

```
38 self.file.seek(0)
39 self.fingerprints.update(...)
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 - Intro

Jan Langer | jan.langer@tu-dortmund.de

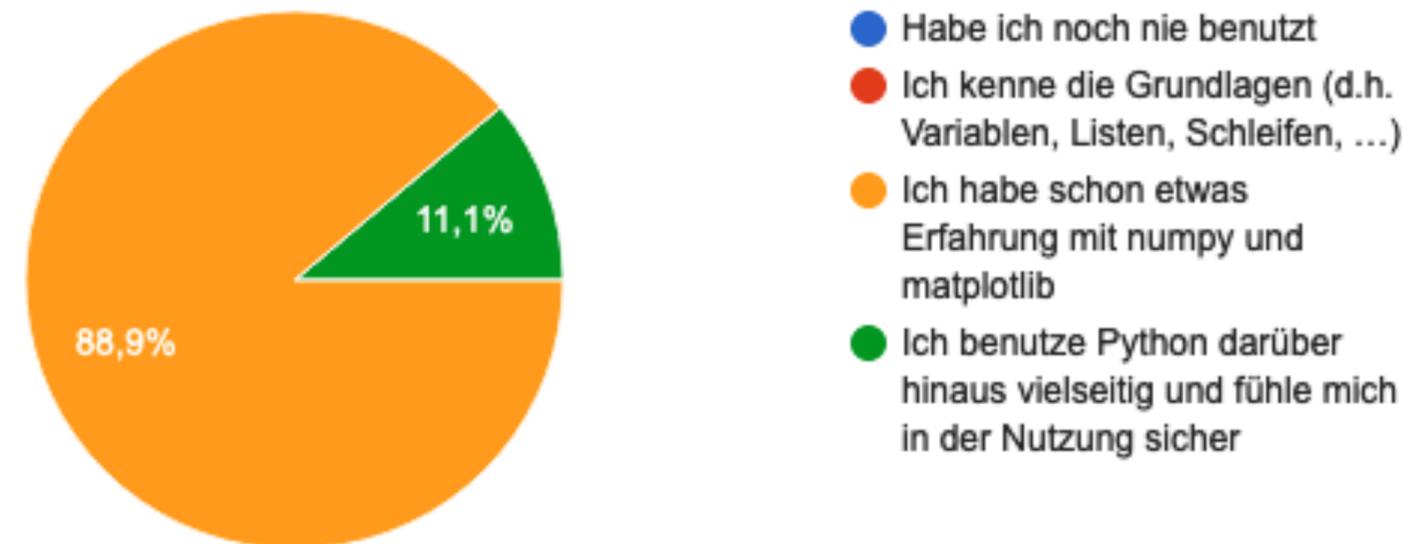
24.02.2020

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

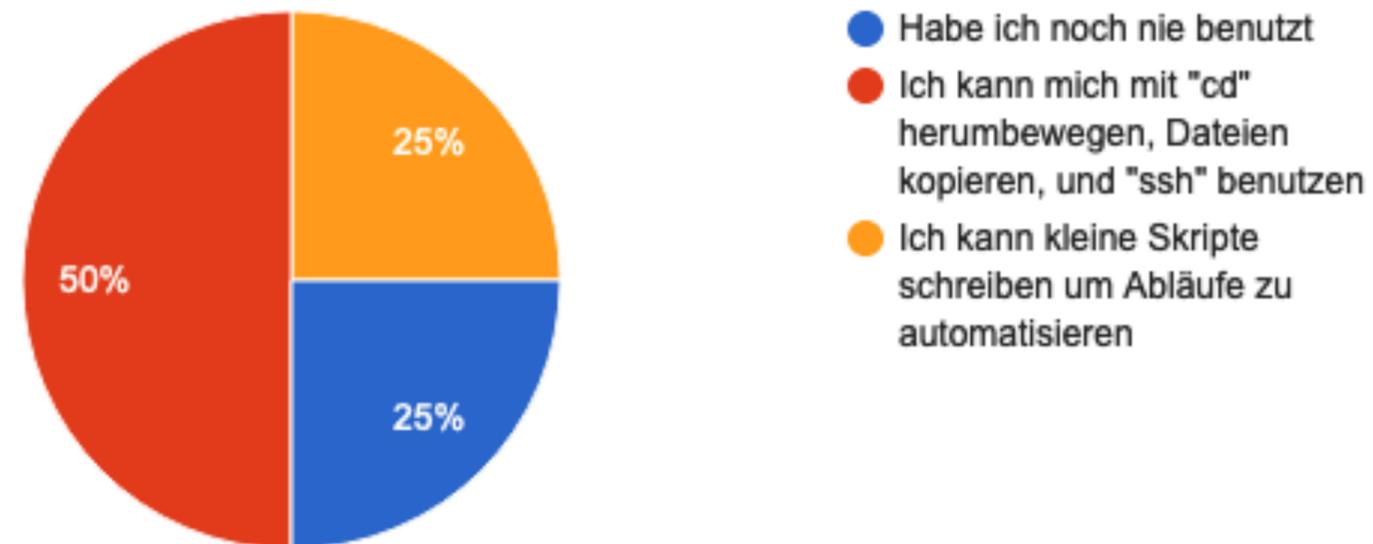
Wie gut kennst du dich mit der Programmiersprache Python aus?

