

```
38 self.file.seek(0)
39 self.fingerprints.update(retrieved)
40
41
42 @classmethod
43 def from_settings(cls, settings):
44     debug = settings.getbool('SUPERFILTER_DEBUG')
45     return cls(job_dir(settings), debug)
46
47 def request_seen(self, request):
48     fp = self.request_fingerprint(request)
49     if fp in self.fingerprints:
50         return True
51     self.fingerprints.add(fp)
```

Programmierkurs 2023 - Intro

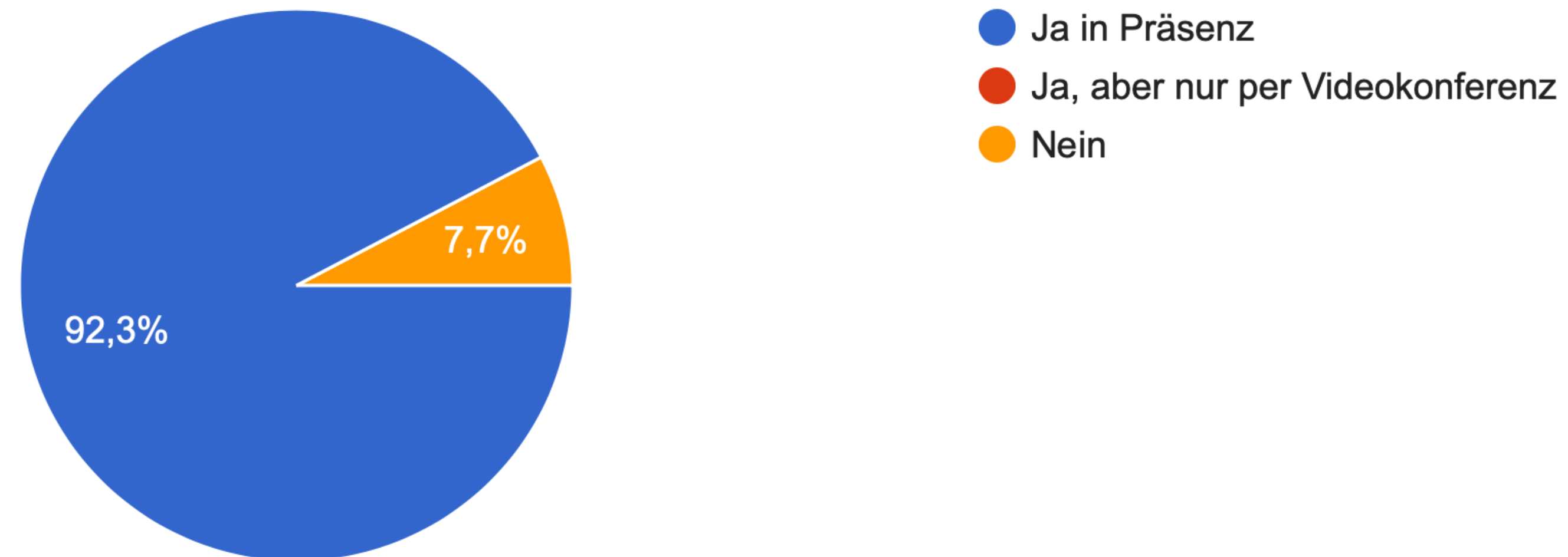
Jannis Speer, Jan Langer, Jonah Blank

Informationen zum Kurs

- ▶ Fragen sind herzlich willkommen
- ▶ Freiwillige Teilnahme
- ▶ Über konstruktives Feedback freuen wir uns
 - Positiv oder negativ
 - Während des Kurses
- ▶ Programm aufgebaut auf Grundlage der Umfrage

