



TRAINING: Python für die Datenanalyse in den Sozialwissenschaften

Python Grundlagen

SPEAKER: Matthias Täschner



GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung



SACHSEN

Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.



Come2Data
Kompetenzzentrum für
interdisziplinäre Datenwissenschaften

Training: Python für die Datenanalyse in den Sozialwissenschaften – Python Grundlagen
Speaker: Matthias Täschner, Universität Leipzig, ScaDS.AI

Folie 1



Begriffe der Programmierung

Was ist Programmierung?

- Verwendung einer Programmiersprache zur Umsetzung von Softwareanforderungen in ein Computerprogramm
- Computerprogramm wird zur Ausführung in Maschinencode umgewandelt (kompiliert oder interpretiert)

Was ist eine Programmiersprache?

- Werkzeug zur Formulierung von Algorithmen und Datenstrukturen
- Formale Sprache mit Syntax und Semantik

Algorithmus

- Anweisungsabfolge, um eine Aufgabe zu lösen
- Anweisung besteht aus erlaubten Mustern

Datenstruktur

- Objekt, um Daten im Speicher zu halten und zu organisieren

Syntax

- Formale Regeln zur Verwendung von Anweisungen
- "Grammatik" einer Programmiersprache

Semantik

- Sinnhafte Bedeutung der Anweisungen

Was ist Python?

Universelle High-Level-Programmiersprache, oft auch als Skriptsprache verwendet

- Veröffentlicht 1994, aktuelle stabile Version ist 3.13
- Ziele: Einfachheit, Erweiterbarkeit
 - Wenige reservierte Schlüsselwörter, reduzierte Syntax
 - Umfangreiche Standardbibliothek, z.B. Datei- / Textverarbeitung, Mathe, ...
 - Einfache Integration von zusätzlichen Paketen / Bibliotheken
- Open Source, portabel auf den meisten Plattformen / OS
- Weit verbreitet in den Bereichen Datenwissenschaft, -analyse, KI/ML
- Einfache Verwaltung und Nutzung von zusätzlichen Paketen und Erweiterungen
 - Eingebauter Paketmanager „pip“ mit [Python Package Index PyPI](https://pypi.org/)
 - Python-Distributionen, die Python bereitstellen + alternativer Paketmanager (z.B. „conda“) + virtuelle Umgebungen + vorinstallierte Pakete - z. B. [Miniconda](https://miniconda.org/), [Anaconda](https://anaconda.org/)



(TM) Trademark of the PSF
<https://www.python.org>

Was ist Python?

Begriffe und Konzepte

Variable

Container für das Speichern von zugewiesenen Daten, mit Namen für die Referenzierung

Objekt

Komplexe Struktur zum Bündeln von Daten und Methoden, um diese zu bearbeiten

Methode

An ein Objekt gebundener Code-Block, aufrufbar über ". "
(„Methode ist meine“)

Alles in Python ist ein Objekt

```
1 x = 5
2
3 class Human:
4     def __init__(self, name):
5         self.name = name
6
7     def say_hello(self):
8         return "Hello, I'm " + self.name
9
10 myself = Human("Matthias")
11
12 print(myself.say_hello())
```

Zuweisung

Variablen-Name

Wert mit spezifischem Typ

Definition eines Objekts

Instanziierung eines Objektes und Zuweisung zu Variablen

Aufruf der Methode eines Objekts

Was ist Python?

Begriffe und Konzepte

Funktion

Unabhängiger wiederverwendbarer Code-Block für spezifische Aufgabe („Funktion ist frei“)

Modul

Datei mit Python-Code, welcher in anderen Python-Code importiert und genutzt werden kann

Kommentar

Code-Zeilen, die von Python nicht interpretiert werden, dienen der Dokumentation, beginnt mit #

Modul für zusätzliche Funktionalität

Kommentar, wird nicht interpretiert

```
1 import math
2
3 # My custom function
4 def greet(name):
5     return "Hello, " + name
6
7 print(greet("Matthias"))
8 print("Pi = " + str(math.pi))
```

Definition einer eigenen Funktion

Aufruf der Funktion

Aufruf einer eingebauten Funktion

Aufruf einer eingebauten Funktion

Aufruf einer Funktion des importierten Moduls

Was ist Python?

