

# Einführung: Digital Humanities Virtual Laboratory

## Inhaltsverzeichnis

<b>1</b>	<b>Unterschiede zwischen DHVlab und Moodle</b>	<b>1</b>
<b>2</b>	<b>Komponenten des DHVlab</b>	<b>2</b>
2.1	Virtueller Desktop	3
2.2	Datenrepositorium (= MySQL Datenbank-Client)	5
2.3	Cloud	7
2.4	Live-Code	8
2.5	GitLab	9
2.6	Squirrel	11
<b>3</b>	<b>Labor- &amp; Benutzerverwaltung</b>	<b>12</b>

## 1 Unterschiede zwischen DHVlab und Moodle

**DHVLab** Das DHVlab unterscheidet sich grundlegend von Organisationsplattformen wie Moodle oder Olat, welche in der digitalen Lehre gängigerweise eingesetzt werden. Zwar können auch in DHVlab sogenannte Labore eingerichtet werden, ähnlich den Kursen bei Moodle, die Funktionen beschränken sich aber nicht auf die Durchführung von Kursen mit allen benötigten Arbeitsmaterialien und Lernaktivitäten. Das DHVlab bietet für diese Labore eine vollumfängliche zentrale Infrastruktur (aus mehreren quelloffenen Softwarekomponenten) zur Realisierung von Projekten oder zur interaktiven Administration von Seminarinhalten.

Im DHVlab können Kursleiter und Teilnehmer über einen Account sowohl auf einen Virtuellen Desktop mit verschiedenen Anwendungsprogrammen, SQL-Datenbanken, Clouddienste, Veröffentlichungsplattformen, GIT-Repositorien, Jupyter-Applikationen und weitere Analysewerkzeuge für Daten zugreifen. Das DHVlab eignet sich damit ideal für:

- Seminare mit anspruchsvollen digitalen Inhalten wie Korpora, Datenbanken, XML, OCR.
- Einführungsseminare im Bereich Statistik (R), SQL/XML, Python.

- Forschungsprojekte (Digital Humanities - Bereich), welche zentrale Datenbank-, Repositiums- und Entwicklungsumgebungen benötigen.

Das DHVlab eignet sich weniger für:

- Einfache Kursverwaltung, Planung und Bereitstellung von Kursmaterialien wie Moodle oder OLAT dies vorsieht.
- Kursleiter ohne Grundkenntnisse im Bereich SQL, XML R oder Python.

## 2 Komponenten des DHVlab

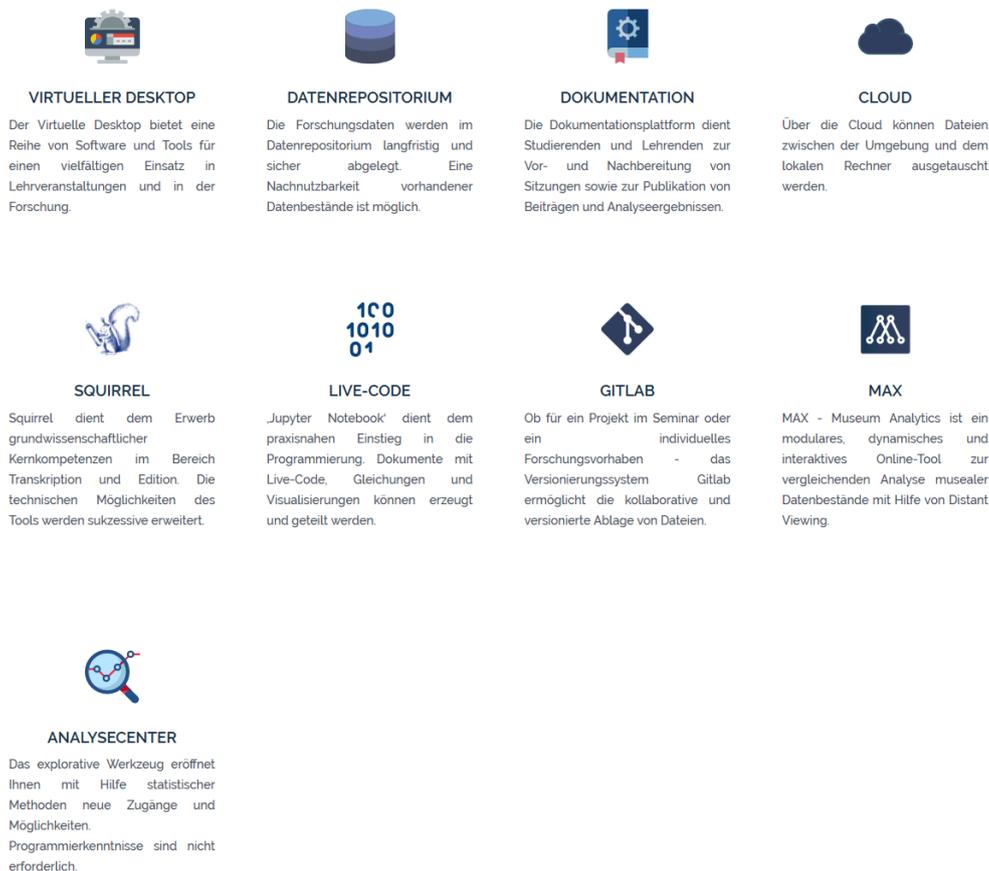


Abbildung 1: Hier sehen Sie alle aktuell (Juli 2019) im DHVlab enthaltenen Komponenten. Diese können unabhängig von einander genutzt werden, die Daten einiger Module können jedoch innerhalb eines Labors synchronisiert werden.

