

# BDL - Grundlagen - 02: Linux

---

## Gestern

- Kennenlernen
- Warum Linux
- Softwareentwicklung im Allgemeinen
- Kursübersicht

## HEUTE

- Das Linux-Dateisystem
- Linux Terminal
- Terminalbefehle
- kleine Übung
- Bonus: "Terminalprogramm"

## Das Linux-Dateisystem

- Bevor wir anfangen, eine Erinnerung
  - **Du musst nicht alles perfekt behalten**
  - Du hast das Internet und viel Zeit zum Üben!
- Du kennst vielleicht die Funktionsweise von Dateien in Windows
- Dort hast du deine C:\ und möglicherweise D:\ usw. Laufwerke, die das Dateisystem enthalten
- In Linux gibt es keine solchen Laufwerke
- Dateien und Ordner (in Linux meist Verzeichnisse genannt) existieren in einer Hierarchie
- Alles beginnt vom Root-Verzeichnis, /
- Jede Datei und jeder Ordner hat einen einzigartigen Pfad
- `/tmp/notes.txt`
  - Im Root gibt es ein Verzeichnis namens tmp
  - In tmp gibt es eine Datei namens notes.txt

`/home/dci/cv.pdf` - Im Root gibt es ein Verzeichnis namens home - In home gibt es ein Verzeichnis namens dci - In dci gibt es eine Datei namens `cv.pdf`

- Vier bemerkenswerte Verzeichnisse, [hier eine ausführlichere Übersicht](#)
  - `/` das Root-Verzeichnis
  - `/etc` Konfigurationsdateien
  - `/var` Log-Dateien und andere Dateien variabler Größe
  - `/home` Home-Verzeichnisse

- Normalerweise hat jeder Benutzer auf einem Linux-Computer sein eigenes Home-Verzeichnis
  - `/home/sarah`
  - `/home/ferdinand`
  - `/home/inkeri`
  - Der Verzeichnisname ihres Home-Verzeichnisses (`sarah`) entspricht ihrem Benutzernamen
  - Dies ist "nach Konvention"
    - Das bedeutet, es ist keine feste Regel, aber es wird praktisch überall verwendet
- Das Home-Verzeichnis des aktuellen Benutzers hat auch eine kleine Abkürzung
  - Der Pfad `~` bezieht sich auf das Home-Verzeichnis des aktuellen Benutzers
  - Wenn man als `student` eingeloggt ist, ist das Home-Verzeichnis oder `~ /home/student`

## Terminal

- Zentrale Schnittstelle des Betriebssystems: Die Shell ist das wichtigste Werkzeug, um direkt mit dem Linux-Betriebssystem zu interagieren und ermöglicht umfassenden Zugriff auf Systemressourcen und Konfigurationen.
- Effiziente Bedienung und Steuerung des Systems: Mit der Shell lassen sich alle Aspekte von Linux schnell und präzise kontrollieren, wodurch sie besonders für Automatisierung und Entwicklung essenziell ist.
- Wenn du arbeitest, empfehle ich **DRINGEND**, immer im Projekte-Ordner zu arbeiten
- Und **IMMER** Backups von allem Wichtigen zu machen

## Terminalbefehle

- siehe [Übersicht](#)

## Wichtige Shortcuts

- `Strg+ Shift+c` - kopieren
- `Strg+ Shift+v` - einfügen
- `Strg+ a` - zum Anfang der Zeile springen
- `Strg+ e` - zum Ende der Zeile springen
- `Pfeil hoch/runter` - durch die letzten Befehle scrollen
- `Tabulator` - Autovervollständigung

## Zusammenfassung Terminal

- Absolute und relative Pfade unterscheiden `./Schreibtisch/projekte/` - **relativer Pfad**, geht vom aktuellen Ort aus, in den Ordner Schreibtisch von dort in projekte  
`./home/student/Schreibtisch/projekte/` - **absoluter Pfad**, startet im root Verzeichnis, geht ebenfalls in projekte
- nutzt die Autovervollständigung mit Tabulator-Taste
- ansonsten schaut nochmal in die Befehlsübersicht, die wichtigsten sind:
  - `cd`, `mkdir`, `cp`, `mv`, `rm`, `cat`, `(nano)`

## Warum das Terminal

- Umfangreiche Tool-Unterstützung und nahtlose Linux-Integration
  - Zugriff auf zahlreiche leistungsfähige Tools, die perfekt in die Linux-Welt eingebettet sind.
- Volle Kontrolle und Anpassungsmöglichkeiten
  - Tiefgehende Konfiguration des Systems und Verwaltung von Berechtigungen direkt über das Terminal.
- Erweiterbarkeit und effiziente Softwareverwaltung
  - Einfache Installation und Verwaltung von Software durch den leistungsstarken Paketmanager apt.
- Nahtlose Unterstützung für Remote-Entwicklung
  - Einfache Verbindung zu entfernten Servern und Verwaltung über SSH, ideal für die Webentwicklung.
- Mächtige Funktionalität durch Piping und Scripting

## Exkurs kleines Bash Script erstellen

1. Skriptdatei erstellen (`nano createWebsite.sh`).
2. Skript schreiben

```
#!/bin/bash
ordner="$1"
mkdir "$ordner"
mkdir "$ordner/bilder"
touch "$ordner/index.html"
touch "$ordner/style.css"
code "$ordner"
```

3. Skript ausführbar machen (`chmod +x ./createWebsite.sh`).
4. Programm von überall ausführbar machen
  - kopieren mit `sudo cp ./createWebsite.sh /usr/local/bin/createWebsite`, so dass es über `createWebsite name` ausführbar ist