

Hydroinformatik - SoSe 2026

UW-BHW-414-06: Programmiersprache Python

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, 15.05.2026

Zeitplan: Hydroinformatik I+II

Sommersemester 2026: Stand: 06.04.2026

Nr.	KW	Datum	ID	Thema
01+02	16	17.04.2026	UW-BHW-414-01/02	Einführung in die Vorlesung, Umweltinformatik
03	16	17.04.2026	UW-BHW-414-03	Werkzeuge, Hello World (in C++)
05	17	24.04.2026	UW-BHW-414-04	Selbststudium: Software-Installationen
07	19	08.05.2026	UW-BHW-414-05	Objekt-Orientierte Programmierung: C++, Klassen
09	20	15.05.2026	UW-BHW-414-06	Programmiersprache Python
11	21	22.05.2026	UW-BHW-414-F	Modellierung, Digitalisierung, Wasser 4.0
00	22	29.05.2026		Vorlesungsfreie Woche
13	23	05.06.2026	UW-BHW-414-G	KI, Maschinelles Lernen, Neuronale Netzwerke
15	24	12.06.2026	UW-BHW-414-H	Kontinuumsmechanik, Hydromechanik
17	25	19.06.2026	UW-BHW-414-I	Differentialgleichungen, Näherungsverfahren
19	26	26.06.2026	UW-BHW-414-J	Finite-Differenzen, explizite Verfahren
21	27	03.07.2026	UW-BHW-414-K	Finite-Differenzen, implizite Verfahren
23	28	10.07.2026	UW-BHW-414-L	Gerinnehydraulik, Grundwasserhydraulik
25	29	17.07.2026	UW-BHW-414-M	Grundwasserhydraulik
27	30	24.07.2026	UW-BHW-414-N	Zusammenfassung, Klausurvorbereitung

1 UW-BHW-414-06: Programmiersprache Python

- Semesterplan
- About
- Installation
- Basics
- Grafik

2 Python

3 Jupyter

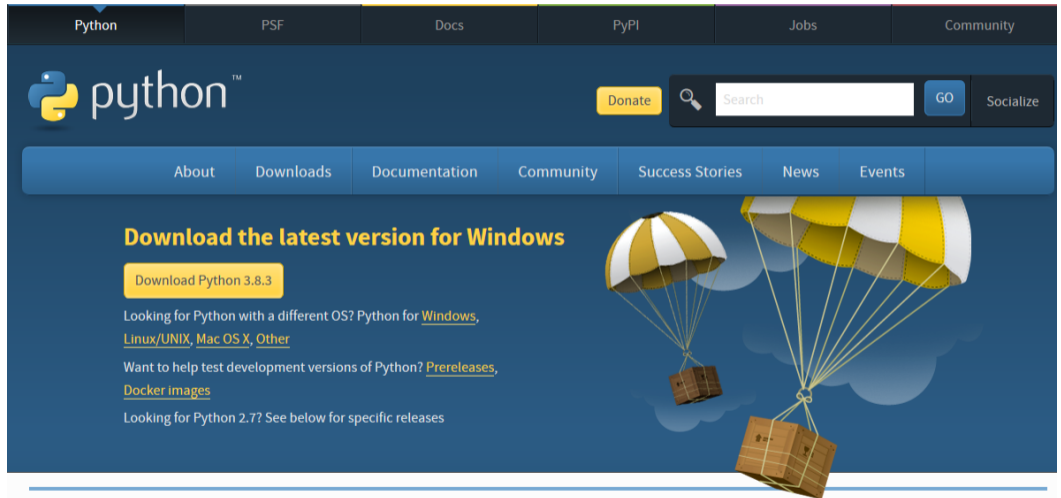
4 Übungen

Python

- ▶ About
- ▶ Installation


- "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)





The screenshot shows the Python.org website with a dark blue header and navigation menu. The main content area is also dark blue and features a large illustration of two cardboard boxes being lowered by yellow and white striped parachutes against a blue sky with white clouds. The text on the page is white and yellow.

