

Toolbox Workshop

PeP et al. Toolbox Workshop



PeP et al. e.V.

Physikstudierende und
ehemalige Physikstudierende
der TU Dortmund

2020

Auf das Praktikum vorbereiten

Daten auswerten

Plotten

Fehlerrechnung

Technische Fähigkeiten, die man in der Wissenschaft braucht

Konkrete Probleme durch Programmieren lösen

Wiederholte Abläufe automatisieren

Versionskontrolle: Wieso? und Wie?

Kommandozeile

Von Anfang an: Best Practices

Spart Zeit und Nerven Verwenden von Dokumentation

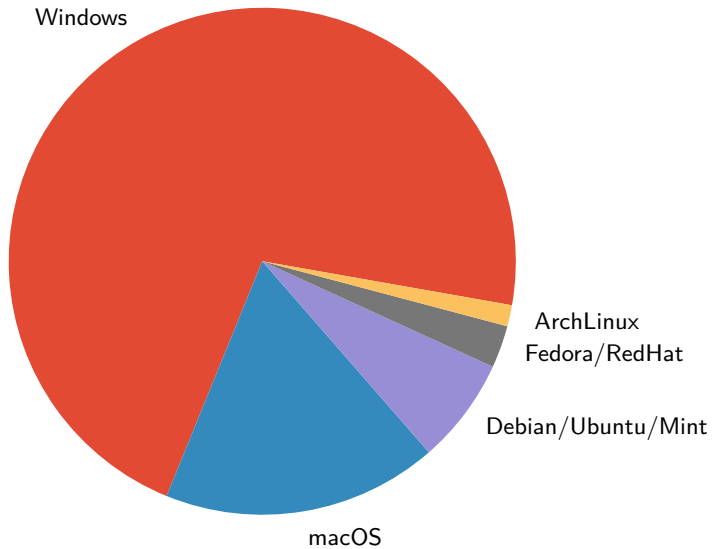
Was sind die Standardwerkzeuge?

