

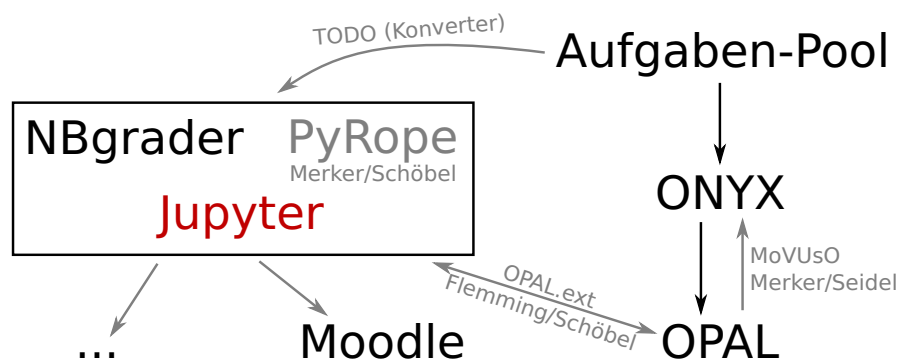
Jupyter's Friends

Ausfallsichere digitale Lehrmaterialien mit freien Tools und Infrastrukturen

Jens Flemming, Westsächsische Hochschule Zwickau, jens.flemming@fh-zwickau.de

zum Mitklicken:

www.fh-zwickau.de/~jef19jdw > Teaching > Data Science I > Bild klicken



Jupyter Notebooks

- Text, Bilder, Code, Videos, LaTeX,... in einem Dokument
- Code an Ort und Stelle ausführ- und veränderbar
- läuft im Webbrowser
- lokal oder remote
- User-Interfaces: Jupyter Notebook (alt) und JupyterLab (neu)
- Open Source, riesige Community

[Demo JupyterLab](#)

Worum geht's?

- nicht: Jupyter Notebooks
- Tools drumrum
- Ausfallsicherheit

Jupyter

- Notebooks in Texteditor erstellen (analog zu LaTeX für PDF)
- eine Quelle für Notebook, HTML, PDF,...
- paralleles Bearbeiten als Textdatei und Notebook
- Textdateien leichter versionierbar

Python Modules

Next to built-in functions like `print` and `input` Python ships with several modules, which can be used on demand. A module is a collection of additional functionality. Everybody can write and publish Python modules. How to do this will be explained later on. Modules either are written in Python or in C programming language.

Before we can use functionality of a module we have to import it:

```
'''(code-cell) ipython3
import numpy as np
'''
```

'''(hint)

A number of modules comes pre-installed with Python (the [Python standard library](#)). But others have to be installed separately. Whenever Python shows `ModuleNotFoundError` you forgot to install the module you want to import. Install a module via Anaconda Navigator or via `conda install module_name` in a terminal, see [\[\]\(projects:\)](#)

The code above imports the module `'numpy'` and makes it accessible under the name `np`. NumPy is a collection of functions and data types for advanced numerical computing. We will dive deeper into this module later on. To use NumPy's functionality we have to write `np.some_function` with `'some_function'` replaced by one of NumPy's functions.

```
'''(code-cell) ipython3
np.sin(0.25 * np.pi)
'''
```

Python Modules

Next to built-in functions like `print` and `input` Python ships with several modules, which can be used on demand. A module is a collection of additional functionality. Everybody can write and publish Python modules. How to do this will be explained later on. Modules either are written in Python or in C programming language.

