ein Teil unserer (neuen) Übungen machen wir mit Jupyter Notebooks (>> Demo)



Jupyter: Example



`https://jupyter.org/install`

Desk und Web-Anwendungen

