

# Sitzungsprotokoll: Technologie- & Workflow- Meeting, 03.03.2021

## Tagesordnung

1. Tagesordnung
2. Termine
3. Nutzung des HedgeDoc Pads
4. Web-Technologien
5. Datenrepräsentation (Hierarchie / Suchfunktion)
6. verfügbare Plattformen

## Termine

Nächstes Meeting: KW11, Mittwoch, 2021-03-17, 18:00

<https://fau.zoom.us/j/92803652912?pwd=VEp6NFAzbXhrRkpCMmNxbXFqK1J3dz09>

## Dieses Pad so weiterführen?

Github-Issuetracker (Anmerkung: Vortrag Julian am RRZE zu Git-Plattformen als "HPC Cafe") -> TODO phi

<https://hpc.fau.de/systems-services/support/hpc-cafe/>

Protokolle in die Wissenhalde ([wiki.betreiberverein.de](https://wiki.betreiberverein.de)) ablegen.

Bitte registrieren: <https://github.com/login>

GH Accounts: cod3monk, phi1010,

# Wer kann/will mit welchen Technologien arbeiten?

<https://terminklick.stuve.fau.de/poll/nPPKg3Uahz/>

-> Python Tendentiell: Django (wg. inkrementellen Upgrades bei Django-Datenmodell) mit Unit-Test-Framework via Github Actions

IDE: diverse (Pycharm, VS Code)

Frontend: Entscheidung offen lassen; Person suchen, dann entscheiden.

# Datenrepräsentation (Hierarchie / Suchfunktion)

-> Bibliothekare einladen

bspw. Eingabeprotokollierung Suchanfragen / Eintragung zwecks Verbesserung der Datenqualität

# Wollen wir bestehende Plattformen erweitern?

Bspw. Friendica

-> Klären, welche Kommunikationsplattform der Verein haben will.

# Welche Personengruppen schauen sich welche Themenbereiche genauer an, und stellen nächstes Mal Ergebnisse vor?

phi: Github / Django / Docker Julian: Frontendler und Bibliothekar:innen; evtl. Fragen zu Django  
Christian: Protkolle ins Wiki

## Was brauche ich für Entwicklung

- Linux
  - Installation: ggf. VM, od. WSL(2) auf Windows
  - Knowhow: Bash, Paketmanager der Wahl, Shebang
- Git
  - Installation: Betriebssystem-Paketmanager
  - Knowhow: Clone, Add, Commit, Push, Pull, Merge
    - <https://ohmygit.org/>
    - <https://www.katacoda.com/courses/git>
- Python3.8
  - Installation: Betriebssystem-Paketmanager
    - evtl. alternativ <https://pipenv.pypa.io/en/latest/advanced/#automatic-python-installation>
  - Knowhow: Syntax.
- Pip
  - Installation: Betriebssystempaketmanager
  - Knowhow: Wie finde ich Paketnamen und verlinkte Dokumentation
    - <https://pypi.org/project/Django/>
- Pipenv
  - Installation: `pip install pipenv`, oder aus Betriebssystempaketmanager
  - Knowhow: Befehle install/run/shell

- <https://pipenv.pypa.io/en/latest/#other-commands>
- <https://pipenv.pypa.io/en/latest/basics/#example-pipenv-workflow>
- Virtualenv / Venv
  - Installation: Automatisch mit Pipenv
  - Knowhow: Unterschied Virtualenv -- System-Python
    - <https://docs.python.org/3/tutorial/venv.html#introduction>
- Docker
  - Installation: Betriebssystempaketmanager
  - Knowhow:
    - Ausführen: docker build, docker run
    - Debuggen / Stoppen: docker exec, docker ps, docker kill
    - Aufräumen: docker container prune, docker image prune
    - Konfiguration: Dockerfile Syntax
    - <https://docs.docker.com/get-started/overview/>
- Docker-Compose:
  - Installation: Automatisch mit Docker
  - Knowhow:
    - Ausführen: docker-compose build, docker-compose up
    - Konfigurieren: docker-compose.yml-Syntax
    - <https://docs.docker.com/compose/gettingstarted/>
- Django:
  - Installation: `pipenv install` in dem Verzeichnis ausführen, in dem die Pipfile liegt
  - Knowhow:
    - Datenbank initialisieren: manage.py migrate
      - <https://docs.djangoproject.com/en/3.1/ref/django-admin/#migrate>
    - Admininterface: manage.py createsuperuser
      - <https://docs.djangoproject.com/en/3.1/ref/django-admin/#createsuperuser>
    - Ausführen: manage.py runserver
      - <https://docs.djangoproject.com/en/3.1/ref/django-admin/#runserver>

---

Revision #2

Created 6 April 2021 16:31:23 by Christian Borss

Updated 12 November 2021 12:55:05 by Christian Borss