9 Antworten



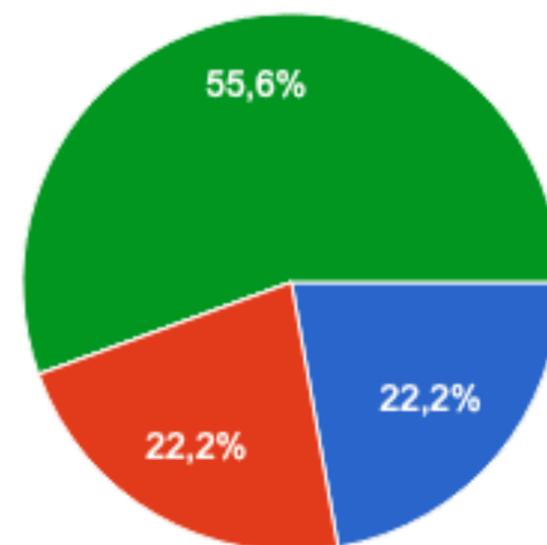
Wie gut kennst du dich mit der Kommandozeile (z.B. bash, zsh, ...) aus?

8 Antworten



Wie gut kennst du dich mit dem Versionskontrollsystem git aus?

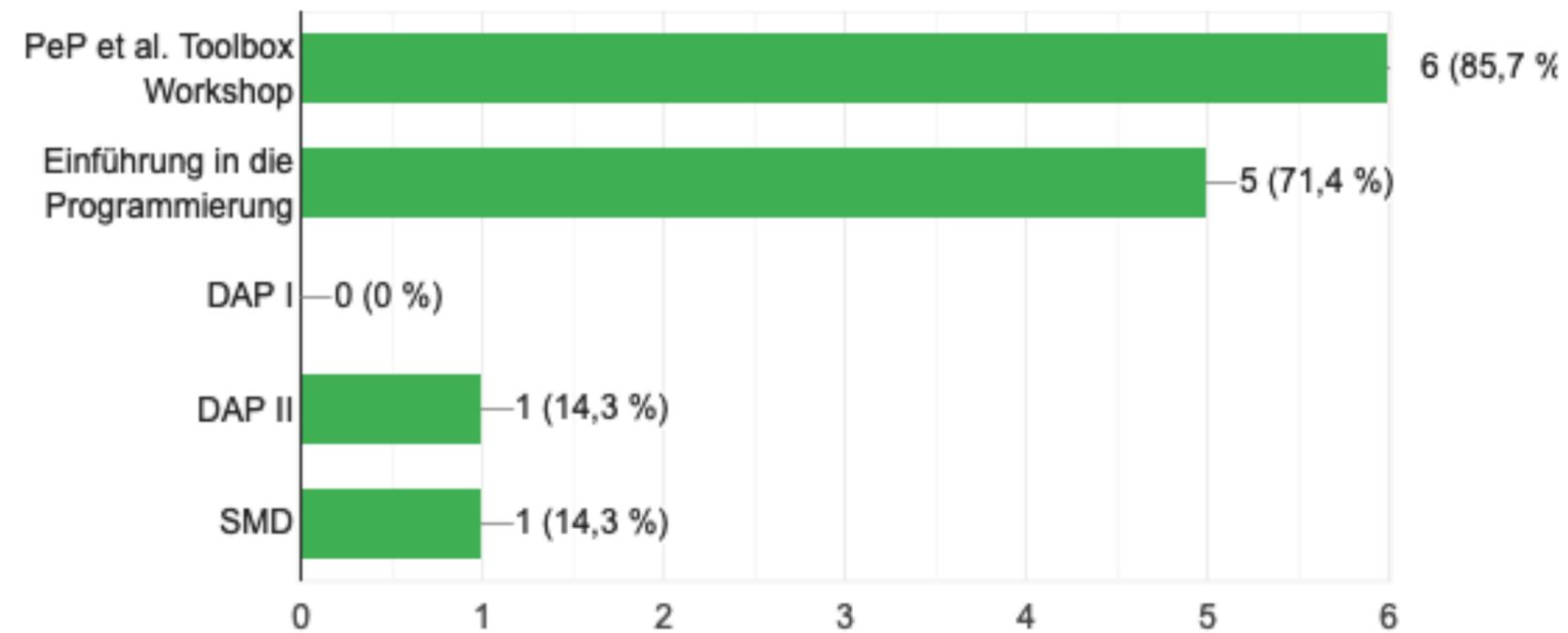
9 Antworten



- Habe ich noch nie benutzt