## 2.1 Virtueller Desktop

Über den virtuellen Desktop steht Ihnen eine voll funktionsfähige LINUX-Desktopumgebung in Ihrem Webbrowser zur Verfügung, die von den Servern der ITG direkt auf Ihren lokalen Rechner gestreamt wird.

Wählen Sie als Verbindung für die Verwendung des virtuellen Desktops **Téchné VCN 1366 x 768**.

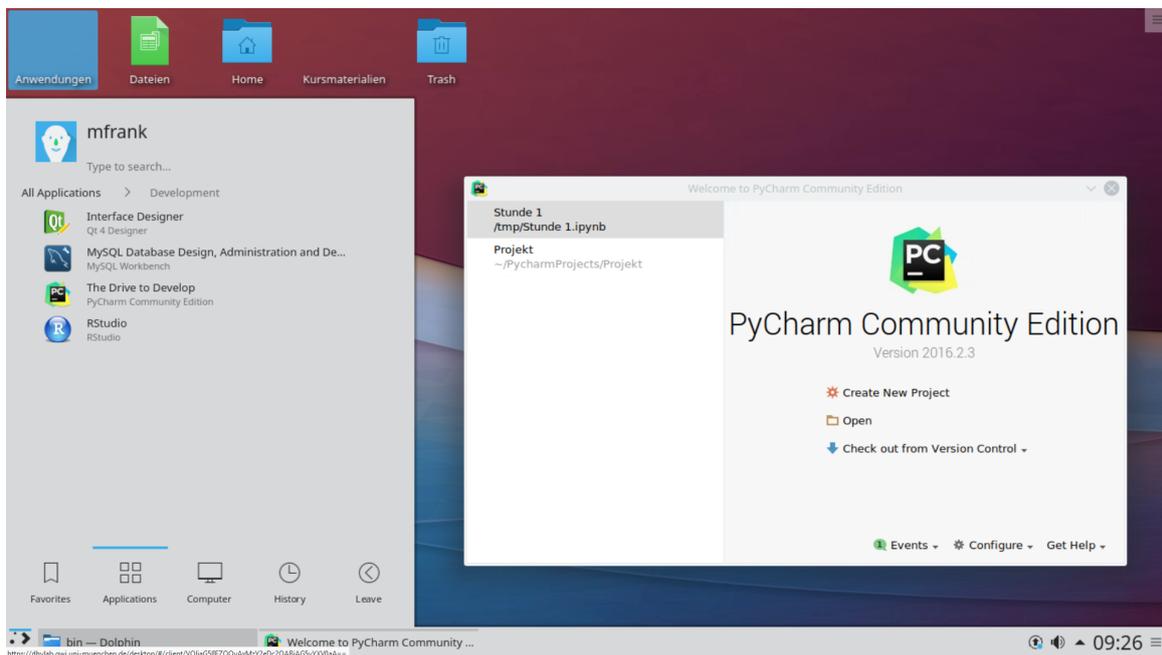


Abbildung 2: Ansicht des virtuellen Desktop mit geöffnetem PyCharm (IDE für Python).

Vorteile gegenüber der dezentralen Nutzung einzelner Seminarrechner:

- Benötigte Software muss nicht lokal installiert werden sondern steht zentral zur Verfügung.
- Kursleiter können die Dateien aller Kursteilnehmer einsehen, verändern, oder Dateien problemlos an alle verteilen.
- Für aufwändige Anwendungen kann die Rechenpower der Servers (leistungsvolle XEON CPUs und Nvidia QuadroPlex GPUs) genutzt werden (z.B. für aufwändige statistische Analysen).

Nachteile der Technik:

- Gute Internetverbindung mit niedriger Latenz erforderlich.
- Software kann nicht selbständig installiert werden, eine Installation muss bei der ITG beantragt werden (und nur quelloffene Software möglich).
- Trotz guter Internetverbindung bleibt eine merkliche Verzögerung der Eingabe.
- Ein umwegloser Austausch zwischen Desktop-Dateien und virtuellem Desktop ist nicht möglich.

Folgende Programme sind im virtuellen Desktop vorinstalliert:

- PyCharm als IDE für Python 3
- RStudio als IDE für R
- MySQL Workbench als IDE für MySQL
- Webbrowser: Firefox, Chrome
- LibreOffice (Writer, Calc, Impress, Draw, Math, Base)
- Vim Texteditor
- Oxygen XML-Editor und XMLSpear-Editor
- AntConc, Praat, Stanford-POSTagger, Tree-Tagger

Einsatzszenarien:

- Lehre: Vorstellung verschiedener Anwendungsprogramme für den wissenschaftlichen Einsatz. Inspektion und Verteilung von Seminarunterlagen an die Studierenden.
- Projekte: Bereitstellung einer vollausgestatteten Arbeitsplattform für alle Beteiligten eines Projektes.

## 2.2 Datenrepositorium (= MySQL Datenbank-Client)

Beim Datenrepositorium handelt es sich um eine MySQL-Datenbank, die über das clientseitige Frontend phpMyAdmin gesteuert wird.

Kursleiter erhalten pro angelegtem Kurs eine lab-Datenbank mit beliebig vielen anlegbaren Relationen. Jeder Kursteilnehmer erhält eine Datenbank mit beliebig vielen anlegbaren Relationen. Kursleiter können alle Teilnehmerdatenbanken einsehen und beliebig verändern (CREATE, DELETE, COPY, UPDATE etc.).