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

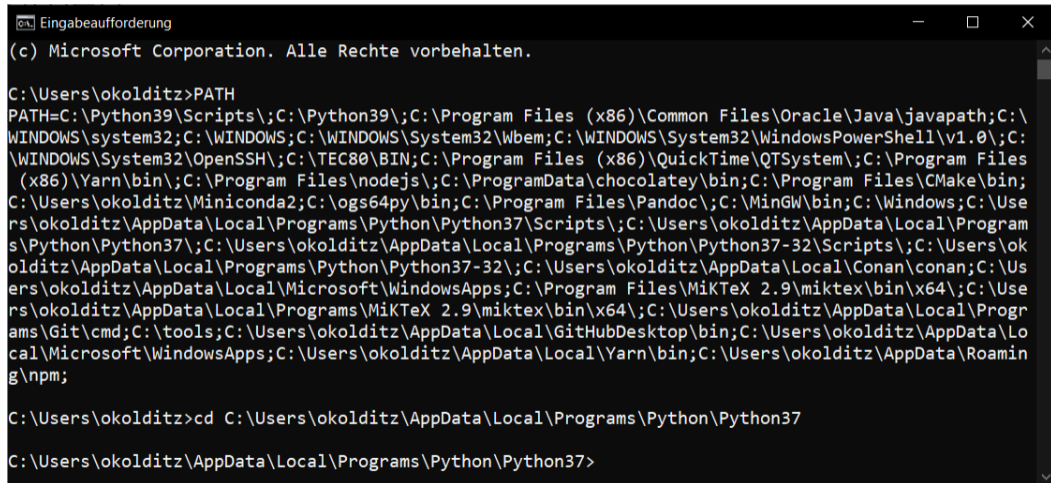
[Download Python 3.8.3](#)

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

Want to help test development versions of Python? [Preleases](#), [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:\Use
rs\okolditz\AppData\Local\Programs\Python\Python37\Scripts\;C:\Users\okolditz\AppData\Local\Program
s\Python\Python37\;C:\Users\okolditz\AppData\Local\Programs\Python\Python37-32\Scripts\;C:\Users\ok
olditz\AppData\Local\Programs\Python\Python37-32\;C:\Users\okolditz\AppData\Local\Conan\conan;C:\Us
ers\okolditz\AppData\Local\Microsoft\WindowsApps;C:\Program Files\MiKTeX 2.9\miktex\bin\x64\;C:\Use
rs\okolditz\AppData\Local\Programs\MiKTeX 2.9\miktex\bin\x64\;C:\Users\okolditz\AppData\Local\Progr
ams\Git\cmd;C:\tools;C:\Users\okolditz\AppData\Local\GitHubDesktop\bin;C:\Users\okolditz\AppData\Lo
cal\Microsoft\WindowsApps;C:\Users\okolditz\AppData\Local\Yarn\bin;C:\Users\okolditz\AppData\Roamin
g\npm;

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

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

Python: Installation

The screenshot shows a Windows File Explorer window for the user 'okolditz'. The address bar indicates the path: 'Dieser PC > Windows (C:) > Benutzer > okolditz'. The left sidebar shows the navigation pane with 'Dieser PC' selected. The main area displays a list of hidden folders with columns for Name, Änderungsdatum, Typ, and Größe. The folders listed are: .android, .cache, .cisco, .conan, .conda, .config, .fontconfig, .idlerc, .ipython, .jupyter, .matplotlib, .openjfx, .ssh, 3D-Objekte, and AppData. The status bar at the bottom indicates '37 Elemente'.

