

Einführung in die Programmierung mit Python

Dr. Bernd Zey

Abteilung Forschungsdatenmanagement

Universitätsbibliothek, TU Dortmund

06.03.2026

- Worum geht es heute hier?
 - Kurz-Einstieg in Python
 - Anwendungsfall
 - Tabellarische Daten laden und auswerten
 - Ergebnisse grafisch darstellen
 - Ergebnisse in Dateien speichern
 - Keine Python-/Programmierkenntnisse erforderlich

Warum Python?

- **Standard** in Data Science
- **Zahlreiche Module** für die statistische Auswertung und Visualisierung (*Open Source*)
 - **Pandas**, NumPy, Matplotlib, Seaborn, ...
- **Einsteigerfreundliche Programmiersprache**
 - Allgemeine Programmiersprache, Plattform-unabhängig
 - Viele (wissenschaftliche) Open Source-Projekte
 - Einfache Syntax
 - Objektorientierung, Prozedural, ...
 - Schnell umfangreiche Programme möglich
 - Einfaches Datei-Handling
 - Viele Module, u.a. für Machine Learning
 - Online-Hilfe



- **Entwicklungsumgebung** – IDE (Integrated Development Environment)

- JupyterLab <https://jupyter.org/>



Entwicklungsumgebungen

- Entwicklungsumgebungen
 - JupyterLab <https://jupyterlab.com/>

The screenshot shows the JupyterLab interface. On the left is a file browser with a search bar and a list of files. The file '2024-03-py...' is selected. On the right is a code editor with a tab titled '2024-03-python-workshop-f'. The code editor shows a notebook with the following content:

```
# Im nächsten Schritt wird das beispielhaft vorgeführt, wenn die Datei im Unterordner "data"
# import os
```

