Jupyter Notebooks nutzen

Information Processing and Analytics

Mai 2026

Dieses Dokument zeigt verschiedenen Möglichkeiten auf, wie mit Jupyter Notebooks gearbeitet werden kann.

1 Jupyter Notebooks ausführen und bearbeiten

Wir empfehlen die folgenden Varianten zum Ausführen und Bearbeiten von Jupyter Notebooks:

1. Eine Installation von Anaconda auf Ihrem eigenen Rechner (lokal),[1] welche Python, Jupyter sowie eine vollständige Entwicklungsumgebung bereitstellt.
2. Die Nutzung des JupyterHubs der Humboldt-Universität zu Berlin.
3. Die Nutzung von Google Colab.

In Tabelle 1 werden diese Varianten in einer Übersicht verglichen und im folgenden im Detail vorgestellt.

Tabelle 1: Vergleich der verschiedenen Varianten zur Nutzung von Jupyter.

	Lokal	JupyterHub	Google Colab
Keine Internetverbindung erforderlich	✓	✗	✗
Keine Installation notwendig	✗	✓	✓
Kein externer Account erforderlich	✓	✓	✗
Zugriff auf Uni-Infrastruktur	✗	✓	✗
Unabhängigkeit der Uni-Infrastruktur	✗	✗	✓
Volle Kontrolle über die Arbeitsumgebung	✓	✗	✗

[1] Üblicherweise ist Anaconda auch auf den Rechnern in den Räumen 117 und 118 installiert.

1.1 Lokal: Jupyter-Installation mit Anaconda

Installation Die Installation erfolgt über den Anaconda-Installationslink für das jeweilige Betriebssystem. Wir empfehlen, stets die aktuellste Version zu verwenden.

Nach der Installation stehen Ihnen der Anaconda Navigator, Python sowie eine vollständige Entwicklungsumgebung zur Verfügung. Über den Navigator können verschiedene Anwendungen gestartet werden, unter anderem auch die lokale Jupyter-Installation.

Notebooks öffnen, bearbeiten und speichern Öffnen Sie zunächst den Anaconda Navigator und starten Sie dort *JupyterLab*, wie in Abbildung 1 dargestellt. JupyterLab öffnet sich anschließend automatisch in Ihrem Standard-Webbrowser.

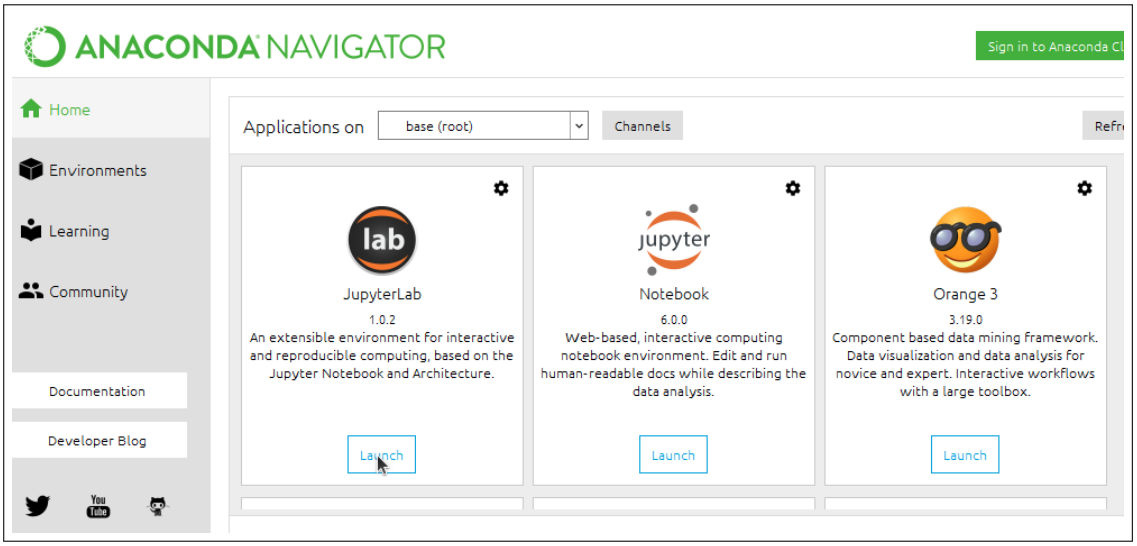

Abbildung 1: Starten von JupyterLab über den Anaconda Navigator.

Auf der linken Seite sehen Sie einen Datei-Explorer, der dem eines Betriebssystems ähnelt (siehe Abbildung 2). Dort können Sie durch Klicks auf Ordner in die gewünschte Verzeichnisstruktur navigieren, bis Sie das entsprechende Notebook-Verzeichnis erreicht haben (Informationen zum Herunterladen von Notebooks finden Sie in Abschnitt 2).