Zur Steuerung des Datenrepositoriums wird die SQL-Abfragesprache für relationale Datenstrukturen in der dialektalen Form MySQL verwendet.

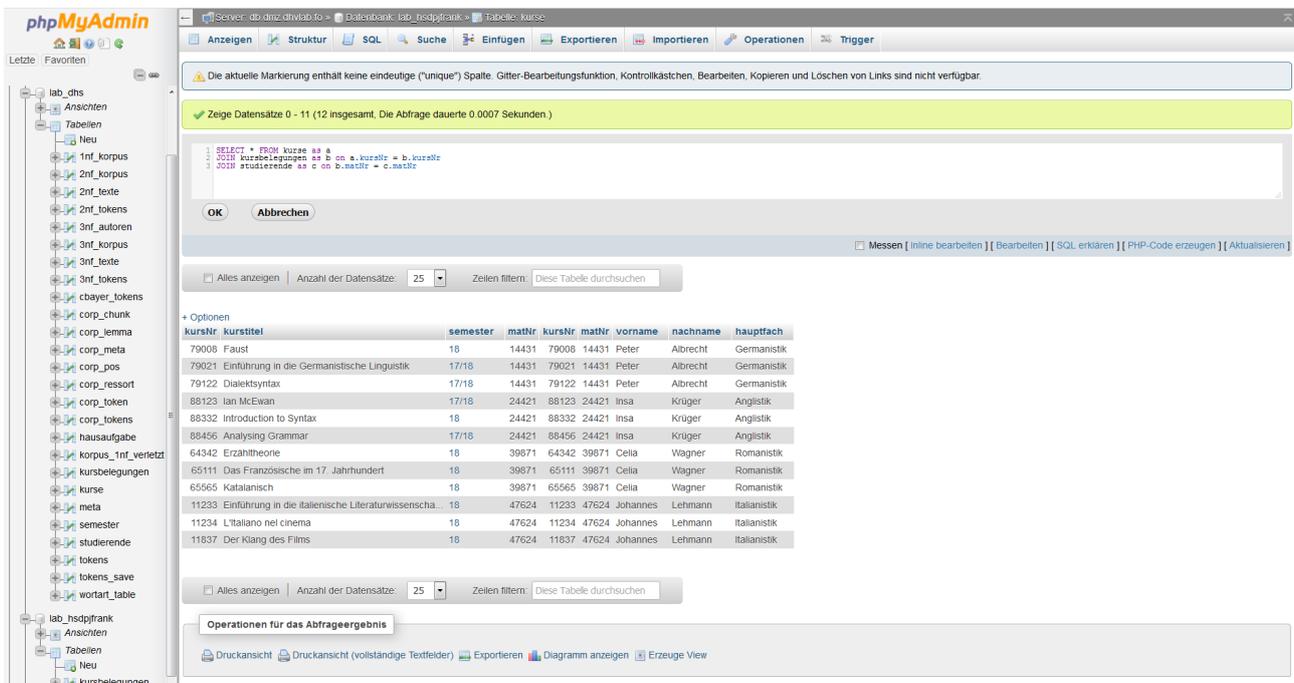


Abbildung 3: Ansicht der phpMyAdmin-Oberfläche zur Steuerung von MySQL-Datenbanken.

Beispiele für MySQL-Abfragen:

1 — Einfaches SQL-Statement

```
2 SELECT matNr, vorname, nachname
3 FROM 'studierende'
4 WHERE hauptfach LIKE 'Germanistik'
5 ORDER BY nachname DESC;
```

7 — SQL-Statement mit JOIN-Syntax über Daten in der 2. Normalform

```
8 SELECT a.kursNr,  
9        a.kurstitel,  
10       CONCAT(a.semester, " ", d.styp) as semester,  
11       CONCAT(c.vorname, " ", c.nachname) as name,  
12       c.hauptfach  
13 FROM kurse as a  
14 JOIN kursbelegungen as b on a.kursNr = b.kursNr  
15     JOIN studierende as c on b.matNr = c.matNr  
16     JOIN semester as d on a.semester = d.semester  
17 ORDER by c.hauptfach  
18  
19 — SQL-UPDATE-Statement  
20 UPDATE studierende  
21     SET nachname = "Becker"  
22     WHERE vorname LIKE "Johannes"
```

Zur Einführung in die Abfragesprache MySQL (und XML):

- Seminar „Grundlagen der Datenmodellierung“ (immer SS im DH-S Studiengang).
- Beaulieu, Alan (2009): Einführung in SQL. O'Reilly.

Einsatzszenarien:

- Lehre: Seminare, welche MySQL zum Gegenstand haben oder solche, in denen relational strukturierte Daten eine Arbeitsgrundlage darstellen.
- Projekte: Alle Projekte, in denen relationale Datenbanken eine wichtige Rolle spielen. Sollten aufwändigere Projekte realisiert werden, können bei der ITG eigene Datenbankserver ohne die Restriktionen des DHVlab angefragt werden, welche getrennt vom DHVlab arbeiten.

Ausblick: Alternativ zu den relationalen Datenbanken soll an dieser Stelle auch Zugang zu Graph-Datenbanken gewährt werden. Hierbei wird es sich voraussichtlich um neo4j handeln, welches mit der Abfragesprache Cypher gesteuert wird.

## 2.3 Cloud

*Private Clouddienste wie Dropbox unterliegen nicht dem deutschen Datenschutzrecht. Ihre Nutzung ist im dienstlichen Umfeld in der Regel unzulässig.*

Die DHVlab Cloud basiert auf OWNCloud und wird allen registrierten Nutzern zur Verfügung gestellt, um Daten zu speichern und auszutauschen. Eine Alternative ist das vom LRZ angebotene Sync+Share (50 GB).

