



Übungen zu Scientific Computing mit Python
Sommersemester 2025

Übungsblatt 2

Ausgabe 25.4., Übungen KW 18+19, Abgabe bis 9.5.

Linux, Netzwerk und Versionsverwaltung

1. Aufgabe: Linux-Terminal und Interaktion in Python

- (a) Mit `>`, `>>` und `|` kann man die Ausgabe von Programmen in Dateien umlenken bzw. verknüpfen. Erkläre was die folgenden Zeilen machen und wo deren Unterschiede liegen
- ```
$ ls > text.txt
$ ls >> text.txt
$ ls | sort > text.txt
```
- (b) Erkläre (mindestens) fünf Informationen, die das Programm `top` in der Kopfzeile anzeigt?
- (c) Was besagt der Nice-Wert eines Prozesses? Mit welchem Kommando kann man ihn ändern bzw. setzen?
- (d) Die Python-Module `os`, `sys` und `platform` erlauben die Interaktion mit dem Betriebssystem.
- Gebe `os.name`, `sys.version`, `platform.system()` und `platform.release()` aus und erkläre die Ausgaben.
  - Finde die Funktion im Modul `os`, mit der man das aktuelle Verzeichnis, die Umgebungsvariable `HOME` und den Inhalt des aktuellen Verzeichnisses ausgeben kann. Probiere es aus.
  - Wie lauten die Funktionen um ein Verzeichnis anzulegen, ein Verzeichnis zu löschen, das Verzeichnis zu wechseln, eine Datei umzubenennen? Lege zum Testen ein Verzeichnis an und wechsele in dieses.

**2. Aufgabe: Netzwerk**

- (a) Das Netzwerk im PhyMa PC-Pool lautet 134.34.140.128/25. Bestimme die Netzmaske, Gateway, Broadcast, Hostbereich, IP eines Rechners und dessen Hostnamen.
- (b) Aus wievielen Bits besteht eine MAC-Adresse? Wieviele verschiedene MAC-Adressen gibt es damit?
- (c) Was bedeuten die Anwendungsprotokolle HTTP, HTTPS, FTP, SSH, IMAP und wofür werden diese verwendet?

- (d) Wofür ist die Option `-X` von `ssh` gut? Probiere es aus.
- (e) Mit dem Python-Modul `socket` lassen sich Netzwerkinformationen finden und einfache Netzwerk-Programmierung. Mit `socket.gethostname()` erhält man den Hostnamen. Finde mit `socket.gethostbyname()` die eigenen IP-Adresse und die IP-Adresse von `www.uni-konstanz.de`.

### 3. 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).