Name	Änderungsdatum	Typ	Größe
.android	20.10.2020 20:07	Dateiordner	
.cache	08.11.2020 00:26	Dateiordner	
.cisco	18.02.2019 15:21	Dateiordner	
.conan	15.01.2020 13:34	Dateiordner	
.conda	07.11.2020 01:29	Dateiordner	
.config	28.09.2018 13:02	Dateiordner	
.fontconfig	31.03.2022 21:38	Dateiordner	
.idlerc	03.10.2018 18:07	Dateiordner	
.ipython	15.09.2019 19:48	Dateiordner	
.jupyter	07.05.2021 10:34	Dateiordner	
.matplotlib	08.11.2020 00:27	Dateiordner	
.openjfx	13.05.2020 17:13	Dateiordner	
.ssh	29.04.2022 16:42	Dateiordner	
3D-Objekte	15.05.2021 21:12	Dateiordner	
AppData	15.05.2021 20:49	Dateiordner	

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\
```

Python

- ▶ Basics

Syntax

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: EX05-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     def __init__(self, name, matrikel):
3         self.name = name
4         self.matrikel = matrikel
5
6     def print_name(self):
7         print("Name " + self.name)
8
9     def print_matrikel(self):
10        print("Matrikel " + str(self.
matrikel))
11
12    def print(self):
13        self.print_name()
14        self.print_matrikel()
```

Listing: Python: Klassen: Definition

```
1 # Create and use instances
2
3 Suse = Student("Suse",
4               192837465)
5 Suse.print()
6 Max = Student("Max",
7              987654321)
8 Max.print()
```

Listing: Python: Klassen: Instanzen

Plotting: Balkendiagramm

Python-Basics

```
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

matplotlib

Plot types Examples Tutorials Reference User guide Develop Release notes

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)
- pie(labels, values)





Plot types

Overview of many common plotting commands in Matplotlib.

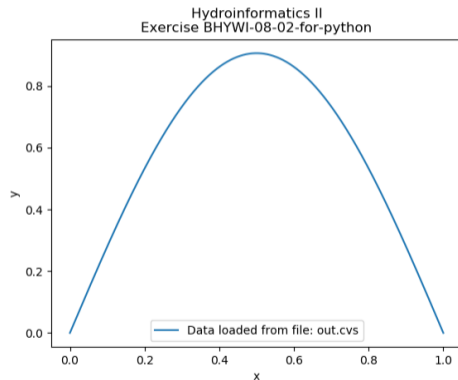
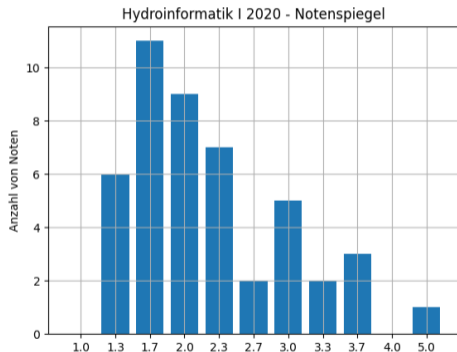
Note that we have stripped all labels, but they are present by default. See the [gallery](#) for many more examples and the [tutorials](#) page for longer examples.

Basic

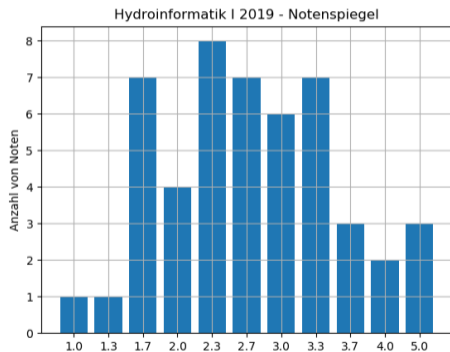
Basic plot types, usually y versus x.

-  plot(x, y)
-  scatter(x, y)
-  bar(x, height) / barh(y, width)
-  stem(x, y)

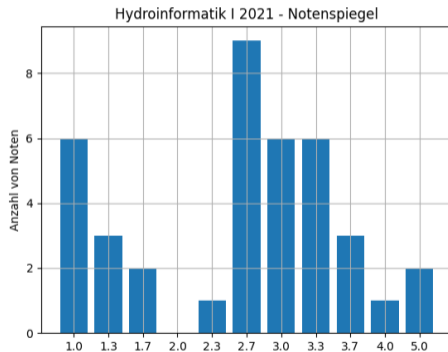
Installation zusätzlicher Module: Seite 9



Normalverteilung (Gauss)



Bi-modale Verteilung



Balkendiagramm (bar charts): EX07-bar-chart.py

Python: Plotting