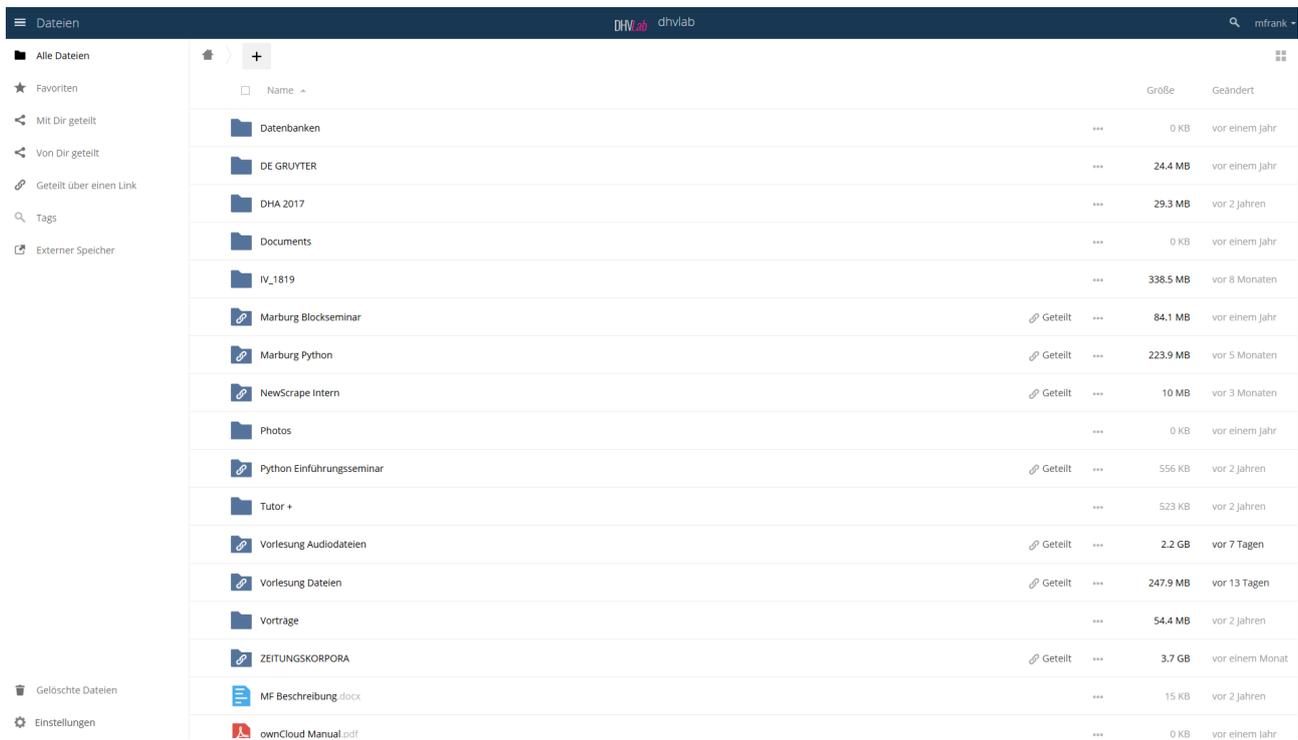


Abbildung 4: Ansicht der DHVlab Cloud mit Dateisystem.

Die Cloud bietet folgende Funktionen:

- **Dateirepositorium** (20 GB, auf Anfrage aber auch mehr): Speichern, Laden und Teilen von Dateien über den Cloud-Dienst. Geteilte Daten können über Zeitstempel und Passwort geschützt werden. Auch ein Datenaustausch zwischen Desktop und virtuellem Desktop ist möglich.
- **Kontakte-Datenbank**: Datenbank mit der Möglichkeit zur Speicherung von Kontakten und Kontaktdaten zum Beispiel für Projektpartner.
- **Kalender**: Kalender mit der Möglichkeit, Veranstaltungen einzutragen.

- **Aufgaben:** Einfaches Tool zur Verwaltung von Dienstaufgaben (idR. auf Projekte bezogen) oder für den einfachen Einsatz im Projektmanagement. Eine komplexe Lösung hierfür wäre z.B. *Factro*.

Einsatzszenarien:

- Lehre: Verteilung größerer Seminar-Dateien über die Cloud.
- Projekte: Erstellung eines zentralen Projektrepositoriums für gemeinsame Dateien, Projektkontakte, Projekttermine und Aufgaben im Projekt (also Nutzung als Projektmanagement-Tool)

## 2.4 Live-Code

Live-Code stellt die Implementierung von Project Jupyter im DHVlab dar. Laborbenutzer können eigene interaktive und über Jupyter Kernel ausführbare Notebook-Dokumente (\*.pynb) erstellen, welche grundlegend aus zwei Komponenten bestehen:

- **Fließtexte**, geschrieben in der leicht zu erlernenden Markupsprache *Markdown*, wie sie auch von GitHub, Reddit oder OpenStreetMap verwendet wird.
- **Codeblöcke** mit ausführbarem Code in Python 3, R oder Julia, welche in die umgebenden Markdownsegmente eingebettet werden können.

Jupyter erlaubt dabei jedem Nutzer die Verwaltung eines individuellen Verzeichnisses im Labor (Anlegen von Ordnerstrukturen, Dateien etc.) für seine konkreten Arbeitsvorhaben.

