



**Übungen zu Scientific Computing mit Python
Sommersemester 2024**

Übungsblatt 3

Ausgabe 10.5., Übungen KW 20+21, Abgabe bis 3.6.

Git, Datenstrukturen und Rechnen mit Python

1. Aufgabe: Git nutzen

- (a) Git - Lokal
 - i. Lege ein Verzeichnis an und damit ein Git-Repository.
 - ii. Lege eine Datei an und füge sie zum Repo hinzu.
 - iii. Ändere die Datei und füge die Änderungen ebenfalls hinzu.
 - iv. Lege einen neuen Branch an und ändere dort die Datei nochmals.
 - v. Füge die Änderung in den Branch hinzu.
 - vi. Merge den Branch mit dem master-Branch.
 - vii. Zeige die Git Historie.
- (b) Git - Remote
 - i. Lege ein Verzeichnis an und damit ein Git-Repository.
 - ii. Lege ein Remote-Repo auf gitlab.inf.uni-konstanz.de oder GitHub an.
 - iii. Klone das Repo und lege darin eine Datei an.
 - iv. Pushe die Änderungen auf das Remote-Repo.
 - v. Zeige die Webansicht des Remote-Repos.
- (c) Software-Repos
 - i. Klone das Repo <https://invent.kde.org/education/labplot>.
 - ii. Wieviele Commits gibt es? Wann war der erste Commit? Wie viele Branches gibt es aktuell? Was sind Tags?
 - iii. Versuche die Software zu kompilieren (siehe INSTALL).

2. Aufgabe: Datenstrukturen in Python

- (a) Lege einen Enum-Datentyp für die Jahreszeiten an und verwende ihn, um eine Jahreszeit vom Benutzer einzulesen und auszugeben.
- (b) Lege eine Datenklasse (dataclass) für ein Teilchen mit Masse, Ort und Geschwindigkeit an. Verwende die Klasse um die Bewegung des Teilchen bei konstanter Geschwindigkeit für 100 Zeitschritte zu berechnen.
- (c) Verwende `deque` um eine Warteschlange für Hausaufgaben und einen Stapel für Bücher zu implementieren.
- (d) Lege eine Liste von Messwerten an und sortiere diese numerisch.

3. Aufgabe: Rechnen mit Python

(a) NumPy Arrays

- i. Erzeuge ein Numpy-Array mit den Werten von 1 - 100.
- ii. Halbiere alle Werte.
- iii. Ersetze alle geraden Werte mit 0.
- iv. Ersetze alle Werte, die Null sind mit 1.
- v. Normiere die Werte (Skaliere mit Faktor, so dass der Maximalwert 1 ist).
- vi. Berechne die Differenz aufeinanderfolgender Werte.

(b) SymPy

- i. Definiere den Ausdruck $f = xe^{-x}$ und werte ihn für $x = 0; 0, 1; 0, 2; 0, 4$ aus.
- ii. Faktorisiere das Polynom $p(x) = x^4 - 2x^3 + x^2 - 2x$ und finde die Nullstellen.
- iii. Bestimme die Potenzreihe von $\sin(x)$ und $\tan(x)$ bis zur 5. Ordnung.
- iv. Integriere die Funktion $f(x) = \sin(x)e^{-x}$ und differenziere sie zur Probe.
- v. Berechne (mit Einheiten): $F = ma$ für $m = 1$ kg und $a = 10$ m/s².

4. Aufgabe: Testen und Multiprocessing

(a) Kommandozeile und Testen

- i. Schreibe eine Funktion, die den Binomialkoeffizienten $\binom{n}{k}$ berechnet.
- ii. Schreibe ein Programm, das n und k von der Kommandozeile übernimmt und rufe damit die Funktion auf.
- iii. Schreibe mit `pytest` die fünf Tests $\binom{0}{0} = 1$, $\binom{5}{1} = 5$, $\binom{5}{5} = 1$, $\binom{-1}{1} = 0$ und $\binom{3}{2} = 0$ für die Funktion und verbessere, wenn nötig, die Funktion.

(b) Multiprocessing

- i. Teste das Multiprocessing-Beispiel aus der Vorlesung für verschiedene Anzahl an Threads.
- ii. Wieviele Cores hat der genutzte Rechner und welche Anzahl an Threads ist optimal (am schnellsten)?