Dateien laden

- csv-Datei (comma separated value-Datei)
- Optional: falls Date im Unterordner: Pfad mit os zusammenbauen

```
[50]: # Dateiname der zu Ladenden Datei
# filename = os.path.join("data", "Titanic-Dataset.csv") # optional: wenn die Datei im Unte
filename = "Titanic-Dataset.csv"
```

```
[51]: # Datei Laden und in Variable "data" speichern
data = pd.read_csv(filename)
```

```
[52]: # Tabelle data anzeigen
# In JupyterLab direkt mit dem Namen möglich
data
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	
0		1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1		2	1	1	Cumings, Mrs. John Bradley (Florence Briggs)	female	38.0	1	0	PC 17599	71.2833	C85	C

- **Entwicklungsumgebung** – IDE (Integrated Development Environment)

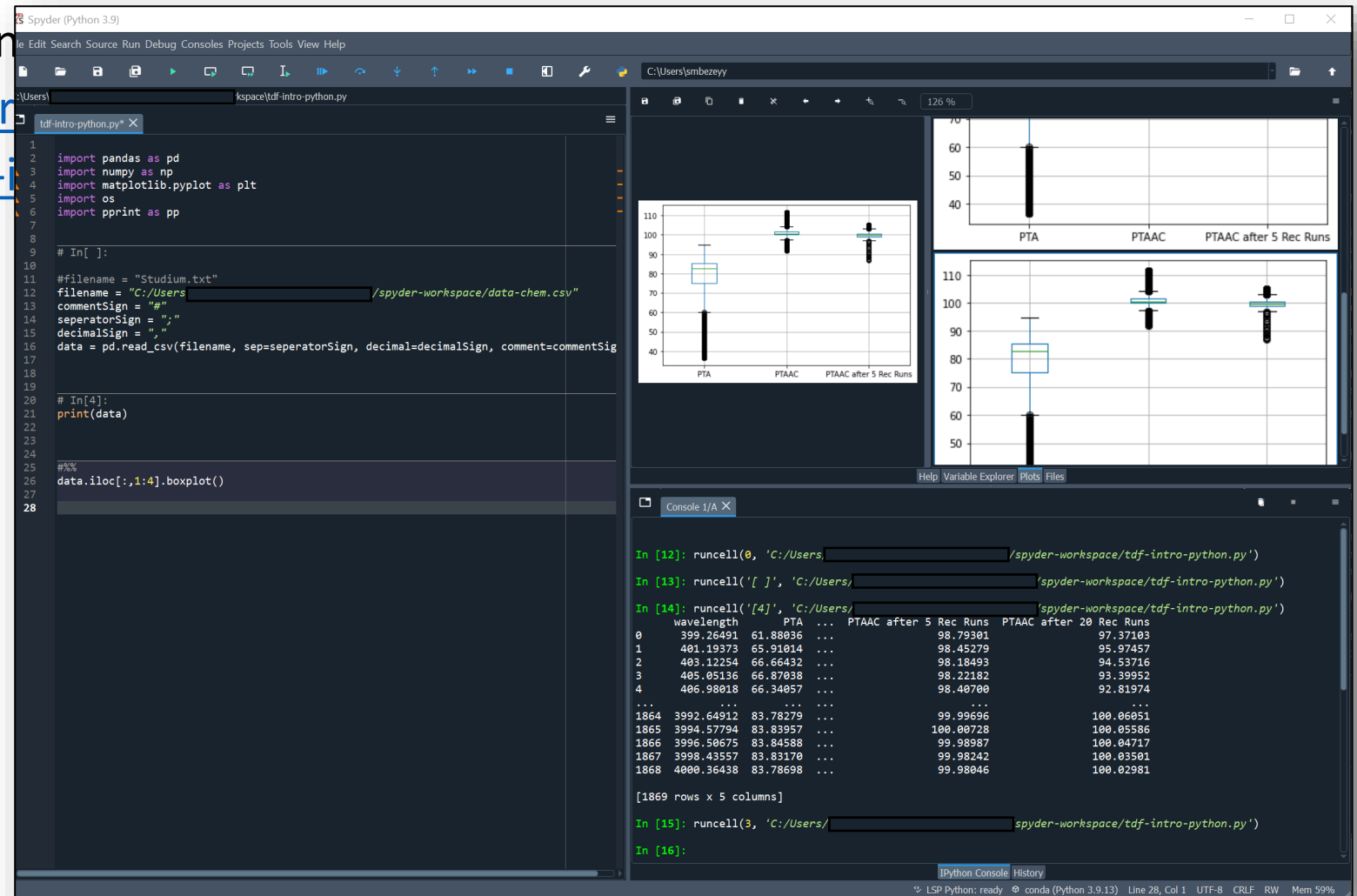
- JupyterLab <https://jupyter.org/>
- Spyder <https://www.spyder-ide.org/>



Entwicklungsumgebungen

- Entwicklungsumgebung – IDE (Integrierte Entwicklungsumgebung)

- JupyterLab <https://jupyter.org/>
- Spyder <https://www.spyder-ide.org/>



- **Entwicklungsumgebung** – IDE (Integrated Development Environment)

- JupyterLab <https://jupyter.org/>
- Spyder <https://www.spyder-ide.org/>
- PyCharm <https://www.jetbrains.com/pycharm/>
- Visual Studio Code <https://code.visualstudio.com/>



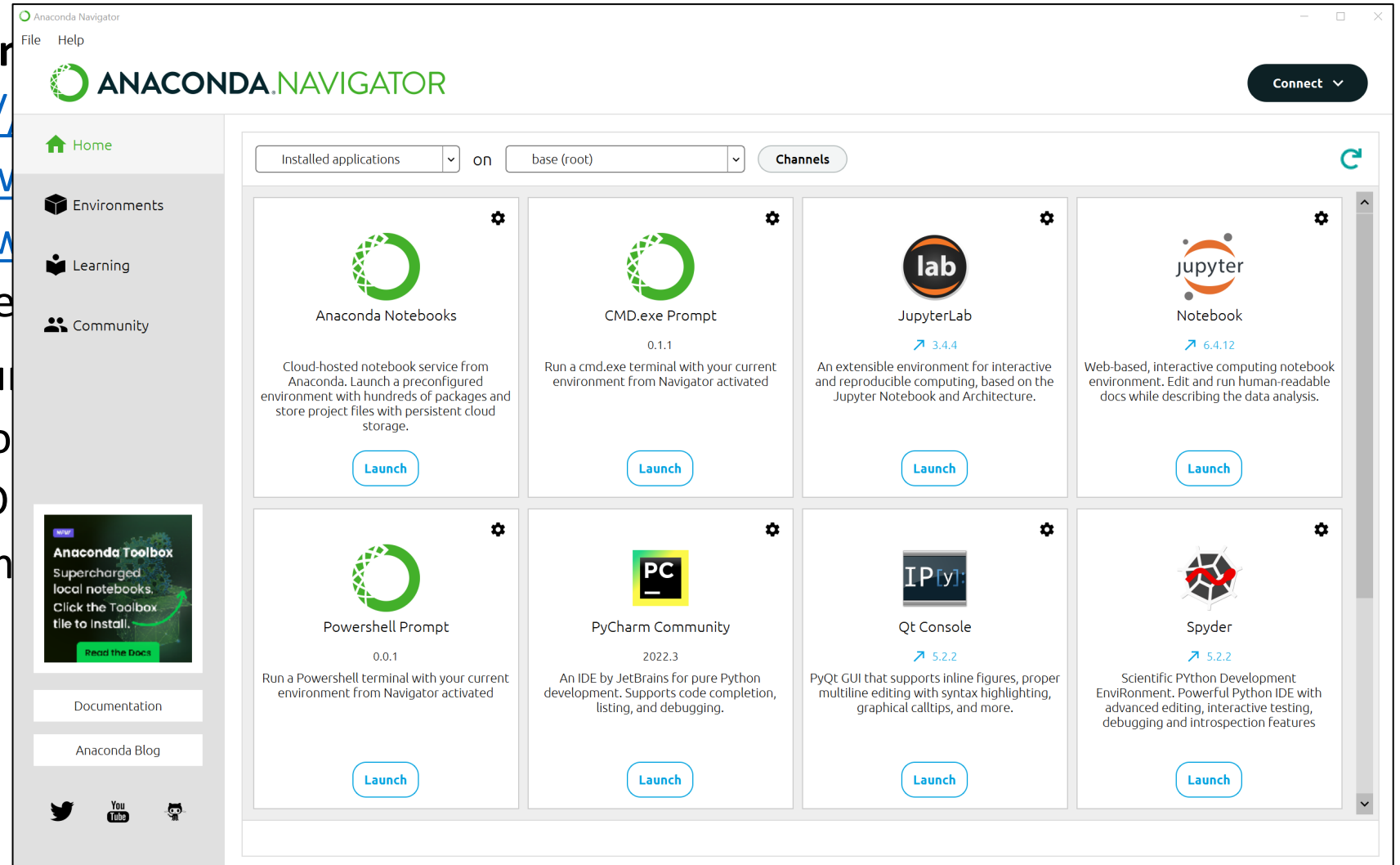
Entwicklungsumgebungen

- **Entwicklungsumgebungen**

- JupyterLab <https://jupyterlab.readthedocs.io/en/latest/>
- Spyder <https://www.spyder-ide.org/>
- PyCharm <https://www.jetbrains.com/pycharm/>