```
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: EX06-bar-chart.py

Zusammenfassung

- ▶ Python: About und Installation (einschl. Module)
- ▶ Python: Basics
 - ▶ Datentypen
 - ▶ Felder (Arrays), Schleifen
 - ▶ Klassen: Definition und Instanzen
- ▶ Python: Plotten mit `matplotlib`
 - ▶ Balkendiagramm

⇒ Funktionen: Rechnen und Plotten

Diffusionsgleichung: Analytische Lösung Tutorial: Abschnitt 2.2

- Repo: EX08-function
- C++ Variante
- C++ / Python Variante
- Python Variante

```
1 echo Compilation
2 g++ Ex08a-function.cpp
3 echo Execution
4 a.exe
5 echo Plotting
6 python EX08b-plot_data_from_file.py
7 echo End
```

Listing: Script file for entire workflow

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

Übung: EX08-function

```
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

Python: Function plotter (analytical solution) #3

Übung: EX08b-plot-data-from-file.py

```
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/>

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

Technische Universität Dresden

Prof. Dr.-Ing. habil. Olof Klotze
Hydroinformatik I (HyBHW-1-01)
[Lehrer-Homepage](#)

Exercise 2 - Figures

```
In [2]: from matplotlib.ticker import FuncFormatter
import matplotlib.pyplot as plt
import numpy as np

year = np.arange(11)
publications = [1,1,7,4,8,7,6,7,3,2,3]

fig, ax = plt.subplots()

ax.set_title('Hydroinformatik I 2019 - Notenspiegel')
ax.set_ylabel('Anzahl von Notizen')

plt.bar(year, publications)
plt.xticks(year, ('1.0', '1.3', '1.7', '2.0', '2.3', '2.7', '3.0', '3.3', '3.7', '4.0', '5.0'))
plt.grid(True)
plt.show()
```

Hydroinformatik I 2019 - Notenspiegel

Note	Anzahl von Notizen
1.0	1
1.3	1
1.7	7
2.0	4
2.3	8
2.7	7
3.0	6
3.3	7
3.7	3
4.0	2
5.0	3

Benchmarks: HM processes

Benchmark 1: Sneddon (Opening profile)

Running OGS Simulation

OpenGeoSys
jupyterlab

U₀/U₀^{*}

X/a

Legend: Closed form solution, UFZ, KITAM, RWMO/Infat

Benchmark 2: Propagating straight fracture (pressure and crack length)