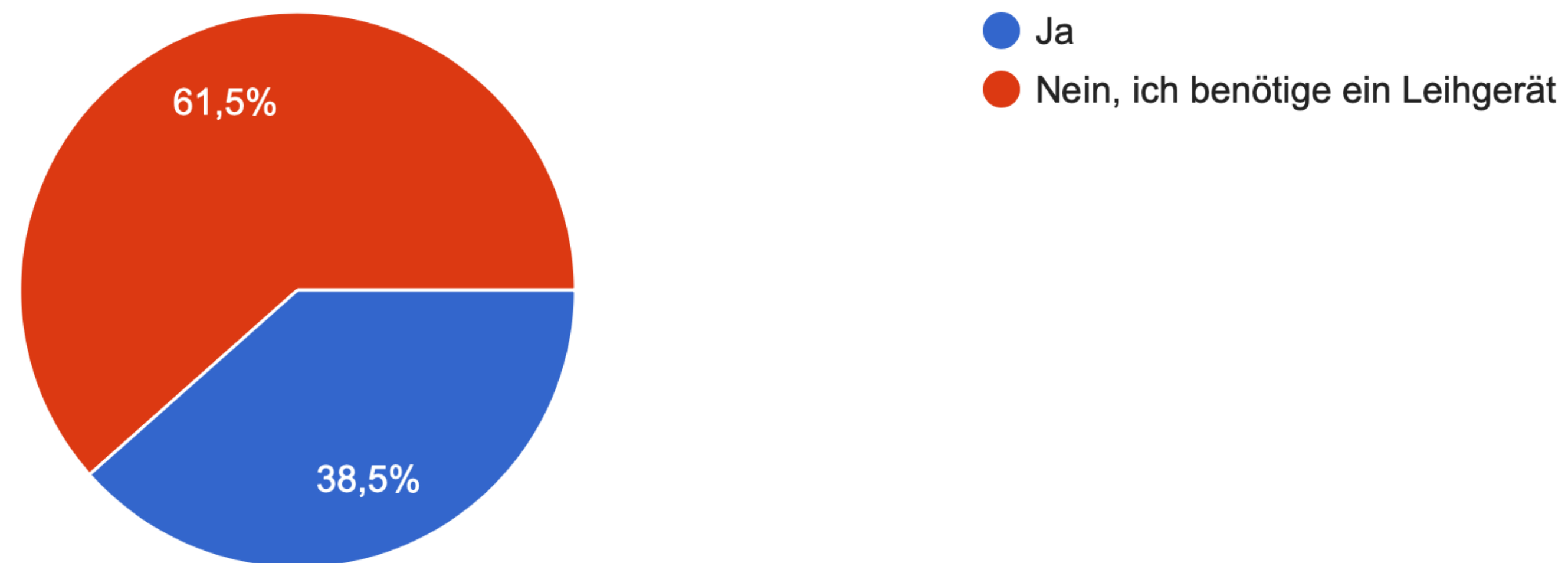
Werden Sie den Programmierkurs vom 28.3.2023 - 31.3.2023 besuchen?

13 Antworten



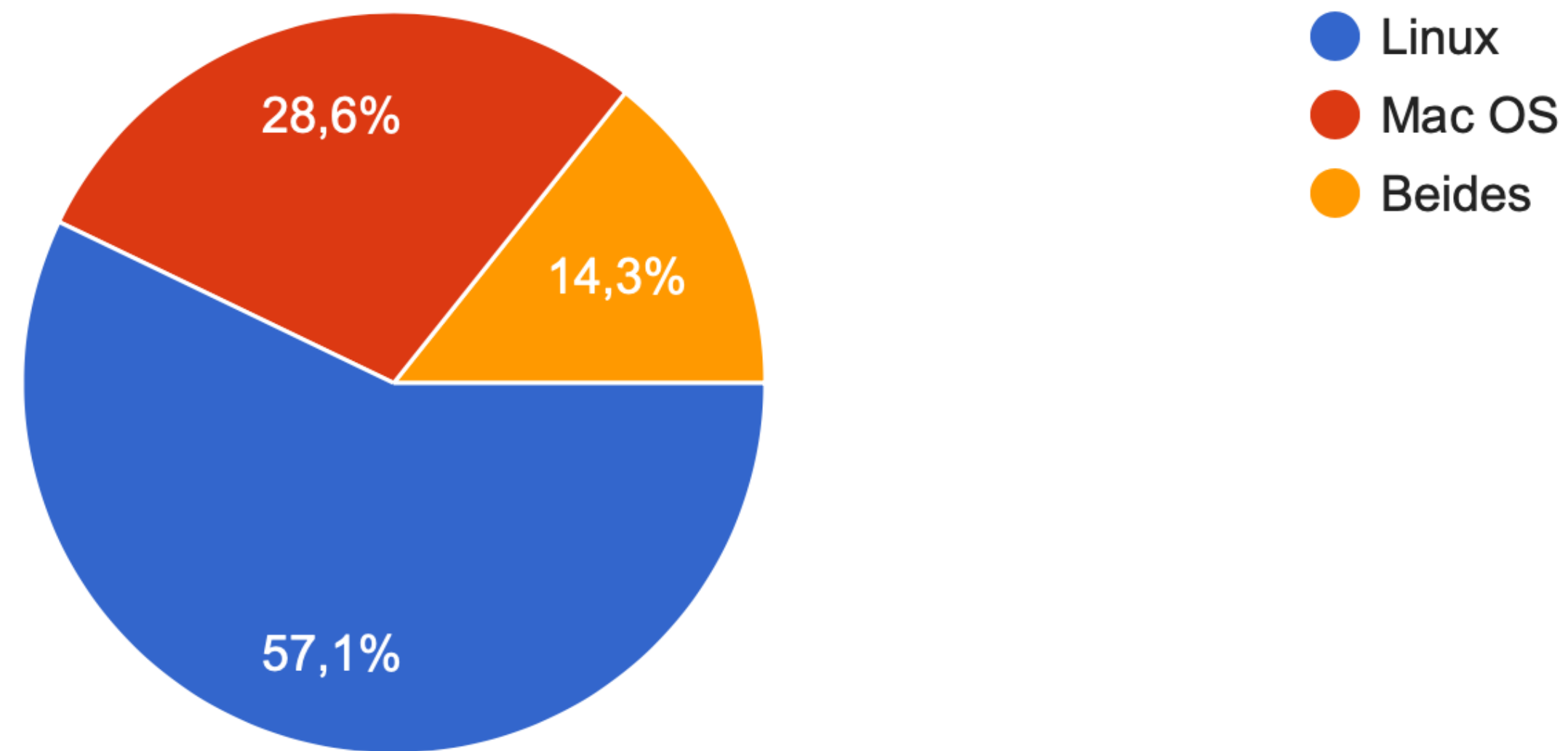
Haben Sie einen eigenen Laptop für den Kurs? (Mac OS oder Linux)