- Ich habe davon gehört, aber benutze lieber Dropbox, Google Drive, oder ähnliches
- Ich habe schon mal ein Repository "gecloned"
- Ich habe es schon benutzt um mit anderen zusammenzuarbeiten

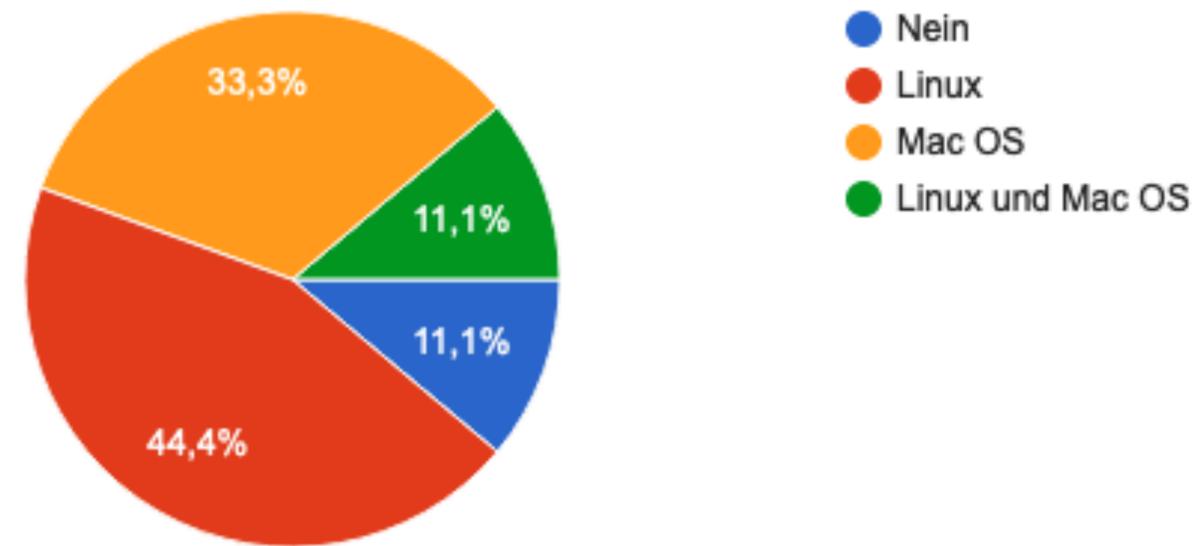
Welche der folgenden Programmierveranstaltungen hast du besucht?

7 Antworten



Hast du Erfahrung mit Linux und/oder Mac OS?