- Visual Studio Code

- **Python:** (Mein Tipp zur Wahl)

- Python-Distributionen
- Liefert mehrere IDEs
- Einfache Environment-Management



- **Entwicklungsumgebung** – IDE (Integrated Development Environment)

- JupyterLab <https://jupyter.org/>
- Spyder <https://www.spyder-ide.org/>
- PyCharm <https://www.jetbrains.com/pycharm/>
- Visual Studio Code <https://code.visualstudio.com/>



- **Python:** (Mein Tipp zum Einstieg) Anaconda ANACONDA.

- Python-Distribution, installiert Python
- Liefert mehrere IDEs (u.a. alle oben genannten), Anaconda Navigator
- Einfache Environment- und Paket-Verwaltung
- [Terms of Service](#):
 - “You can use the Platform for free if:
 - (1) you are an individual that is using the Platform for your own personal, non-commercial purposes;
 - (2) you are using the Platform on behalf of or in association with an Eligible Academic Institution [...]”
 - “You must pay for a ‘Business Plan’ Subscription from Anaconda if you are using the Platform on behalf of a for-profit organization with more than 200 total employees or contractors [...]”

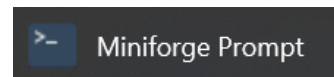
- **Entwicklungsumgebung** – IDE (Integrated Development Environment)

- JupyterLab <https://jupyter.org/>
- Spyder <https://www.spyder-ide.org/>
- PyCharm <https://www.jetbrains.com/pycharm/>
- Visual Studio Code <https://code.visualstudio.com/>



- **Freie Alternative:** z.B. Miniforge

- Installation-Beschreibung: siehe <https://fdm.tu-dortmund.de/informationen/workshop-einfuehrung-in-python/>