Einsatzszenarien:

- Lehre: Erstellung dynamischer, interaktiver Seminarskripte mit wechselnden Markdown- und Code-Blöcken.
- Projekte: Erstellung komplexer Python- und R-Skripte, welche eine ausführliche Dokumentation benötigen. Dank ausführbarer Markup-Dokumente kann die Dokumentation optimal in den Programmcode integriert werden. Dies erleichtert die Kooperation mit Projektpartnern, die nur über geringe Programmierkenntnisse verfügen.

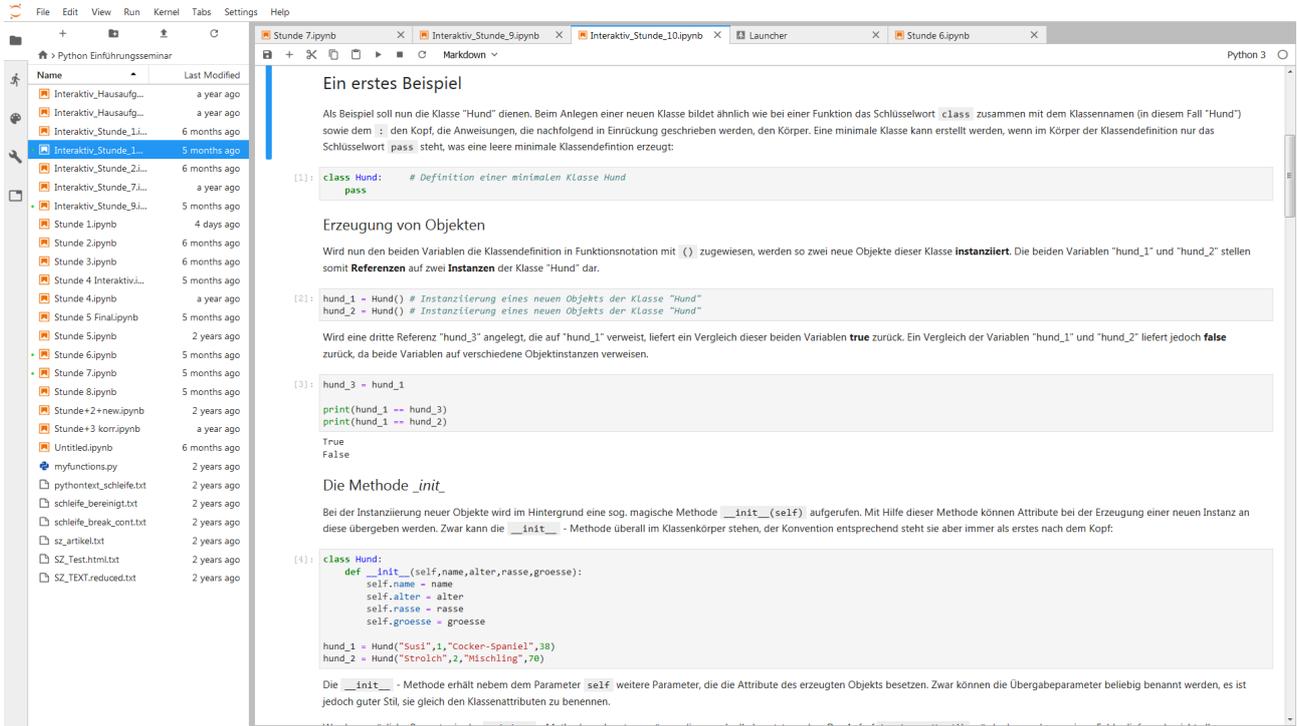


Abbildung 5: Beispiel eines geöffneten Jupyter Notebooks mit Markdown- und Code-Elementen.

**Achtung:** Bei der Verwendung der Notebooks sollte das Kernel-Verhalten von Python 3 beachtet werden! Der Kernel führt nur Zellen aus, zu deren Ausführung er explizit angewiesen wird. Variablen bleiben erhalten, bis der Kernel neu gestartet wird. In einigen Situationen kann es notwendig sein, den Kernel für das jeweilige Notebook anzuhalten oder neu zu starten, um etwaige Probleme mit Variablen oder auch Schleifen (Endlosschleifen) zu beheben. Dies kann über den Reiter Kernel in der oberen Menüleiste vorgenommen werden.

## 2.5 GitLab

Das DHVlab verfügt auch über eine eigene GitLab-Instanz. GitLab stellt eine Webanwendung zur Versionsverwaltung für Softwareprojekte auf Basis von Git dar. Als solche bietet GitLab diverse Managementfunktionen für Softwareprojekte aber auch andere größere Projekte, an denen mehrere Personen kollaborativ arbeiten.

Im Kern bietet Gitlab die Möglichkeit, über das Internet lokale Datei- und Ordnerstrukturen

mit einem zentralen Repository (der GitLab-Instanz im DHVlab) zu koppeln. Nutzern bieten sich dabei unter anderem folgende Möglichkeiten:

- Klonen aller oder ausgewählter Dateien im GitLab-Projekt auf den lokalen Rechner.
- Veränderung lokaler Daten mit anschließendem Upload in das zentrale GitLab-Projekt-repositorium