Before we can use functionality of a module we have to import it:

```
[1]: import numpy as np
```



Hint

A number of modules comes pre-installed with Python (the [Python standard library](#)). But others have to be installed separately. Whenever Python shows `ModuleNotFoundError` you forgot to install the module you want to import. Install a module via Anaconda Navigator or via `conda install module_name` in a terminal, see [\[\]\(projects:\)](#)

The code above imports the module `'numpy'` and makes it accessible under the name `np`. NumPy is a collection of functions and data types for advanced numerical computing. We will dive deeper into this module later on. To use NumPy's functionality we have to write `np.some_function` with `'some_function'` replaced by one of NumPy's functions.

```
[4]: np.sin(0.25 * np.pi)
```

```
[4]: 0.7071067811865476
```

Python Modules

Next to built-in functions like `print` and `input` Python ships with several modules, which can be used on demand. A module is a collection of additional functionality. Everybody can write and publish Python modules. How to do this will be explained later on. Modules either are written in Python or in C programming language.

Before we can use functionality of a module we have to import it:

```
import numpy as np
```



Hint

A number of modules comes pre-installed with Python (the [Python standard library](#)). But others have to be installed separately. Whenever Python shows `ModuleNotFoundError` you forgot to install the module you want to import. Install a module via Anaconda Navigator or via `conda install module_name` in a terminal, see [\[\]\(projects:\)](#)

The code above imports the module `'numpy'` and makes it accessible under the name `np`. NumPy is a collection of functions and data types for advanced numerical computing. We will dive deeper into this module later on. To use NumPy's functionality we have to write `np.some_function` with `'some_function'` replaced by one of NumPy's functions.

```
np.sin(0.25 * np.pi)
```

Git und GitLab

- Versionieren von Textdateien
- Veröffentlichen und Teilen
- gemeinsames Arbeiten an Material
- steile Lernkurve, aber extrem wertvoll

Demo...

Jupyter Book

- fasst einzelne Notebooks zu "Buch" zusammen
- generiert Inhaltsverzeichnis, Index, Querverweise, Literaturlisten,... (ähnlich zu LaTeX)
- Export als statische Website, Jupyter Notebooks, LaTeX, PDF,...
- exzellente Integration mit GitLab und anderen Jupyter-Tools

Demo...

JupyterHub

- Server-Software, die mehreren Nutzern jeweils ein JupyterLab zur Verfügung stellt
- keine Installationsarbeiten auf Nutzerseite (nur Webbrowser)

NBgitpuller

- kopiert Notebooks aus GitLab in JupyterLab des Nutzers
- aktualisiert Notebooks bei jedem Start
- nimmt Rücksicht auf Bearbeitungen des Nutzers
- völlig transparent aus Nutzersicht

[Demo...](#)

Binder

- öffentlicher JupyterHub
- nahtlose Integration in Jupyter Book und Co. (außer NBgrader)

Thebe

- Quellcode innerhalb Website bearbeiten und ausführen
- basiert auf Binder

[Demo...](#)

Stabilität

JupyterHub tot?

anderen oder mybinder.org nehmen; lokal arbeiten; ggf. Thebe

GitLab tot?

anderes nehmen, z.B. gitlab.com, github.com

Webspace tot?

anderen nehmen (statische Website braucht nichts spezielles) oder offline arbeiten

NBgrader auf Server tot?

lokal installieren und Notebooks per E-Mail verteilen/einsammeln

Shibboleth-Login tot?

Website läuft, Material-Updates und Verteilung via GitLab läuft, mybinder.org läuft

Wesentliche Vorteile

- Browser-basiert (durchsuchbar, interaktiv, barrierearm, Übersetzungs-Tools,...)
- zentrale Anlaufstelle für Studierende (Website)
- kein Online-Zwang (PDF oder Offline-HTML)
- echte OER (keine "Login-Wall")
- gemeinsames Nutzen/Bearbeiten einfach
- keine Lizenzkosten, alles frei und Open Source
- riesige Community, aktive Weiterentwicklung

Fehlt da noch was?

- Musterlösungen (nicht) mit veröffentlichen (unlösbar?)
- **hochschulübergreifender JupyterHub** (Mitstreiter?)

Das war's

angucken und ausprobieren:

1. www.fh-zwickau.de/~jef19jdw
2. > Teaching
3. > Data Science I
4. > Bild klicken

Infos zum Setup im GitLab (und bei mir)