13 Antworten



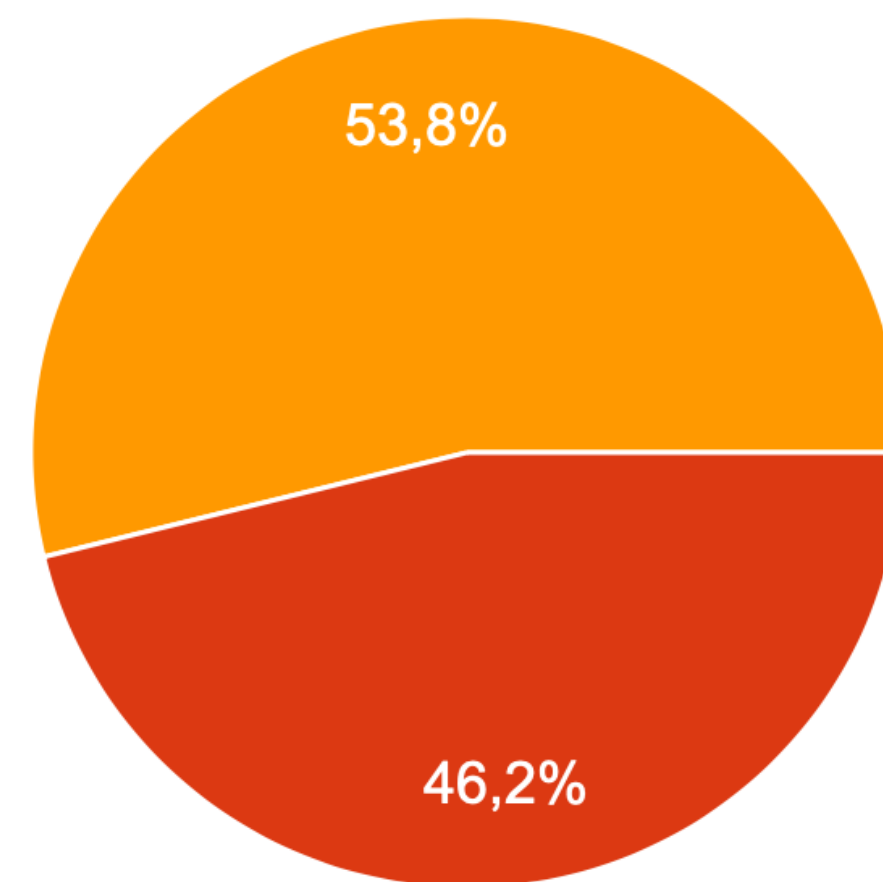
Haben Sie Erfahrung mit Linux und/oder Mac OS?

7 Antworten



Haben Sie Programmiererfahrung?

13 Antworten



- Ich habe noch nie Quellcode geschrieben
- In Vorlesung X mussten wir mal Sprache Y nutzen
- Ich habe schon kleinere oder größere Projekte umgesetzt