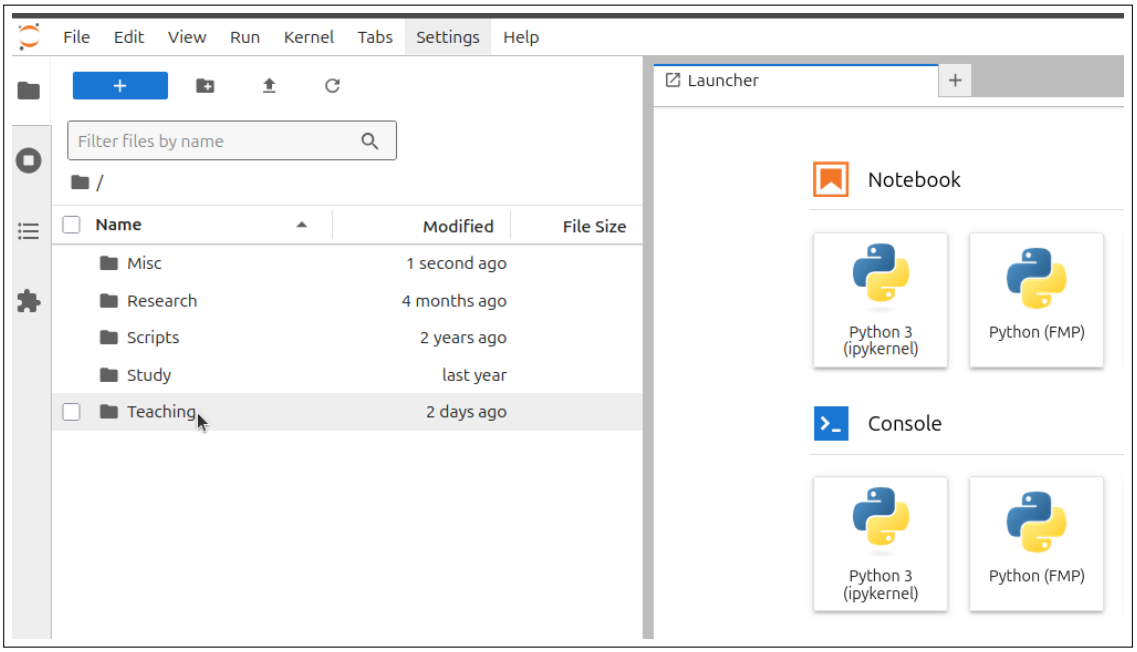

Abbildung 2: Übersicht von JupyterLab mit Dateibrowser.

Durch Anklicken eines Notebooks öffnet sich dieses im Arbeitsbereich von JupyterLab (siehe Abbildung 3).

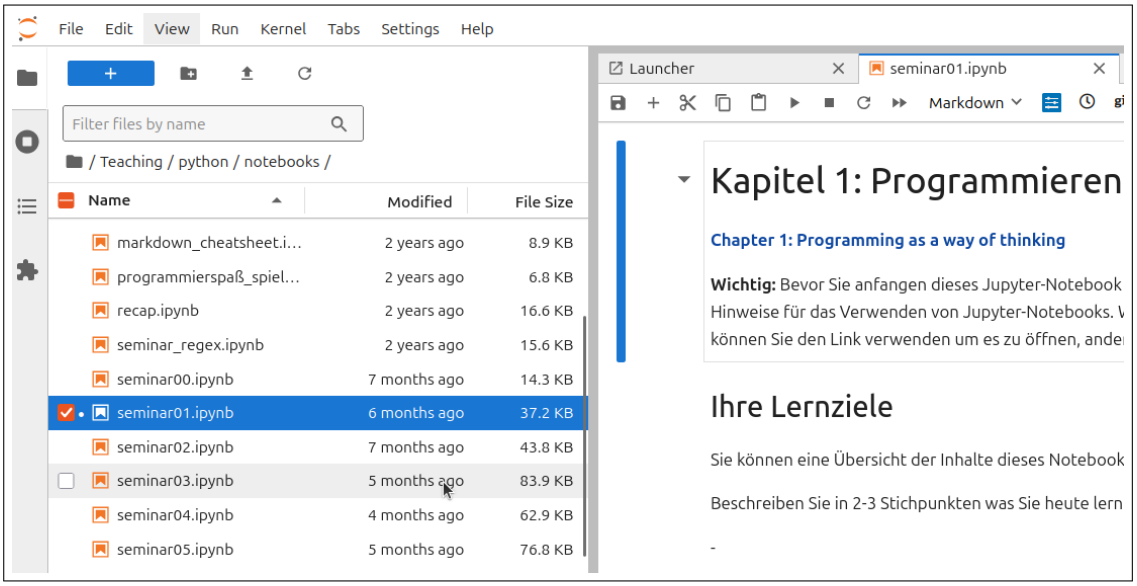

Abbildung 3: Geöffnetes Jupyter Notebook innerhalb von JupyterLab.