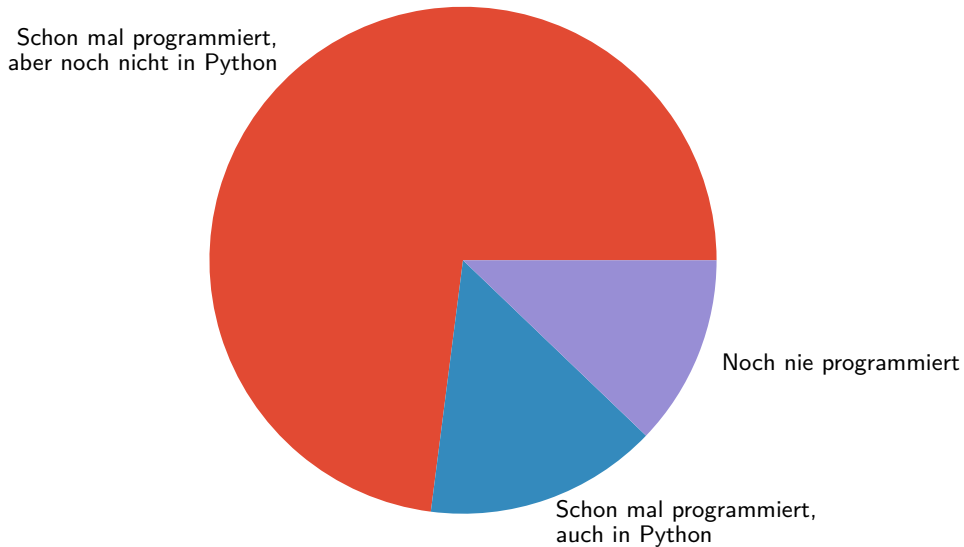


Toolbox Workshop

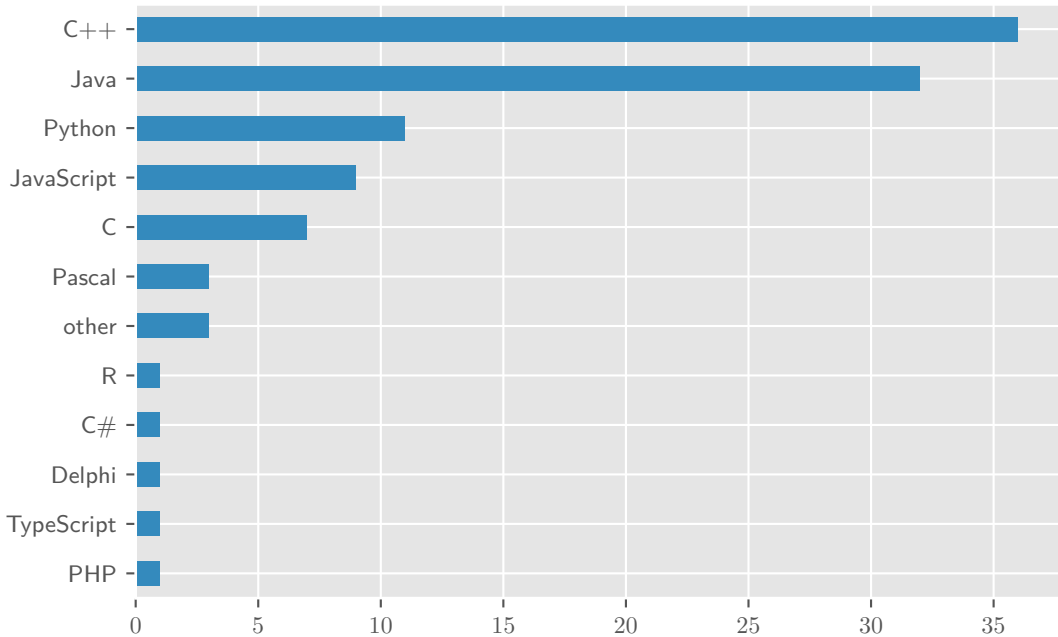


Ergebnisse der Umfrage

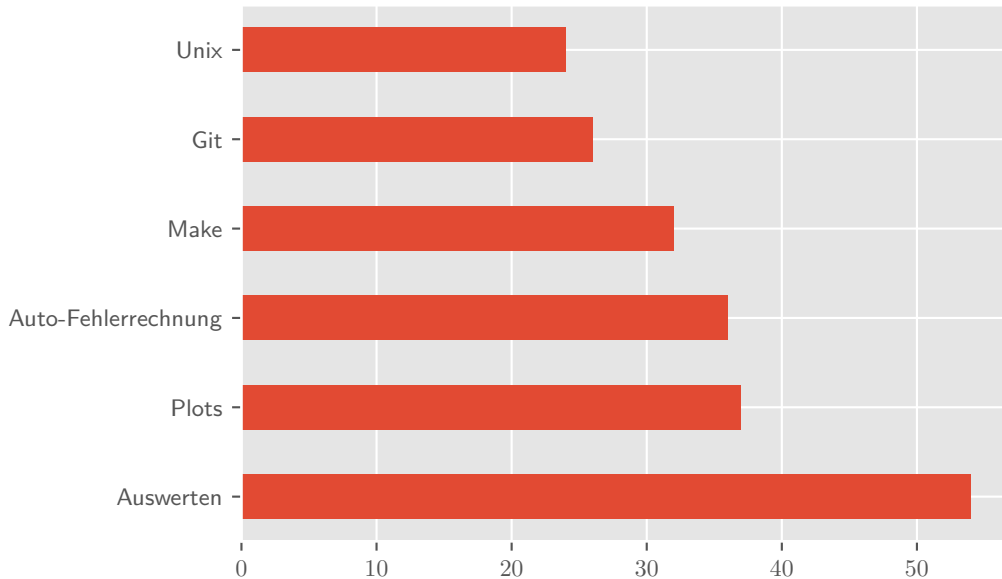




Programmiersprachen



Interessen



Montag Programmieren mit Python

Dienstag Erstellen von Plots / Auswerten

→ NumPy

→ matplotlib

Mittwoch Auswerten / Fehlerrechnung

→ scipy

→ uncertainties

Donnerstag Kommandozeile und Automatisierung

→ Unix

→ make

Freitag Versionskontrolle

→ git

→ Abschließende Übungen

And now for something completely different...

Texteditoren

Was haben die mit diesem Kurs zu tun?

