

Computergrundlagen 2025

Blatt 7: Python und Zotero

- Abgabetermin für die Lösungen: **07.12.2025, 20 Uhr/ für Montagsgruppe: 05.12.2025, 12 Uhr**
- Bei Fragen wendet euch bitte an eure/n Tutor/in:
 - Mo 11:30: Stephan Haag: st170833@stud.uni-stuttgart.de
 - Di 09:45: Julian Hoßbach: julian.hossbach@icp.uni-stuttgart.de
 - Mi 14:00: Julian Peters: julian.peters@icp.uni-stuttgart.de
 - Do 09:45: Rebecca Stephan: rebecca.stephan@icp.uni-stuttgart.de
 - Fr 09:45: Jonas Höpker: st182335@stud.uni-stuttgart.de
- Die Übungsaufgaben sollen in der Regel in **Zweiergruppen** bearbeitet werden. Nur in **begründeten Ausnahmefällen** sind Dreiergruppen möglich.
- Die Abgabe der Übungsblätter erfolgt über Ilias.
- Mit Abgabe der Lösungen erklärt Ihr, dass Ihr die Lösung euren Mitstudierenden im Rahmen der Übungsbesprechung vorstellen könnt. Um dies zu überprüfen, muss mindestens zweimal von jedem Teilnehmenden vorgetragen werden. Wenn Ihr das nicht könnt, werden euch die Punkte für die entsprechenden Aufgaben wieder abgezogen.
- **Befehle, die nicht in der Vorlesung besprochen wurden, müssen gegebenenfalls recherchiert werden.**
- **Alle erstellen Skripte (.py and .ipynb) sowie ein mit markown oder Latex erstellter Report (.pdf) sind Teil der Abgabe**

Tipp: Für Abgaben in Python kann der Editor <https://jupyter.org/> hilfreich sein. Hier könnt ihr euren Code *interactive* ausführen sowie mit Markdown Zellen ausführlich kommentieren. Die *.ipynb* Dateien können ebenfalls als Lösungen abgegeben werden und sind leicht zu validieren.

Literaturmanagement mit Zotero (3 Punkte)

Ziel dieser Aufgabe ist es, die grundlegenden Funktionen zur Literaturverwaltung mit Zotero kennenzulernen, insbesondere die automatisierte Erfassung von Quellen, die manuelle Eingabe und den Export in LaTeX-Projekte.

Installation:

Stelle sicher, dass die neueste Version von Zotero (Desktop-Anwendung) und der zugehörige Zotero Connector (Browser-Plugin) installiert sind. * Download: <https://www.zotero.org/download/>

1. Automatische Erfassung per Zotero Connector Der Zotero Connector ermöglicht dir das Speichern von bibliografischen Daten direkt aus dem Webbrowser in deine Zotero-Bibliothek.

- Suche mithilfe einer wissenschaftlichen Suchmaschine (z.B. Google Scholar) nach einem aktuellen Zeitschriftenartikel zum Thema "Quantum Machine Learning".
- Speichere diesen Artikel über das Symbol des Zotero Connectors in deiner Zotero-Bibliothek.
- Nachweis: Füge einen Screenshot bei, der zeigt, wie der Artikel korrekt mit allen Metadaten in deiner Zotero-Bibliothek erscheint.

2. Manuelle Erfassung Füge der Zotero-Bibliothek manuell einen Bucheintrag hinzu:

- Klicke in Zotero auf das Plus-Symbol (+), wähle “Buch” und trage folgende Pflichtfelder für das Buch “Experimentalphysik 1” von Wolfgang Demtröder ein:
 - **Author**
 - **Title**
 - **Date** (Erscheinungsjahr)
 - **Publisher** (Verlag)
- Nachweis: Füge einen Screenshot des ausgefüllten Eingabefensters (rechter Reiter in Zotero) bei.

3. Export und LaTeX-Integration Um die Quellen in deinem wissenschaftlichen Bericht zu verwenden, musst du sie im passenden Format exportieren.

- Export: Wähle die beiden erfassten Quellen (Artikel und Buch) in deiner Zotero-Bibliothek aus.
 - Erstelle eine Exportdatei im Format BibTeX (**.bib**).
- Integration: Binde diese **.bib**-Datei in deinen LaTeX-Bericht für dieses Übungsblatt ein.
- Zitieren & Literaturverzeichnis:
 - Zitiere beide Quellen mindestens einmal im Haupttext deines LaTeX-Dokuments (mit `\cite{...}`).
 - Erstelle am Ende des Dokuments ein vollständiges Literaturverzeichnis.

List Comprehension (5 Punkte)

In dieser Aufgabe soll das bereitgestellte, jedoch unvollständige Skript `list_comprehension.py` vervollständigt werden. Die Funktion `generate_random_point_in_square` ist bereits implementiert und generiert zufällige Punkte in einem Quadrat gegebener Seitenlänge.

- Implementiere die Funktion `distance`, welche den Euklidischen Abstand zwischen zwei Punkten berechnet.
- Implementiere die Funktion `point_in_circle`, welche für einen gegebenen Punkt überprüft, ob dieser innerhalb eines Kreises mit gegebenem Radius um einen gegebenen Mittelpunkt liegt.
- Nutze List Comprehension in Verbindung mit `generate_random_point_in_square` um eine Liste von 1000 Punkten innerhalb eines Quadrates der Seitenlänge 2.0 zu erzeugen.
- Nutze List Comprehension um eine Liste zu erzeugen, die nur diejenigen Punkte der vorherigen Liste enthält, die in einem Kreis mit Radius 1.0 um den Ursprung liegen.
- Berechne das Verhältnis der Anzahl der Punkte in der gefilterten Liste zur Gesamtanzahl erzeugter Punkte. Gegen welche Konstante konvergiert diese Verhältnis, wenn die Anzahl erzeugter Punkte gegen Unendlich geht? Begründe deine Antwort.
- Nutze List Comprehension um separate Listen für die x - und y -Werte der beiden obigen Listen zu erstellen und erzeuge anschließend einen Scatter Plot.

Dateien lesen und Daten plotten (2 Punkte)

Die bereitgestellte Datei `messdaten.txt` enthält Koordinatenpaare (x - und y -Werte) von einer (fiktionalen) Messung. Die Datei ist nicht perfekt: Sie beinhaltet leere Zeilen oder Kommentare.

Ziel der Aufgabe ist es, ein Python-Programm zu schreiben, das diese Datei liest, die Daten bereinigt und alle gültigen (x,y)-Tupel extrahiert, um sie anschließend grafisch darzustellen. Verwende dafür das bereitgestellte Skript `read_data.py` und ergänze es an den Stellen, die mit `# DEIN CODE HIER` markiert sind.

1. Zerlegen der Zeile:
 - Teile die `bereinigte_zeile` in zwei Einzelteile (die potenziellen x- und y-Werte) auf.
2. Konvertierung und Speicherung:
 - Wandle die extrahierten Teile in Gleitkommazahlen (`float`) um.
 - Speichere die konvertierten x- und y-Werte in den jeweiligen Listen `x_werte` und `y_werte`.
3. Visualisierung:
 - Erstelle einen Plot (`plt.scatter()`) der gesammelten `x_werte` gegen die `y_werte`.