Änderungen in einem Notebook werden in der Regel automatisch gespeichert. JupyterLab speichert dabei regelmäßig im Hintergrund (standardmäßig alle 120 Sekunden). Zusätzlich kann jederzeit manuell gespeichert werden, indem Sie entweder auf das Disketten-Symbol der Toolbar klicken oder die Tastenkombination `Strg + S` (Windows/Linux) bzw. `Cmd + S` (macOS) verwenden.

1.2 Online: JupyterHub der HU

1.2.1 Jupyter Notebooks hochladen, bearbeiten und speichern

Beim Start des JupyterHub werden Sie zunächst aufgefordert, ein Profil auszuwählen. Sofern in der Lehrveranstaltung nichts anderes angegeben wird, wählen Sie bitte das erste Profil *Datascience JupyterLab* (siehe Abbildung 4).

Abbildung 4: Auswahl eines Profils beim Start des JupyterHub.

Nach dem Start befinden Sie sich auf einem Server der Humboldt-Universität in Ihrem persönlichen Arbeitsverzeichnis (siehe Abbildung 5). Die Oberfläche ähnelt der lokalen Nutzung von JupyterLab sehr stark, da auch der JupyterHub auf JupyterLab basiert.

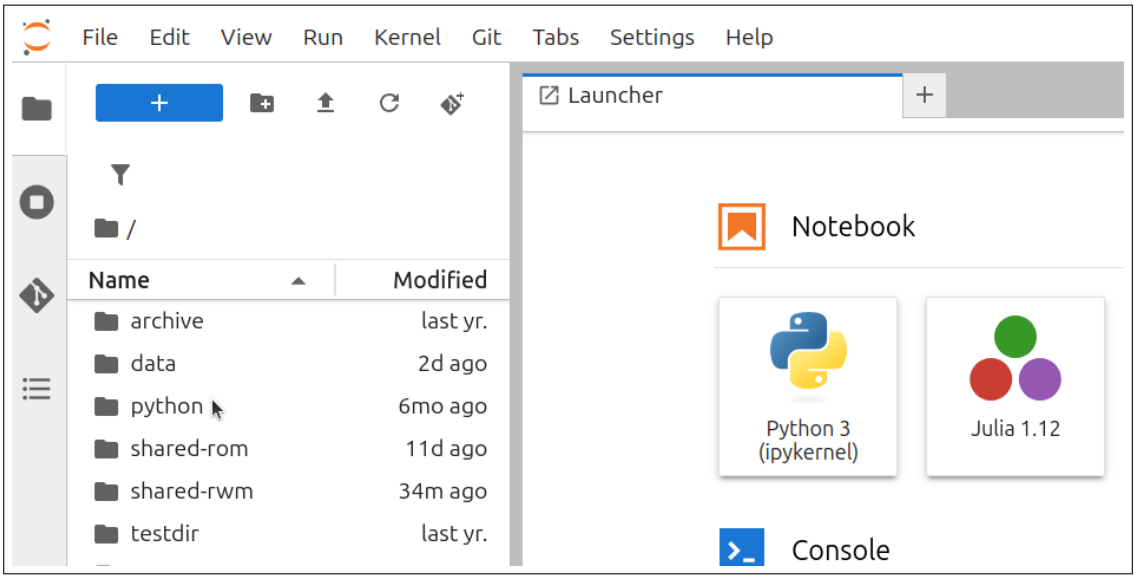

Abbildung 5: Arbeitsumgebung im JupyterHub.

Sie können nun lokale Notebooks in den JupyterHub hochladen und dort bearbeiten. Klicken Sie hierzu auf den Upload-Button ⬆ der Dateiansicht (siehe Abbildung 6).

Diese Vorgehensweise eignet sich insbesondere für das Hochladen einzelner Dateien. Alternativ können auch vollständige Repositorien direkt eingebunden werden, wie im folgenden Abschnitt beschrieben.

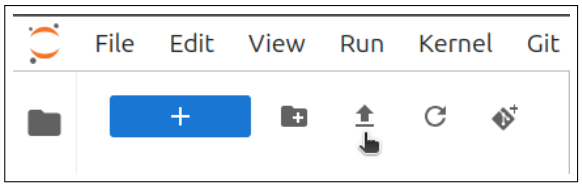

Abbildung 6: Hochladen von Dateien in den JupyterHub.