Ausführung von Python Code

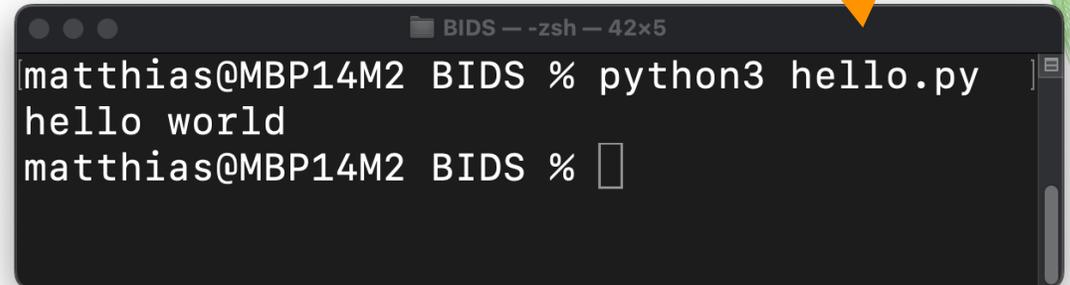
Ausführung einer Python Datei

- Code in Datei mit Endung „.py“ speichern
- Datei mit installiertem Python ausführen

Interaktive Ausführung im Terminal

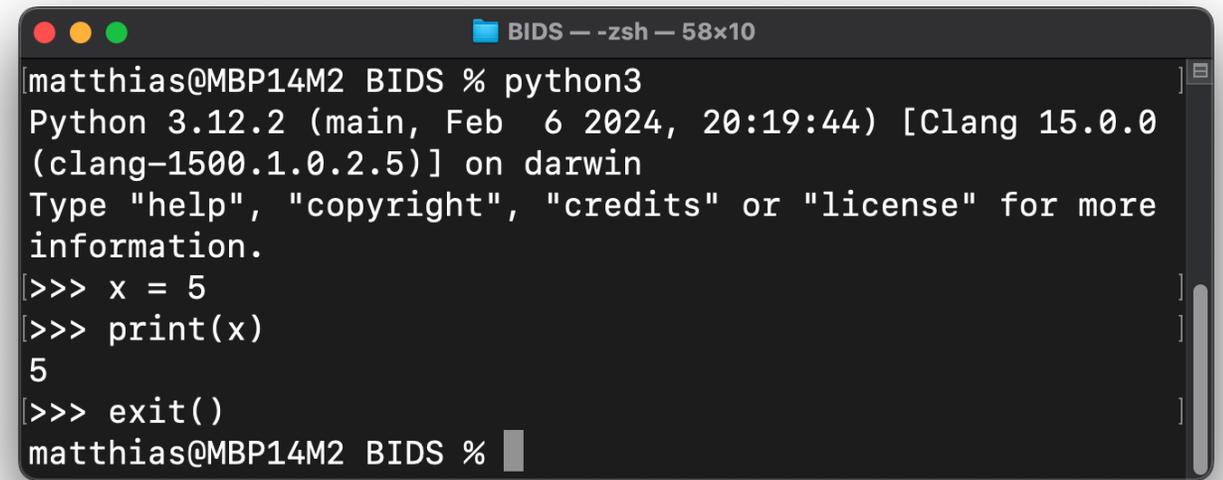
- Start einer interaktiven Python Session
- Zeilenweise Eingabe und Ausführung von Anweisungen

```
hello.py  
1 print("hello world")
```



A terminal window titled "BIDS - -zsh - 42x5" showing the execution of a Python script. The prompt is "matthias@MBP14M2 BIDS %". The command "python3 hello.py" is entered, and the output "hello world" is displayed. The prompt returns to "matthias@MBP14M2 BIDS %". An orange arrow points from the code above to the terminal window.

```
matthias@MBP14M2 BIDS % python3 hello.py  
hello world  
matthias@MBP14M2 BIDS %
```



A terminal window titled "BIDS - -zsh - 58x10" showing an interactive Python session. The prompt is "matthias@MBP14M2 BIDS %". The command "python3" is entered, and the Python 3.12.2 startup banner is displayed. The user enters "x = 5", "print(x)", and "exit()" in sequence. The output "5" is shown for the print statement. The prompt returns to "matthias@MBP14M2 BIDS %".