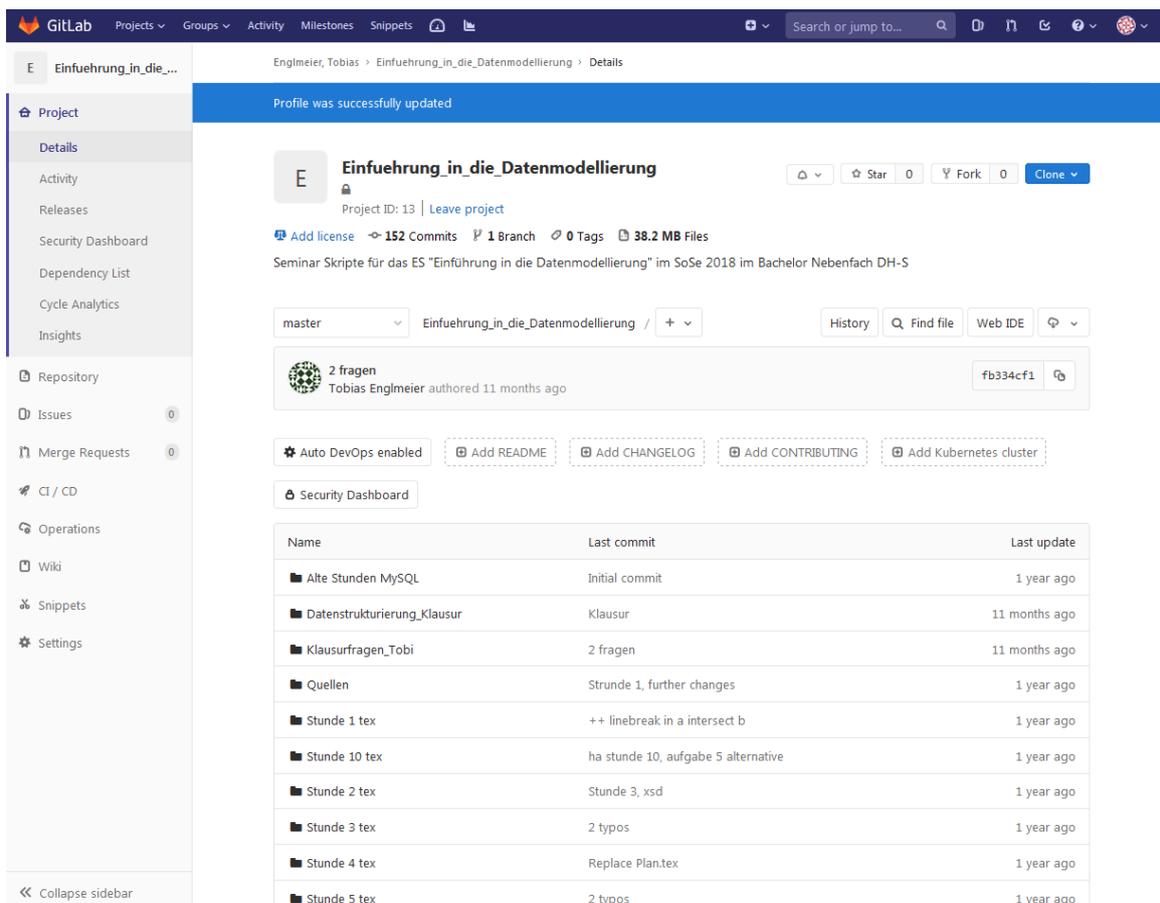


Abbildung 6: Beispiel eines über GitLab synchronisierten Seminarskriptes aus dem DH-S Nebenfach.

GitLab wird folglich dazu benutzt, alle lokalen und zentralen Daten auf einem synchronen Stand zu halten. Alle Beteiligten können jede Veränderung im Dateibestand, aber auch in einzelnen Dateien, nachvollziehen, übernehmen, aber auch rückgängig machen. Außerdem bietet GitLab die Möglichkeit, mehrere Versionen eines gemeinsamen Projektes anzulegen und diese auch beispielsweise parallel zu entwickeln oder daran zu experimentieren (über sog. forks). Auf diese Weise kann eine konsolidierte Hauptversion der Projektdaten erstellt werden, während

verschiedene Mitarbeiter oder Gruppen von Mitarbeitern an unterschiedlichen Varianten zu dieser Hauptversion weiterarbeiten.

Einsatzszenarien:

- Lehre: Arbeit mit mehreren Studierenden an einem gemeinsamen Satz von Seminardateien. Gemeinsame Arbeit mehrerer Dozenten an einem Seminar bzw. einer Vorlesung. Sehr geeignet für Textdateien, weniger als Bildrepositorium.
- Projekte: Synchronisation zentraler Projektdaten (v.a. Textdateien) und gemeinsame Arbeit aller Projektmitarbeiter an diesen Daten. Wird Forschungssoftware entwickelt, wird die Verwendung eines GitLab-Repositoriums zur Versionskontrolle dringend empfohlen, vor allem wenn mehrere Mitarbeiter an technischen Komponenten arbeiten. Arbeitet nur eine Person an den IT-Komponenten des Projektes, kann die Verwendung von GitLab zur Versionierung dennoch empfohlen werden.

## 2.6 Squirrel

Bei Squirrel handelt es sich um ein Tool zum Erwerb grundwissenschaftlicher Kompetenzen im Bereich der Edition und Transkription.

Nach dem Login in das Tool sehen Sie alle Dokumente im Bestand inklusive Metadaten, welche Sie auch durchsuchen können. Wollen Sie weitere Dokumente hinzufügen, drücken Sie F9 und Sie erhalten ein Menü zum Upload von Bilddateien für die spätere Transkription und Annotation. Laden Sie die entsprechenden Bilder hoch, geben Sie verfügbare Metainformationen an und drücken Sie am Ende auf das [+] -Symbol in der oberen Leiste neben dem Anmeldung-Link. Die Datei ist kann nun von Ihnen bearbeitet werden.