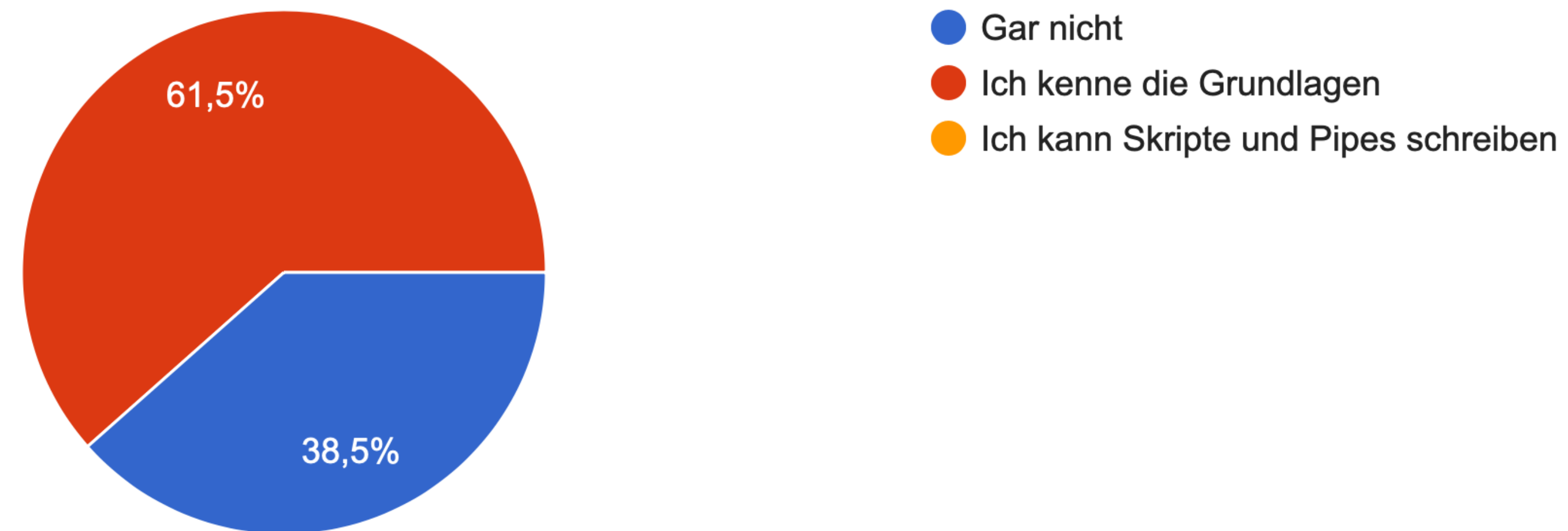
Wie gut kennen Sie sich mit der Programmiersprache Python aus?

13 Antworten



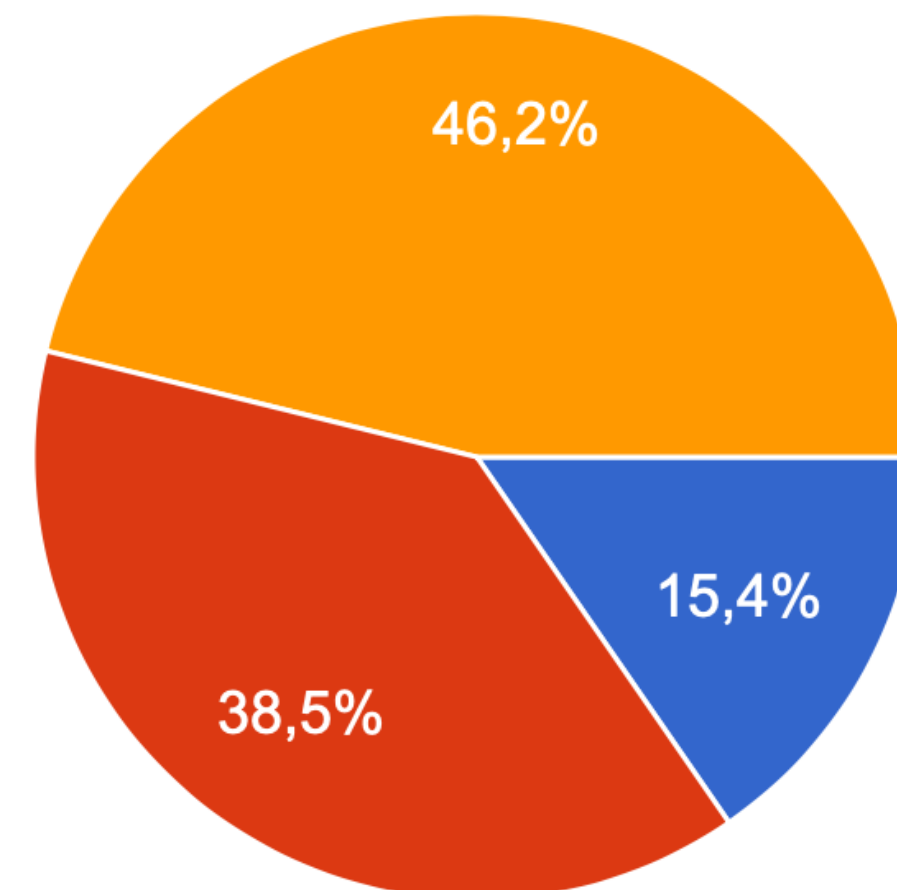
Wie gut kennen Sie die Unix-Kommandozeile (bash, zsh etc.)?

13 Antworten



Wie gut kennen Sie das Versionskontrollsystem Git?