9 Antworten



Kursprogramm

	Tag 1	Tag 2	Tag 3	Tag 4
8:30 - 11:30	Intro & Installation	Python (basics)	Python advanced	ROOT & ML & Geant 4
13:00 - 16:00	Bash & Git	Python & Snakemake		Roofit & ML & Geant 4

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: Mittwoch um 09:00
- Analyse: Montag um 12:00
- Hardware: Freitag um 9:30
- Medizinphysik: Freitag um 10:00

► Gruppenspezifische Meetings

- Fragt eure direkten Betreuer

► Heißgetränke

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



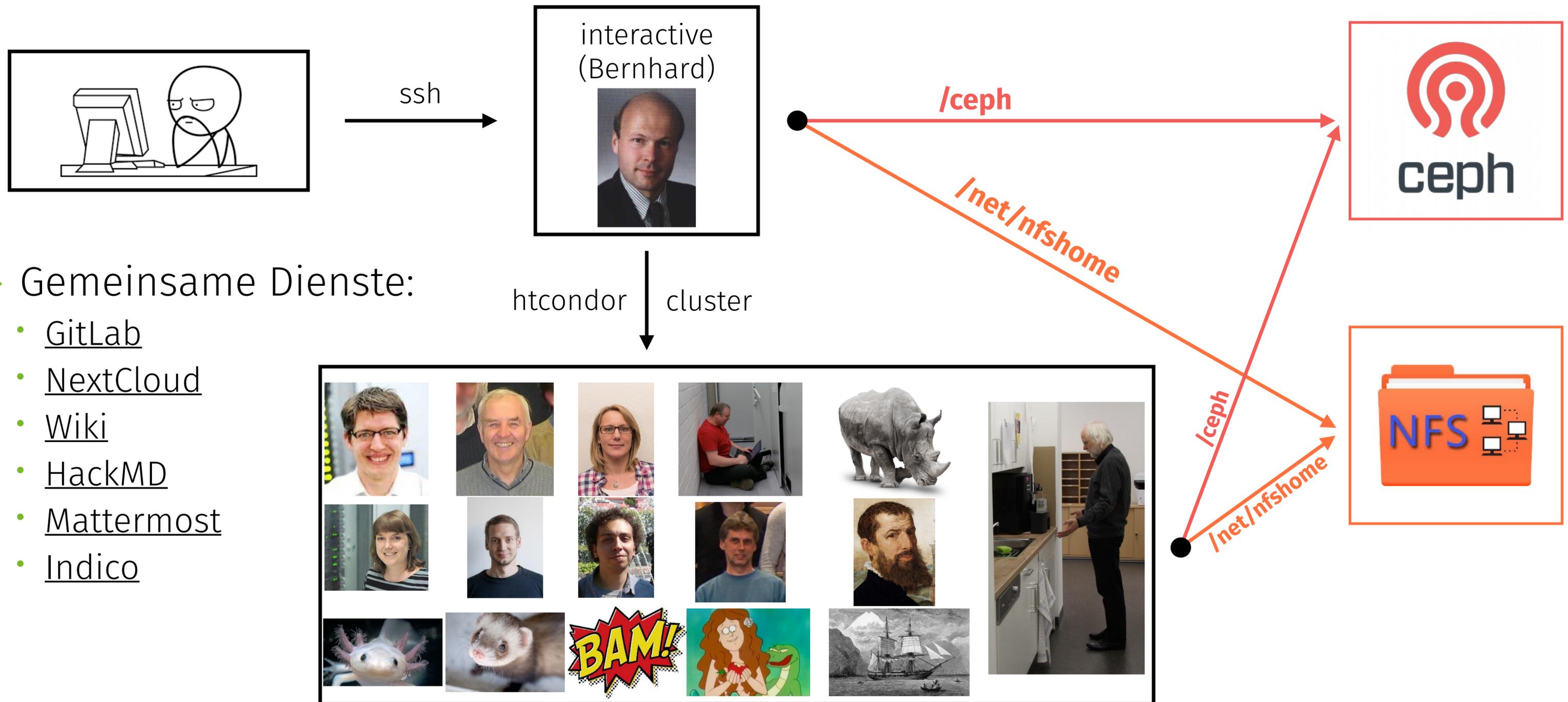
Mac

Setup & Installation Solutions



Einrichtung von Hardware und Software

Infrastruktur



► Gemeinsame Dienste:

- [GitLab](#)
- [NextCloud](#)
- [Wiki](#)
- [HackMD](#)
- [Mattermost](#)
- [Indico](#)

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 antje.moedden@tu-dortmund.de oder martin.bieker@tu-dortmund.de
- ▶ Die Mail sollte umfassen
 - Vorname, Nachname
 - TU Email-Adresse
 - Public ssh-keys ([Anleitung zur Erstellung eines ssh-keys](#))
- ▶ Accountname wird dann lauten
 - 1. Buchstabe des Vornamens + Nachname
Beispielsweise: jlanger

```

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
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.txt"), "w")
        self.file.seek(0)
        self.fingerprints.update(self.request_fingerprint(request))

    @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)

```

Vorbereitende Installationen

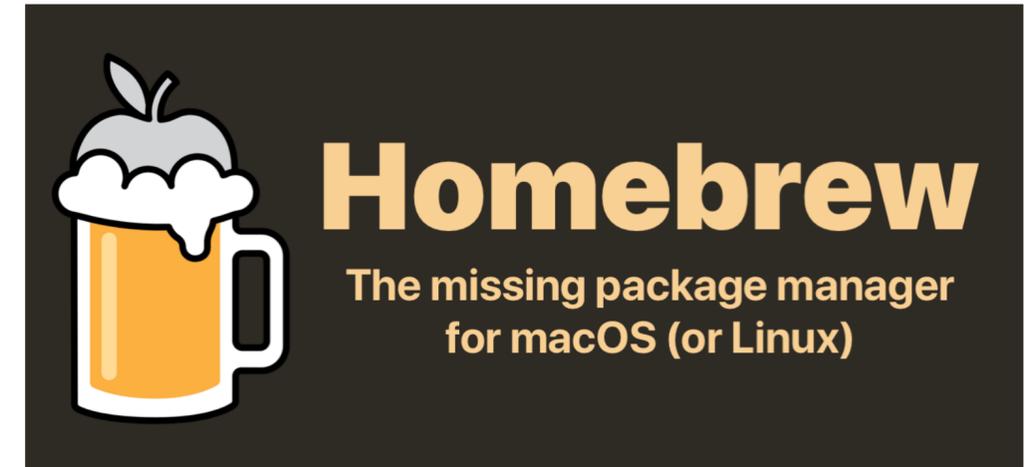
- ▶ Benötigt wird