```
Miniforge Prompt - conda install pandas
(base) C:\Users\<user>>conda install pandas
Retrieving notices: done
Channels:
 - conda-forge
Platform: win-64
Collecting package metadata (repodata.json): \_

Miniforge Prompt - jupyter lab
(base) C:\Users\<user>>jupyter lab
[I 2025-02-20 12:17:02.039 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2025-02-20 12:17:02.045 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2025-02-20 12:17:02.056 ServerApp] jupyterlab | extension was successfully linked.
[I 2025-02-20 12:17:02.414 ServerApp] notebook_shim | extension was successfully linked.
[I 2025-02-20 12:17:02.464 ServerApp] notebook_shim | extension was successfully loaded.
[I 2025-02-20 12:17:02.468 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2025-02-20 12:17:02.469 ServerApp] jupyter_server_terminals | extension was successfully loaded.
```

- **Entwicklungsumgebung** – IDE (Integrated Development Environment)

- JupyterLab <https://jupyter.org/>
- Spyder <https://www.spyder-ide.org/>
- PyCharm <https://www.jetbrains.com/pycharm/>
- Visual Studio Code <https://code.visualstudio.com/>



- **Freie Alternative:** z.B. Miniforge

- **Online-Lösungen**

- Temporär: <https://jupyter.org/try> → JupyterLab
 - `%pip install openpyxl`
 - Keine sensiblen Daten!
- Mit Google-Account:
 - Google Colab <https://colab.research.google.com/>
 - ggf. Einstellungen anpassen oder anderen Browser wählen (z.B. Chrome)
- JupyterHub.NRW
 - <https://www.jupyterhub.nrw/>
 - Aktuell in *Pilotphase* (Uni MS, H-BRS, RUB, TU Dortmund)

- **JupyterLab, Live Coding**
 - Abschnittsweise Ausführung
 - Markdown-Dokumentation und -Beschreibung
 - Direkte Ausgabe der Diagramme
- **Code und Daten verfügbar auf Homepage**
 - <https://fdm.tu-dortmund.de/informationen/workshop-einfuehrung-in-python/>
 - Code-Lizenz: MIT
 - **Verwendete Daten**
 - <https://www.kaggle.com/datasets/yasserh/titanic-dataset>
 - Datensatz der Titanic
 - Lizenz: CC0

- *Take-Away Message:*
Python *einfache* Programmiersprache
→ Automatisierung
- Mächtige Module (heute: Pandas) für Data Science
- Online gute Hilfe (Diskussionen, Code-Schnipsel)
 - Websuche → „python pandas *thema*“
 - z.B. <https://stackoverflow.com>
 - KI-Tools generieren (guten) Code
(i.d.R. noch anzupassen)