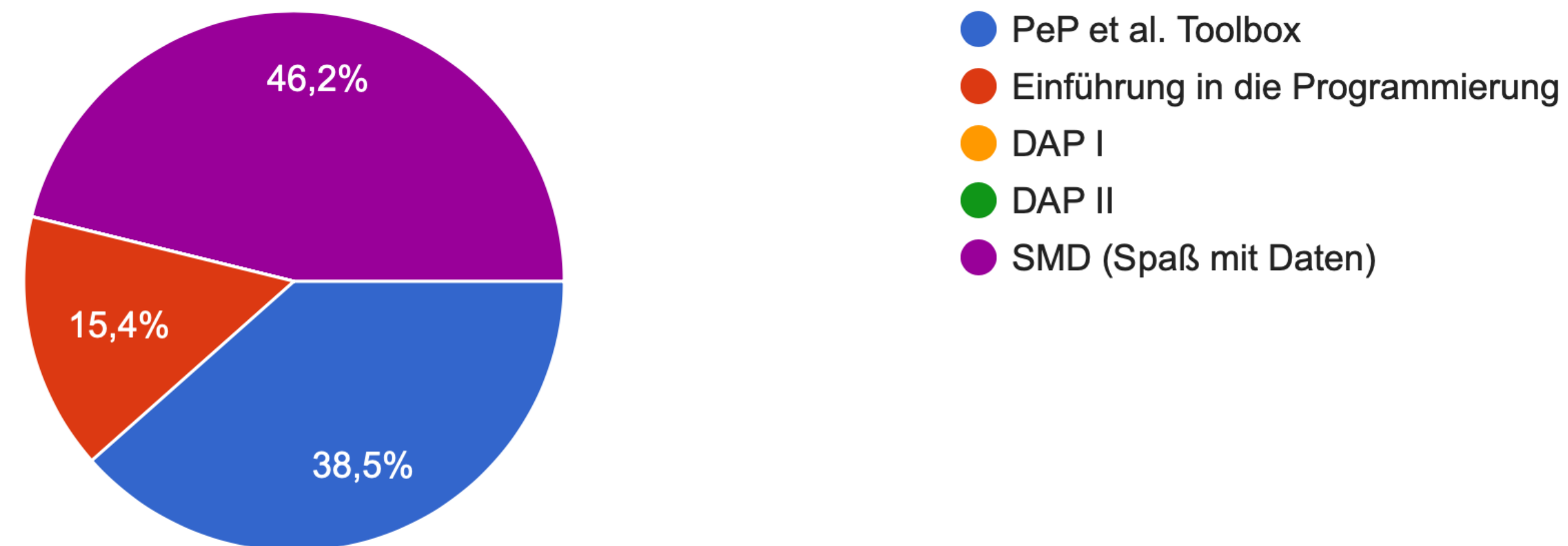
13 Antworten



- Gar nicht
- Ich habe damit schon mal ein Repository geklont
- Ich habe es schon benutzt um mit anderen über Github/GitLab zusammenzuarbeiten

Welche Programmierveranstaltungen haben Sie besucht?

13 Antworten



Kursprogramm

	Tag 1	Tag 2	Tag 3	Tag 4
8:30 - 11:30	Intro & Installation	Python (basics)	Python (Advanced I)	ROOT / Scikit-HEP / Geant4
13:00 - 16:00	Bash & Git	Python & Snakemake	Python (Advanced II)	ROOT / Scikit-HEP / Geant4

Agenda auf [Indico](#)

Kursmaterial

- ▶ Material liegt in der Nextcloud
 - Alle Folien
 - Code-Beispiele / Übungsaufgaben
 - Wird während des Kurses weiter bearbeitet
- ▶ Zugriff auf die Nextcloud während der Installations Session
- ▶ Nutzung der Nextcloud über
 - Browser
 - App

Termine

► Meetings

- Gruppenmeeting: Dienstag um 12:15
- Medizinphysik: Montag 14:00
- AstroQCD: Mittwoch um 16:30
- Rare Decays: Donnerstag um 13:00
- CPV: Freitag um 10:15
- Hardware: Freitag um 13:00
- RTA: Mittwoch um 12:00