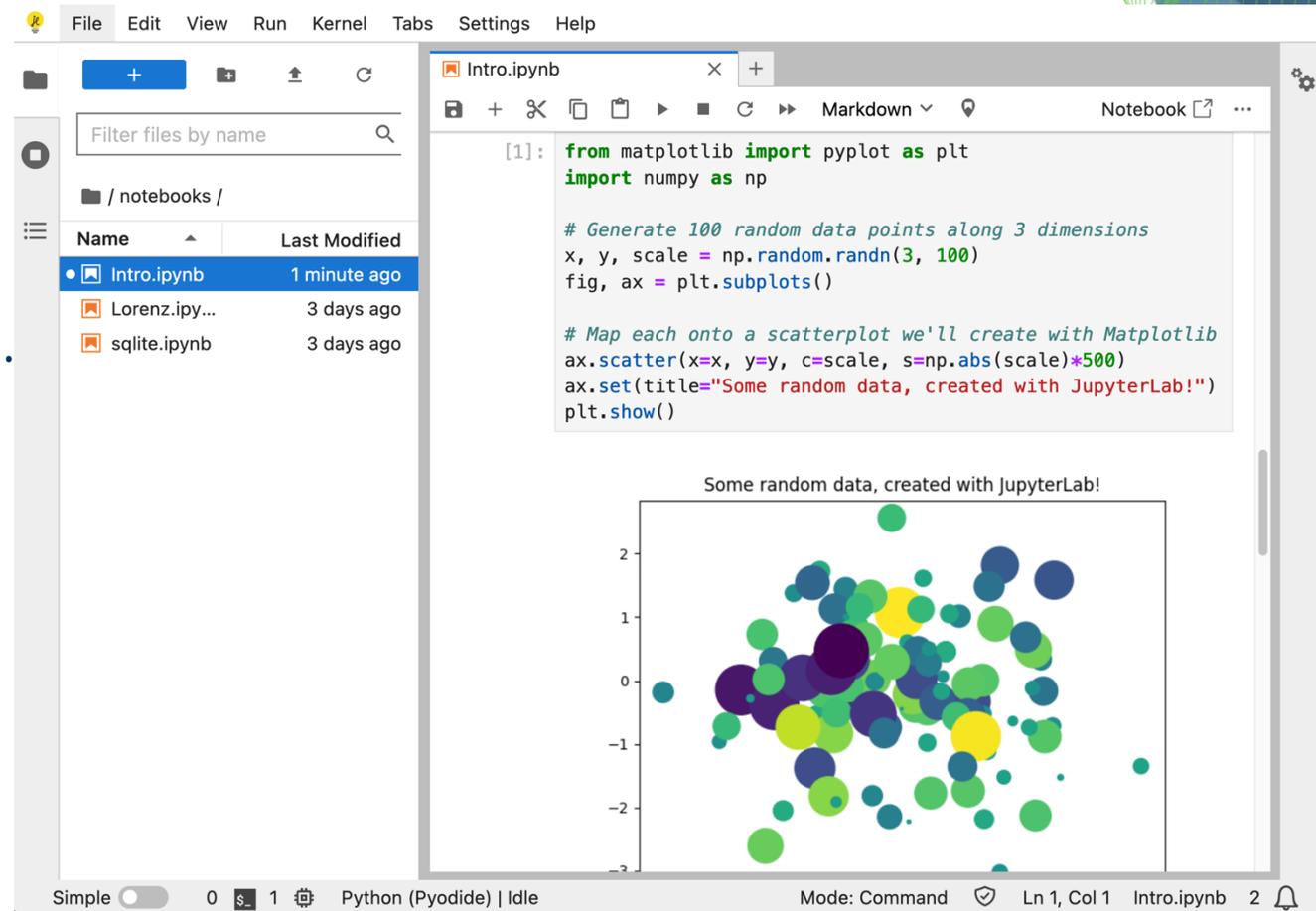
```
matthias@MBP14M2 BIDS % python3  
Python 3.12.2 (main, Feb 6 2024, 20:19:44) [Clang 15.0.0  
(clang-1500.1.0.2.5)] on darwin  
Type "help", "copyright", "credits" or "license" for more  
information.  
>>> x = 5  
>>> print(x)  
5  
>>> exit()  
matthias@MBP14M2 BIDS %
```

Was ist Python?

Ausführung von Python Code

Interaktive Ausführung in Jupyter Notebook

- Web-basiertes Interface mit Zellen für
 - Ausführbaren Python Code
 - Text für Dokumentation
 - Ausgabe von Visualisierung, Diagramme, ...
- Jupyter Lab als Erweiterung mit
 - Jupyter Notebook
 - Datei-Browser
 - Zugriff auf Terminal
 - Plugins für weitere Funktionalitäten



The screenshot displays the Jupyter Notebook interface. On the left, a file browser shows a directory named '/notebooks/' containing three files: 'Intro.ipynb' (modified 1 minute ago), 'Lorenz.ipynb' (modified 3 days ago), and 'sqlite.ipynb' (modified 3 days ago). The main area shows a code cell with the following Python code:

```
[1]: from matplotlib import pyplot as plt
import numpy as np

# Generate 100 random data points along 3 dimensions
x, y, scale = np.random.randn(3, 100)
fig, ax = plt.subplots()

# Map each onto a scatterplot we'll create with Matplotlib
ax.scatter(x=x, y=y, c=scale, s=np.abs(scale)*500)
ax.set(title="Some random data, created with JupyterLab!")
plt.show()
```

Below the code, a scatter plot is displayed with the title "Some random data, created with JupyterLab!". The plot shows 100 data points in a 2D space, where the color and size of each point correspond to a third dimension (scale). The points are scattered across the plot area, with colors ranging from dark purple to bright yellow and green. The axes range from approximately -3 to 2 on the y-axis and -2 to 2 on the x-axis.

<https://jupyter.org/try-jupyter/lab/index.html>