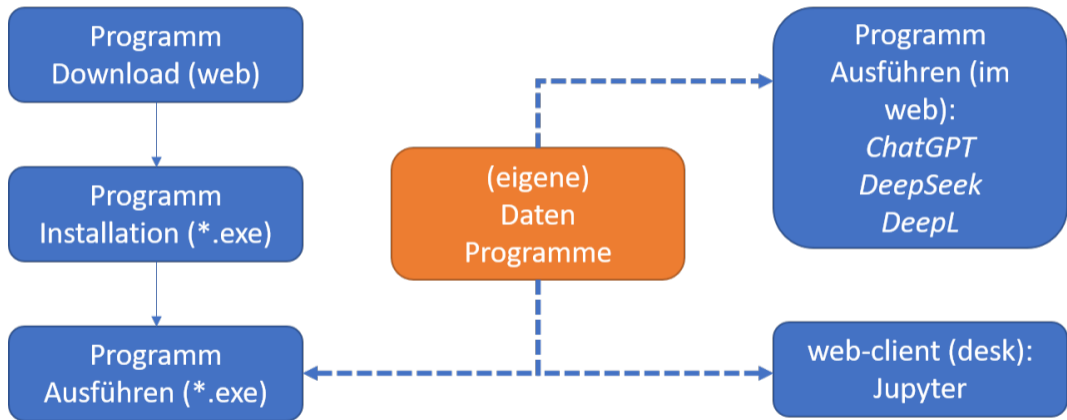
1.9689748756558228

p/p₀

W/W₀

```
[7]: plt.plot(np.array(0)/x_c, np.array(length)/w, 'black', label='Closed form solution')
plt.plot(np.array(x_075)/x_c, np.array(length_075)/w, 'blue', label='UFZ')
plt.plot(np.array(x_075)/x_c, np.array(length_075)/w, 'red', label='KITAM')
plt.plot(np.array(x_075)/x_c, np.array(length_075)/w, 'green', label='RWMO/Infat')
```

`https://jupyter.org/install`



Jupyter Notebook

- ▶ Zusammenführen von C++ und Python Funktionen

Zusammenführen von Funktionen (C++ und Python)

Jupyter Notebooks

Hydroinformatik-UW BHW-414

EX09-jupyter-notebook

localhost:8888/notebooks/EX09-jupyter-not... Zusammenfassen

Overleaf videoconferenes ufz projects pvt gitlab pub events tmp

Jupyter EX09-jupyter-notebook Last Checkpoint: 7 minutes ago

File Edit View Run Kernel Settings Help Trusted

Markdown JupyterLab Python 3 (ipykernel)

Technische Universität Dresden

Professur für Angewandte Umweltsystemanalyse an der TU Dresden
Prof. Dr.-Ing. habil. Olaf Kolditz
Hydroinformatik (UW-BHW-414)
[Lehre-Webseite](#)

Übung: EX09 Jupyter Notebook

In dieser Übung werden die beiden letzten Übungen, Funktionsberechnungen mit C++ und Python, in einem Jupyter Notebook zusammengeführt. Dies zeigt einen der Vorteile von Jupyter Notebook, in dem verschiedene Aufgaben in einer Web-Umgebung zusammengeführt und ausgeführt werden.

Inhalt:

- Funktionsberechnung mit dem C++ code und Ergebnisse als Datei schreiben
- Funktionsberechnung mit dem Python code
- Vergleich der Ergebnisse
- KI Vorschlag für Funktionsberechnungen und Vergleich

EX08a: Funktionsberechnung mit C++

```
[5]: lg++ EX08a-function.cpp  
la.exe
```

Jupyter Notebooks:

- ▶ internet-fähig
- ▶ funktionieren zellenbasiert, Daten bleiben erhalten
- ▶ Beschreibung und Berechnungen, grafische Darstellungen
- ▶ ...

Übungen

Informatik

- EX01: C++ "Hello World"
- EX02: C++ Datentypen
- EX03: C++ Klassen
- EX04: C++ Datenbank
- EX05: Python Datentypen
- EX06: Python Klassen
- EX07: Python Balkendiagramme
- EX08: Funktionen: C++, Python
- EX09: Jupyter Notebooks
- ...

Übungen

Mechanik

- EX20: Jupyter Notebook
- EX21: Kontinuumsmechanik: Skalarprodukt
- EX22: Hydromechanik: Divergenzfreie Strömung
- EX23: Analytische Lösung: Elliptische Gleichung
- EX24: Analytische Lösung: Parabolische Gleichung (Diffusion)
- EX25: Analytische Lösung: Transportgleichung (ADE)
- EX26: Finite-Differenzen-Methode (FDM) explizit
- EX27: Finite-Differenzen-Methode (FDM) implizit
- EX28: Gerinnehydraulik