► Weitere gruppenspezifische Meetings

- Fragt eure direkten Betreuer

► Gemeinsames Mittagessen

- Treffen um 11:20
- Essen um 11:30

► Heißgetränke

- Empfehlung: Telegram-Bot abonnieren
@e5traditionsbot mit /start anschreiben



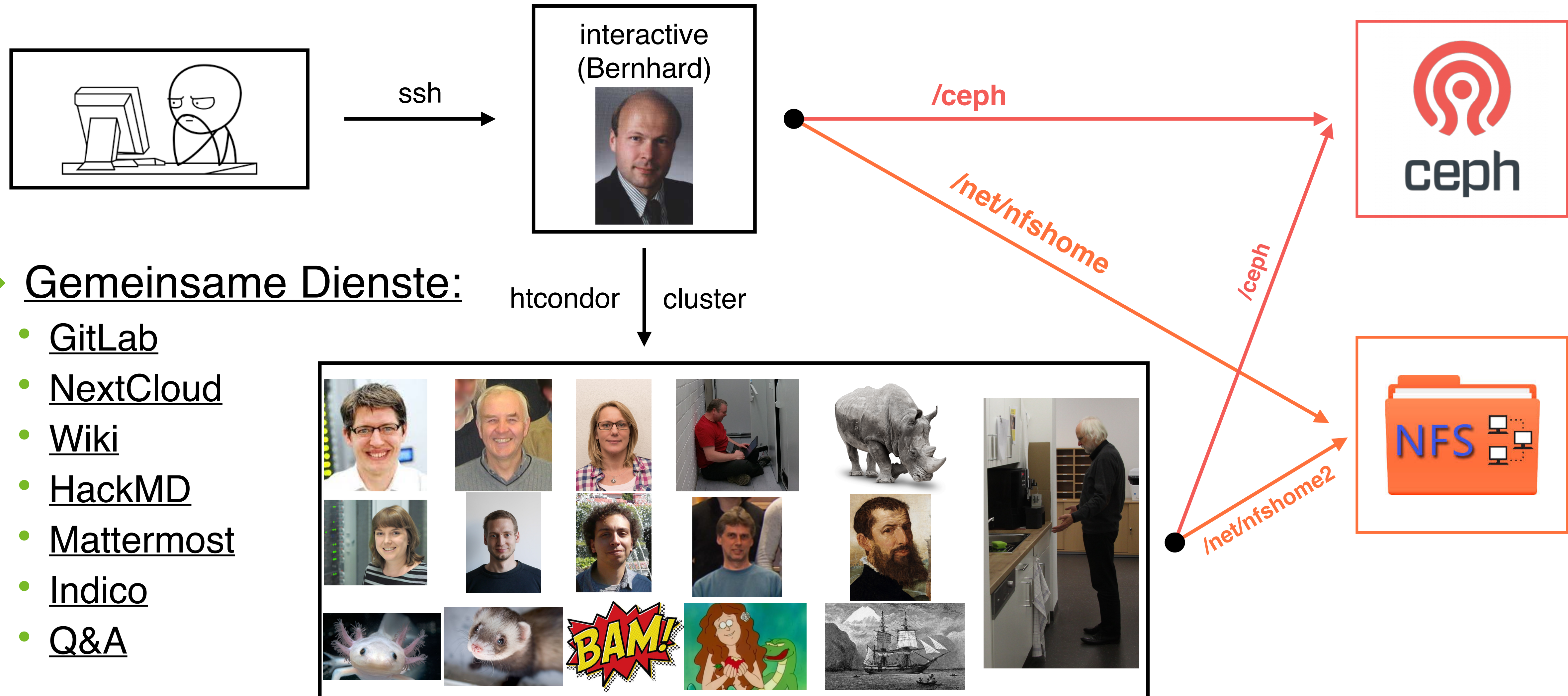
Mac

Setup & Installation Solutions



Einrichtung von Hardware und Software

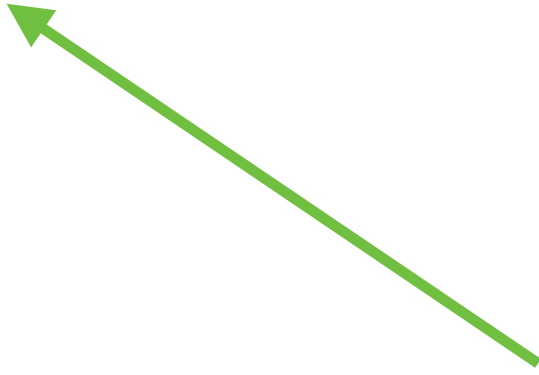
Infrastruktur



E5 Docker Cloud

Gemeinsame Dienste

- [GitLab – Collaborative Coding](#)
- [NextCloud – File Sharing](#)
- [Indico – Meetings and Room Booking](#)
- [Wiki](#)
- [HackMD – Collaborative Note Taking](#)
- [Mattermost Chat](#)
- [QA/stackoverflow Service](#)
- [Kaffee-Ministerium](#)
- [Pizza-Bestellung](#)
- [Doxygen](#)

A green arrow points from the URL text to the 'Gemeinsame Dienste' header.

Homepage für alle unsere Services
abrufbar unter folgender URL:
<https://cloud.e5.physik.tu-dortmund.de/>

