

1. Das Setup

1.1 Die Shell

= Terminal, Eingabeaufforderung

- Beispiele: bash, zsh
- direkte Anweisungen an das Betriebssystem
- mehr Möglichkeiten, mehr Kontrolle
- Programme ausführen
- schneller als Mausbedienung
- grundlegende Terminal-Befehle:

Befehl	Beschreibung
ls	list - Inhalt eines Ordners anzeigen
cd	change directory - Ordner wechseln
pwd	print working directory
mkdir	make directory - Ordner erstellen
cp	copy - Dateien oder Ordner kopieren
mv	move - verschieben oder umbenennen
rm	remove - löschen von Dateien oder Ordnern
nano	kleiner Editor
man	manual - Hilfe anzeigen
touch	Datei anlegen oder Timestamp aktualisieren

- Programme:
 - top: Übersicht über Prozesse
 - vim: Texteditor
 - python3: Python-Interpreter
- Beispiel:

```
1 cd ~/workspace
```

/Users/martin/Workspace

Zeige Inhalte des aktuellen Ordners mit ls an.

```
1 ls
```

```
Java_Projects/      Lecture_Projects/  text.txt
Jupyter_Notebooks/  SQL_Projects/      vscode setup/
```

Erweiterte Anzeige mit Rechten, Besitzer, Gruppe, Größe, Zeitstempel

```
1 ls -al
```

```
total 24
drwx-----@ 11 martin  staff    352 Aug 12 10:35 ./
drwxr-x---+ 72 martin  staff   2304 Aug 12 10:35 ../
-rw-r--r--@  1 martin  staff  10244 Aug  7 17:29 .DS_Store
drwxr-xr-x   2 martin  staff    64 Mar 25  2024 .ipynb_checkpoints/
drwxr-xr-x   4 martin  staff   128 Apr 21  2024 .vscode/
drwx-----@  8 martin  staff   256 Aug  7 16:45 Java_Projects/
drwxr-xr-x@ 10 martin  staff   320 Jan 30  2025 Jupyter_Notebooks/
drwxr-xr-x   7 martin  staff   224 Jul 23 09:01 Lecture_Projects/
drwxr-xr-x   9 martin  staff   288 Jan 26  2025 SQL_Projects/
-rw-r--r--@  1 martin  staff     0 Aug 12 10:35 text.txt
drwxr-xr-x@  3 martin  staff    96 Feb 14 12:46 vscode setup/
```

Name des aktuellen Ornders

```
1 pwd
```

```
'/Users/martin/Workspace'
```

Neue Verzeichnis mit Namen "mydata" erstellen.

```
1 mkdir mydata
```

Überprüfen, ob der neue Ornder auch angelegt worden ist.

```
1 ls -al
```

```
total 24
drwx-----@ 12 martin  staff    384 Aug 12 10:37 ./
drwxr-x---+ 72 martin  staff   2304 Aug 12 10:35 ../
-rw-r--r--@  1 martin  staff  10244 Aug  7 17:29 .DS_Store
drwxr-xr-x   2 martin  staff    64 Mar 25  2024 .ipynb_checkpoints/
drwxr-xr-x   4 martin  staff   128 Apr 21  2024 .vscode/
drwx-----@  8 martin  staff   256 Aug  7 16:45 Java_Projects/
drwxr-xr-x@ 10 martin  staff   320 Jan 30  2025 Jupyter_Notebooks/
drwxr-xr-x   7 martin  staff   224 Jul 23 09:01 Lecture_Projects/
drwxr-xr-x@  2 martin  staff    64 Aug 12 10:37 mydata/
drwxr-xr-x   9 martin  staff   288 Jan 26  2025 SQL_Projects/
-rw-r--r--@  1 martin  staff     0 Aug 12 10:35 text.txt
drwxr-xr-x@  3 martin  staff    96 Feb 14 12:46 vscode setup/
```