Schreiben Sie auf der rechten Seite Ihren Transkriptionstext, und weisen Sie diesem anschließend die entsprechenden Bereiche im Bilddokument zu. Markieren sie hierzu zunächst das Wort im Transkriptionstext, und rahmen Sie anschließend bei gedrückter Alt-Taste die entsprechende Position im Bild. Auch verschiedene Annotationen können im Text über ein Auswahlmü am unteren rechten Rand des Tools angebracht werden (Personen, Orte, Zeitpunkte etc.). Zum Speichern Ihres aktuellen Bearbeitungsstandes drücken sie auf das Häkchen in der oberen Menüleiste.

Eine genauere Anleitung zur Benutzung von Squirrel finden Sie im DHVlab-Wiki unter [https://dhvlab.gwi.uni-muenchen.de/wiki/4.lv\\_edit](https://dhvlab.gwi.uni-muenchen.de/wiki/4.lv_edit).

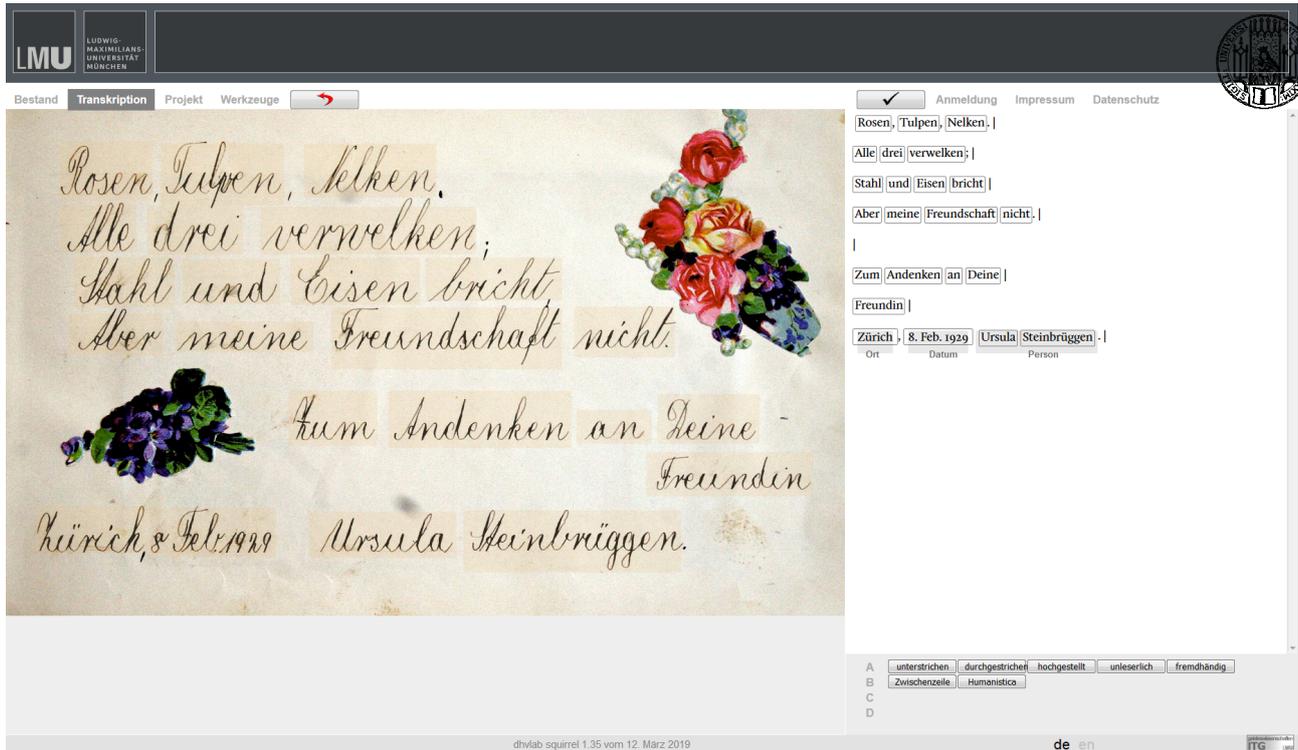


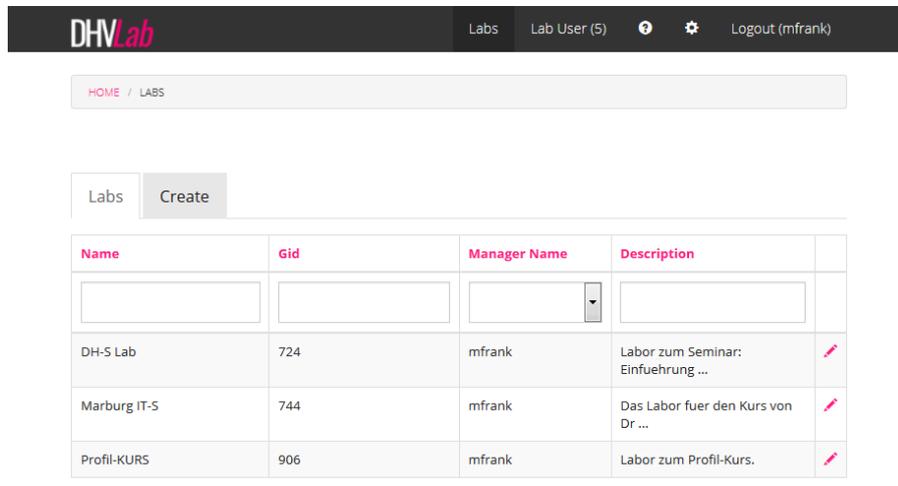
Abbildung 7: Das Editorfenster von Squirrel mit einem beispielhaft getagten Spruch aus einem Poesiealbum.