Vorbereitung auf den Kurs

- ▶ Ihr bekommt Zugang zur Infrastruktur über LDAP Accounts
- ▶ Einloggen über
 - Services: LDAP-login
 - Interaktive Maschinen: über einen im LDAP Account hinterlegten ssh-key
- ▶ Schickt dazu eine Mail an jonah.blank@tu-dortmund.de
- ▶ Die Mail sollte umfassen
 - Vorname, Nachname
 - TU Email-Adresse
 - Public ssh-keys
- ▶ Accountname wird dann lauten
 - ➔ 1. Buchstabe des Vornamens + Nachname
Beispielsweise: jspeer

```

31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
def __init__(self, settings):
    self.file = None
    self.fingerprints = set()
    self.logdupes = True
    self.debug = debug
    self.logger = logging.getLogger(__name__)
    if path:
        self.file = open(os.path.join(path, 'fingerprint.log'), 'a')
        self.file.seek(0)
        self.fingerprints.update(self.get_fingerprints())

    @classmethod
    def from_settings(cls, settings):
        debug = settings.getbool('debug', False)
        return cls(job_dir(settings), debug)

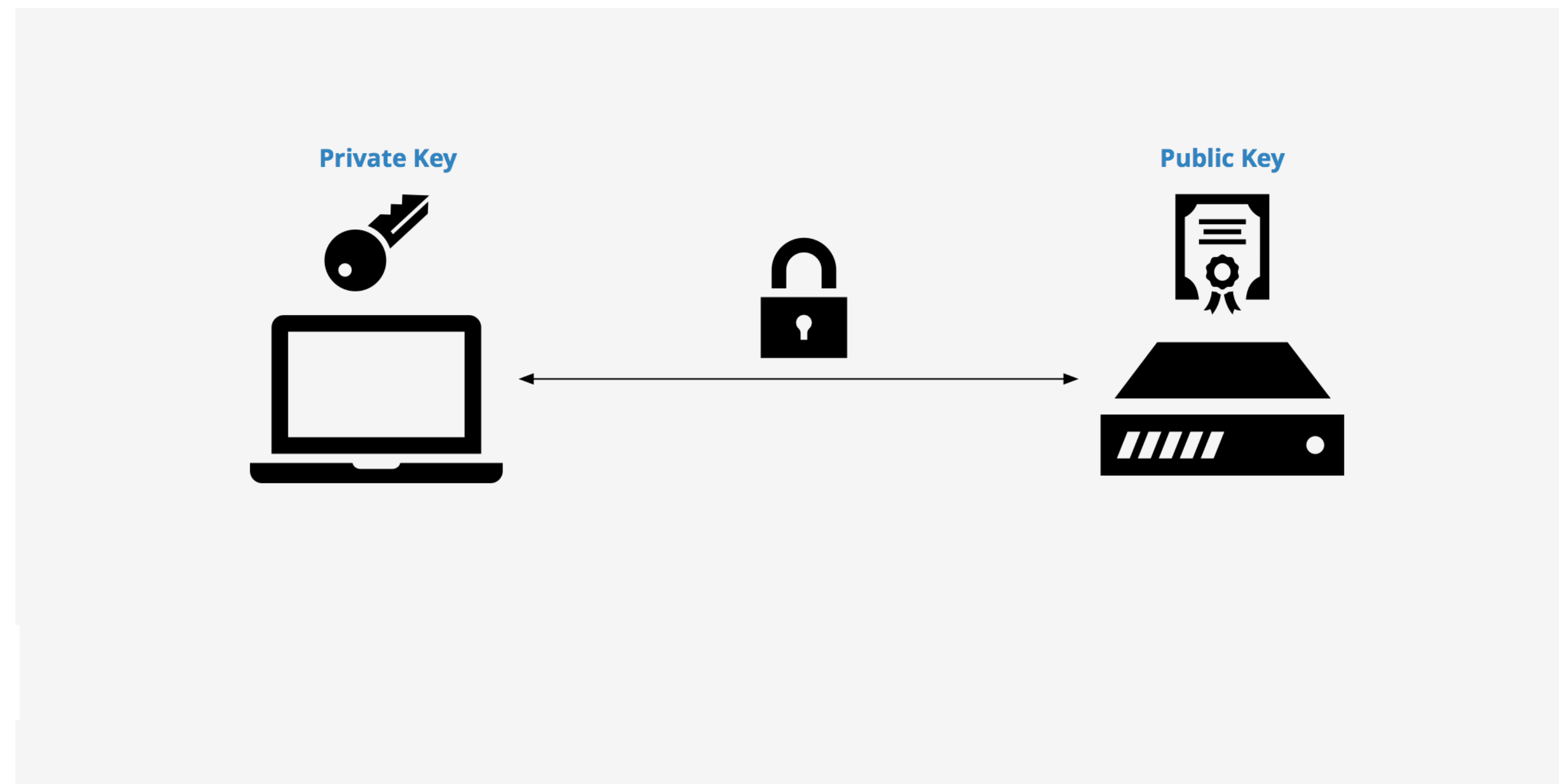
    def request_seen(self, request):
        fp = self.request_fingerprint(request)
        if fp in self.fingerprints:
            return True
        self.fingerprints.add(fp)
        if self.file:
            self.file.write(fp + os.linesep)

    def request_fingerprint(self, request):
        return request_fingerprint(request)

```


Anleitung zur Erstellung eines ssh-keys

- ▶ Ssh ist ein Netzwerkprotokoll für Zugriff auf einen anderen Computer
- ▶ Verwendung von Schlüsselpaaren zur Authentifizierung
- ▶ Anleitung für Setup findet ihr unter folgendem Link:
<https://git.e5.physik.tu-dortmund.de/e5/ssh-config>