In den neue Ornder "mydata" wechseln.

```
1 cd mydata
```

/Users/martin/Workspace/mydata

Inhalt des Ordners "mydata" anzeigen. Er ist leer.

```
1 ls
```

Neue Datei namens text.txt. erstellen.

```
1 !touch text.txt
```

Überprüfen, ob die neue Datei auch vorhanden ist.

```
1 ls
```

text.txt

Aus dem Ordner "mydata" in den übergeordneten Ordner mit einer relativen (zum aktuellen Ordner) Pfad wechseln. Der übergeordnete Ordner wird mit .. adressiert.

```
1 cd ..
```

/Users/martin/Workspace

Weiteres Beispiel eines relativen (relativ zum aktuellen Ordner) Pfades.

```
1 cd mydata/../../
```

/Users/martin

Aus dem aktuellen Ordner in den Unter- Unterordner "mydata" wechseln.

```
1 cd Workspace/mydata/
```

/Users/martin/Workspace/mydata

Der aktuelle Ordner ist nun "mydata". In ihm befindet sich die Datei "text.txt". Diese soll mit der Anweisung cp in den übergeordneten Ordner kopiert werden.

allgemeine Syntax: cp source-file target-file

```
1 cp text.txt ..
```

Alternative: Man wechselt in den Zielordner und adressiert diesen mit .

Syntax: cp source-file .

Wir wechseln daher in den übergeordneten Ordner:

```
1 cd ..
```

/Users/martin/Workspace

und kopieren dann die Datei "text.txt" in diesen hinein.

```
1 cp mydata/text.txt .
```

Überprüfung des Ergebnisses mit `ls`

```
1 ls
```

```
Java_Projects/      Lecture_Projects/  SQL_Projects/      vscode setup/
Jupyter_Notebooks/ mydata/            text.txt
```

Nun soll die doppelte Datei "text.txt" im Unterordner "mydata" gelöscht werden:

```
1 rm mydata/text.txt
```

```
1 ls mydata
```

Welche Texte sind in der Datei "text.txt" gespeichert? Um in die Datei zu schauen, muss man einen Editor, am einfachsten einen Editor des Terminals verwenden.

Commandline Editoren:

- vim: hohe Lernkurve, mächtig, wird von vielen Programmierern und Systemadministratoren verwendet.
- Aufruf: `vim text.txt`
- nano: intuitiver zu bedienen, nicht so mächtig.
- Aufruf: `nano text.txt`

Die Datei "text.txt" und den Ordner "mydata" benötigen wir nun nicht mehr und löschen diese. Anstatt die Datei "text.txt" mit dem Befehl `rm text.txt` und den Ordner mit dem Befehl `rmdir mydata` getrennt zu löschen, werden wir den Ordner samt und Datei einzeln mit dem Befehl `rm` zu löschen, verwenden wir den Befehl `rmdir` mit dem Flag `-R` (R steht für rekursiv), der dafür sorgt, dass der Ordner mitsamt Inhalt gelöscht wird. Damit das funktioniert, darf der aktuelle Ordner nicht der zu löschende Ordner oder ein Unterordner sein.

```
1 rm -R mydata
```

Informationen zu einem bestimmten Terminal-Befehl erhält man mit der `man`-Anweisung:

```
1 man rmdir
```

- [CheatSheet zum Terminal \(pdf\)](#)

1.2 Microsoft Visual Studio Code (VS Code)

Visual Studio Code (VS Code) ist mehr als ein Texteditor, aber schlanker als eine vollwertige IDE. Es bietet komfortables Programmieren mit Features wie Autovervollständigung und Syntaxhervorhebung. Ein integriertes Terminal erleichtert den Workflow. Durch Extensions lässt sich VS Code flexibel erweitern und anpassen.

[CheatSheet zu VS Code](#)

Extensions installieren

In Visual Studio Code in der linken Leiste auf das Symbol mit den vier Quadraten drücken. In der Suchleiste folgende vier Extensions suchen und installieren:

- Python
- Pylance
- Flake8 -autopep8
- Jupyter

VS-Code Extensions einrichten

1. Settings (Einstellungen) öffnen. Nach Auto Save suchen (Suchfeld) und auf after-Delay stellen.
2. Nach Type Checking Mode suchen und auf basic stellen.