### 3 Labor- & Benutzerverwaltung

Um das DHVlab in Ihrem Kurs einsetzen zu können, wenden Sie sich bitte über das entsprechende Formular auf der DHVlab-Homepage an das Projektteam. Das Projektteam wird Ihnen dann einen entsprechenden Account zur Anlegung und Verwaltung von Laboren einrichten.

Nach dem Login sind Sie nun in der Lage, ein Labor einzurichten. Sie können diesem einen Namen und eine Beschreibung geben. Nutzer können sich daraufhin im DHVlab für das von Ihnen eingerichtete Labor anmelden.

Für jeden angemeldeten Benutzer erhalten Sie eine Email, um ihn im DHVlab freizuschalten. Alternativ können Sie über das LAB USER-Menü neue Benutzer auch manuell in Ihre Kurse eintragen, ihnen Zugriff gewähren und ihre Registrierungsdaten ändern.

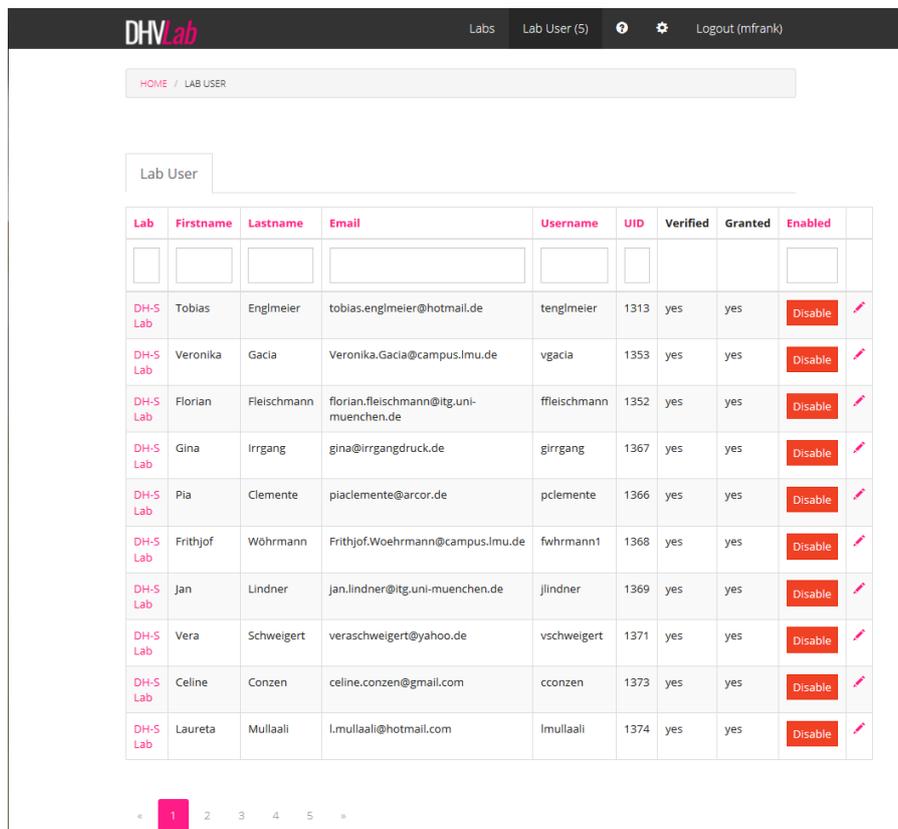


HOME / LABS

Labs **Create**

Name	Gid	Manager Name	Description
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
DH-S Lab	724	mfrank	Labor zum Seminar: Einfuehrung ...
Marburg IT-S	744	mfrank	Das Labor fuer den Kurs von Dr ...
Profil-KURS	906	mfrank	Labor zum Profil-Kurs.

Abbildung 8: Seite zur Verwaltung der Labore.



HOME / LAB USER

Lab User

Lab	Firstname	Lastname	Email	Username	UID	Verified	Granted	Enabled
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
DH-S Lab	Tobias	Englmeier	tobias.englmeier@hotmail.de	tenglmeier	1313	yes	yes	Disable
DH-S Lab	Veronika	Gacia	Veronika.Gacia@campus.lmu.de	vgacia	1353	yes	yes	Disable
DH-S Lab	Florian	Fleischmann	florian.fleischmann@itg.uni-muenchen.de	ffleischmann	1352	yes	yes	Disable
DH-S Lab	Gina	Irrgang	gina@irrgangdruck.de	girrgang	1367	yes	yes	Disable
DH-S Lab	Pia	Clemente	piaclemente@arcor.de	pclemente	1366	yes	yes	Disable
DH-S Lab	Frithjof	Wohrmann	Frithjof.Woehrmann@campus.lmu.de	fwormann1	1368	yes	yes	Disable
DH-S Lab	Jan	Lindner	jan.lindner@itg.uni-muenchen.de	jlindner	1369	yes	yes	Disable
DH-S Lab	Vera	Schweigert	veraschweigert@yahoo.de	vschweigert	1371	yes	yes	Disable
DH-S Lab	Celine	Conzen	celine.conzen@gmail.com	cconzen	1373	yes	yes	Disable
DH-S Lab	Laureta	Mullaali	lmullaali@hotmail.com	lmullaali	1374	yes	yes	Disable

« 1 2 3 4 5 »

Abbildung 9: Seite zur Verwaltung der Benutzer in den von Ihnen angelegten Laboren.