 - Text Editor
 - Sublime-Text
 - Atom
 - VisualStudio
 - Git
 - Python
 - sshfs
 - Chat zur Kommunikation mit den Betreuern
 - Mattermost
 - Telegram-Desktop
- ▶ Auf den folgenden Slides findet ihr eine Installationsmöglichkeit für die benötigte Software mit Hilfe von Homebrew

Software Installation

- ▶ Paketverwaltungssoftware Homebrew
 - Vergleichbar mit apt oder yum
 - Eigene Ordner für Programme, im System nur verlinkt
 - Übersichtlich, einfach zu warten

- ▶ Homebrew installieren

```
/usr/bin/ruby -e "$(curl -fsSL  
https://raw.githubusercontent.com/  
Homebrew/install/master/install)"
```



- ▶ Für command-line Software:
`brew install software_name`
- ▶ Für graphische Software:
`brew cask install software_name`

Installieren mit Homebrew

Installieren mehrerer Programme:

- ▶ `brew bundle` installiert eine Liste von Programmen
- ▶ Brewfile anlegen oder vorhandenes ausführen (siehe rechts)
- ▶ `brew bundle` ausführen
`brew bundle /Path/To/Brewfile`
- ▶ `brew bundle dump` erstellt eine brewfile aus euren installierten Programmen

Installation auf MacOS und Linux unterscheidet sich nicht

- ▶ `brew bundle` installiert automatisch nur die Programme die auf eurem OS verfügbar sind
- ▶ Ein Brewfile mit dem nötigsten liegt [hier](#)
 - Durch einkommentieren installiert ihr zusätzliche Programme
- ▶ Installation graphischer Software auf diese Weise eventuell nicht möglich
 - Überprüft ob ihr alle benötigten Programme installiert habt
 - Im Notfall manuelle Installation