Nach Diagnostic Severity Overrides suchen und auf edit in settings.json drücken. Hier nach folgendem Eintrag suchen:

```
` "python.analysis.diagnosticSeverityOverrides": {  
},`
```

und folgende zwei Einträge einfügen, sodass das ganze wie folgt aussieht:

```
`"python.analysis.diagnosticSeverityOverrides": {  
    "reportGeneralTypeIssues": "information",  
    "reportWildcardImportFromLibrary": "none"  
},`
```

settings.json wieder schließen.

Nach autopep8 suchen und bei autopep8-Einstellung Argv auf Add Item drücken. Folgendes einfügen und anschließend auf OK drücken:

```
--ignore=F4,E123,E501,E704,E731,W5,C901
```

Nach Flake8 suchen und bei Flake8 Argv auf Add Item drücken. Folgendes einfügen und anschließend auf OK drücken:

```
--ignore=F4,E123,E501,E704,E731,W5,C901
```

Nach Severity suchen und bei Flake8 Severity alle Einträge unter Value auf Warning setzen. Dazu rechts auf den Stift zum Bearbeiten drücken und bei Value Warning auswählen (auch hier wieder mit OK bestätigen)

1.3 Jupyter Notebook

- quelloffene und kostenlose WebAnwendung
- Integriert
 - Markdown Umgebung
 - (Python) Umgebung
- Verwendung von Data Scientists
- Austausch von Daten
- bis zu 40 Programmiersprachen werden unterstützt
- Dokumente gut lesbar

1.4 Markdown Language

Markdown ist eine einfache Auszeichnungssprache. Sie ist leicht zu erlernen. Die wichtigsten Tags sind im Folgenden zusammengefasst.

Ergebnis	Markdown-Code
Überschrift 1	# Überschrift 1
Überschrift 2	## Überschrift 2
Überschrift 3	### Überschrift 3
Überschrift 4	#### Überschrift 4
Überschrift 5	##### Überschrift 5
fetter Text	**fetterText**
<i>kursiver Text</i>	<i>*kursiver Text*</i>
• Element	* Element
> Zitat	> Zitat
InlineCode	`InlineCode`
Codeblock	```Codeblock```
– horizontale Linie	----
* escaped Stern	* escaped Stern
- Element - Unterelement	- Element\n - Unterelement

[CheatSheet zu Markdown \(pdf\)](#)

1.5 LaTeX

LaTeX ist ein Textsatzsystem, das besonders für wissenschaftliche Arbeiten und Dokumente verwendet wird. Es eignet sich hervorragend zum Schreiben von Texten mit Formeln, Tabellen und Literaturverzeichnissen. Anders als in Word wird der Text in LaTeX mit Befehlen und Codes geschrieben.

Innerhalb der Markdown-Umgebung kann LaTeX für mathematische Formeln verwendet werden. Hier sind einige grundlegende LaTeX-Befehle:

Beschreibung	LaTeX-Befehl
Inline-Formeln	$\text{\texttt{\$}\texttt{\text{Formel}}\texttt{\$}}$
Block-Formeln	$\text{\texttt{\$}\texttt{\text{Formel}}\texttt{\$}}$

Beschreibung	LaTeX-Befehl
α	<code>\alpha</code>
$\frac{z}{n}$	<code>\frac{z}{n}</code>
\sqrt{a}	<code>\sqrt{a}</code>
$\sum_{i=1}^n i$	<code>\sum_{i=1}^n i</code>
$\int_a^b f(x) dx$	<code>\int_{a}^{b} f(x) \, dx</code>
\vec{v}	<code>\vec{v}</code>
$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$	<code>\begin{pmatrix} a & b \\ c & d \end{pmatrix}</code>
$a = b$	<code>a = b</code>
$c = d$	<code>c = d</code>

[CheatSheet zu LaTeX \(pdf\)](#)

1.6 Python Buch als Openbook

Johannes Ernesti, Peter Kaiser: Python3. <https://openbook.rheinwerk-verlag.de/python/index.htm>