Vorbereitende Installationen

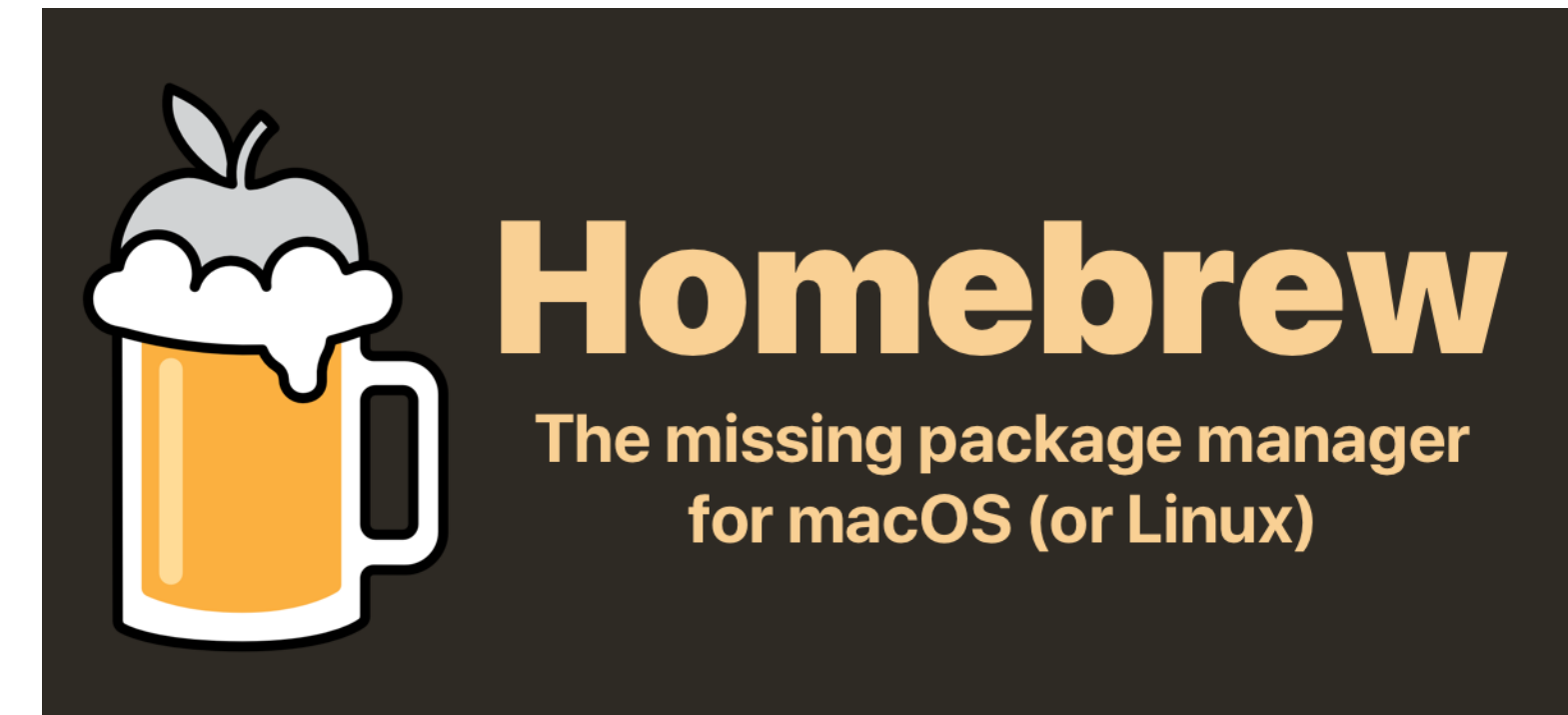
► Empfohlen:

- Text Editor
 - VisualStudio Code (vscode)
 - Sublime-Text
- Git
- Python (version 3.x)
- Nextcloud
- sshfs (auf OSX via macfuse)
- Chats
 - Mattermost
 - Telegram-Desktop

► Installation: nächste Seiten

Installieren mit Homebrew I

- ▶ Homebrew
 - Vergleichbar mit apt oder yum auf Linux
 - Einfache Wartung, übersichtlich
- ▶ Homebrew installieren
 - Im Browser [brew.sh](#) öffnen
 - Anweisungen dort folgen



Installieren mit Homebrew II

Installieren mehrerer Programme:

- ▶ Brewfile anlegen oder runterladen (siehe rechts)
- ▶ `brew bundle dump --file filename`
erstellt eine Brewfile aus euren installierten Programmen
- ▶ `brew bundle install --file filename`
installiert Programme aus Brewfile

E5a Brewfile benutzen

- ▶ Brewfile mit dem nötigsten liegt hier
 - Durch entfernen von Kommentaren ("##") installiert ihr zusätzliche Programme
- ▶ Installation einiger Software auf diese Weise nicht möglich
 - Überprüft ob es Fehlermeldungen gab
 - Im Notfall manuelle Installation

- ▶ Probleme mit der Installation von sshfs via brew:
 - Direkte Installation via osxfuse
 - Download und Installation von macFUSE und SSHFS