- Viele Dateien, denen man in der Wissenschaft begegnet, enthalten (plain) text
 - Paper/Arbeiten mit \LaTeX
 - Programm-Code
 - Config-Files
 - Notizen
 - Daten (csv, json, yaml, ...)
 - Emails
- Es lohnt sich also, einen guten Texteditor zu wählen und den Umgang damit zu erlernen!
- Das spart auf lange Sicht Zeit und macht die Arbeit angenehmer
- Zwei Varianten: Terminal / GUI

Was ist eigentlich eine Textdatei?

- In einer Datei stehen immer Binärdaten in Bytes, 1 Byte = 8 Bit, 0-255
- Es gibt (gab) viele Varianten, Text in Binärdaten umzuwandeln (Encoding)
- Heute sollte immer Unicode enkodiert als **utf-8** verwendet werden

Unicode → Sammlung von Schriftzeichen, Buchstaben, Akzente, Emojis, ...
→ Aus allen Sprachen.
→ Ordnet Zeichen „Codepoints“ zu
→ Beispiele: **LATIN SMALL LETTER A: 97, PILE OF POO: 128169**

UTF-8 Encoding um Unicode-Text in Bytes zu speichern

Windows und Unix-Systeme verwenden unterschiedliche Konventionen für ein Zeilenende.

Unix `\n` / LF (Linefeed)

Windows `\r\n` / CR LF (Carriage Return + Linefeed).

VS Code erkennt auf allen Betriebssystemen welche Konvention im aktuellen File genutzt wird und behält sie bei.

Empfehlung: immer Unix-Konvention nutzen

Was muss ein Editor können?

In absteigender Wichtigkeit

- Zeilennummern
- Syntax-Highlighting
- Simple Autovervollständigung
- Plugins / Anpassbarkeit
- Linting (Warnhinweise für falschen Code)
- Komplexe Autovervollständigung (Snippets, Library-Funktionen)

Nano

```
      :::  
iLE88Dj. :jD88888Dj:  
.LGitE888D.f8GjjjL8888E;  
iE :8888Et. .G8888.  
;i  E888,      ,8888,  
    D888,      :8888,  
    D888,      :8888,  
    D888,      :8888,  
    D888,      :8888:  
    888W,      :8888:  
    W88W,      :8888:  
    W88W:      :8888:  
    DGGD:      :8888:  
                :8888:  
                :W888:  
                :8888:  
                E888i  
                tw88D
```

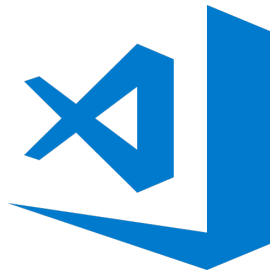
- Einfacher Texteditor fürs Terminal
- Auf fast jedem Unix-System vorhanden
- Wenige Features, nicht erweiterbar

Vim

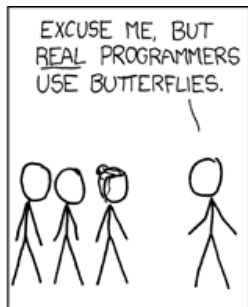
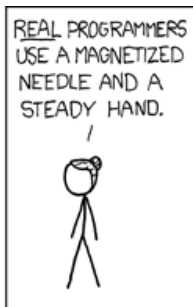
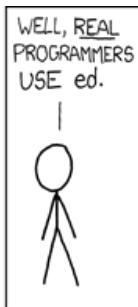
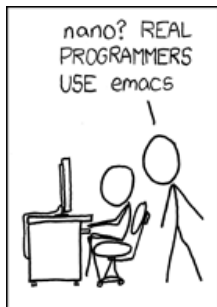


- Moden-basiert
- Erweiterbar
- Auf fast jedem Unix-System default
- Harter Einstieg

VS Code



- GUI Editor von Microsoft
- Leichter zu bedienen
- Batteries included
- Viele nützliche Plugins



THE DISTURBANCE RIPPLES OUTWARD, CHANGING THE FLOW OF THE EDDY CURRENTS IN THE UPPER ATMOSPHERE.



THESE CAUSE MOMENTARY POCKETS OF HIGHER-PRESSURE AIR TO FORM,

WHICH ACT AS LENSES THAT DEFLECT INCOMING COSMIC RAYS, FOCUSING THEM TO STRIKE THE DRIVE PLATTER AND FLIP THE DESIRED BIT.