1.2.2 Repositorien über GitLab herunterladen

Neben der Upload-Funktion besteht auch die Möglichkeit, ein GitLab-Repository direkt in den JupyterHub zu klonen. Die entsprechende Option finden Sie in derselben Menüleiste, wie in Abbildung 7 dargestellt.

Abbildung 7: Klonen eines GitLab-Repositories im JupyterHub.

Anschließend öffnet sich ein Eingabefeld zur Angabe der Repository-Adresse (siehe Abbildung 8).

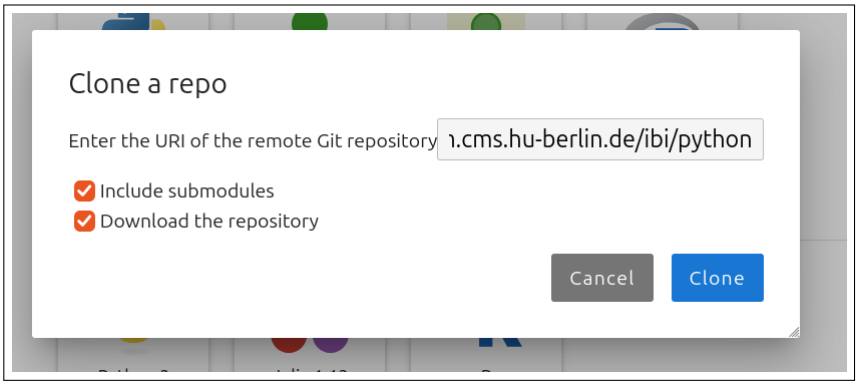

Abbildung 8: Eingabe der Repository-Adresse.

Geben Sie hier die Git-Clone-Adresse ein, wie in Kapitel 2.2 beschrieben, achten Sie darauf, dass *Download the repository* gesetzt ist und klicken Sie anschließend auf *Clone*. Das Repository wird daraufhin in Ihr Arbeitsverzeichnis geladen.

Im Beispiel des Python-Seminars muss anschließend noch in das Unterverzeichnis `notebooks` navigiert werden. Das Ergebnis ist in Abbildung 9 dargestellt.

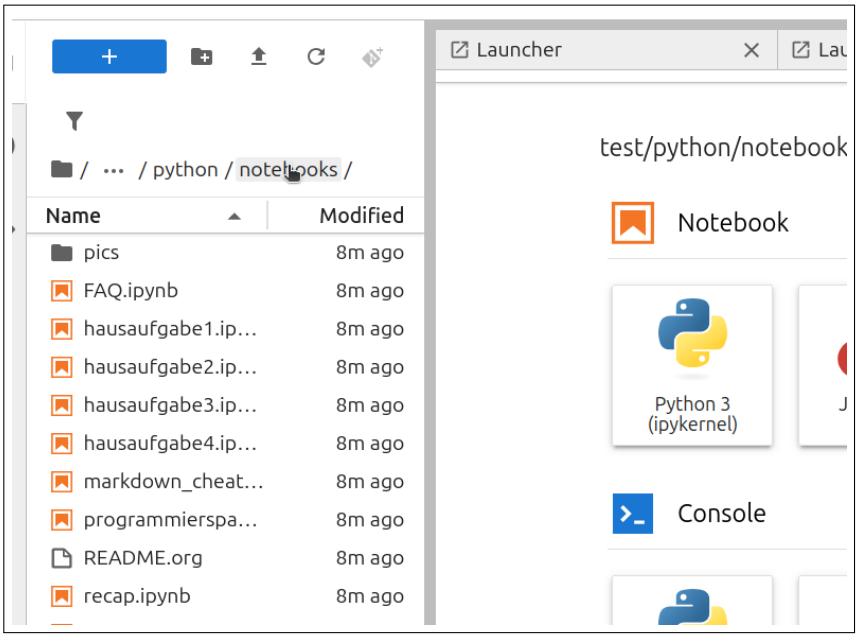

Abbildung 9: Geklontes Repository im JupyterHub.

Damit Ihre Notebooks stets auf dem aktuellen Stand bleiben, sollte regelmäßig (beispielsweise wöchentlich) ein `git pull` durchgeführt werden. Dies kann über den Reiter `Git` in der Menüleiste ausgeführt werden (siehe Abbildung 10).

Hinweis: Um versehentliche Überschreibungen zu vermeiden, empfiehlt es sich, bearbeitete Notebooks unter einem leicht angepassten Dateinamen zu speichern, beispielsweise durch das Anhängen der eigenen Initialen.

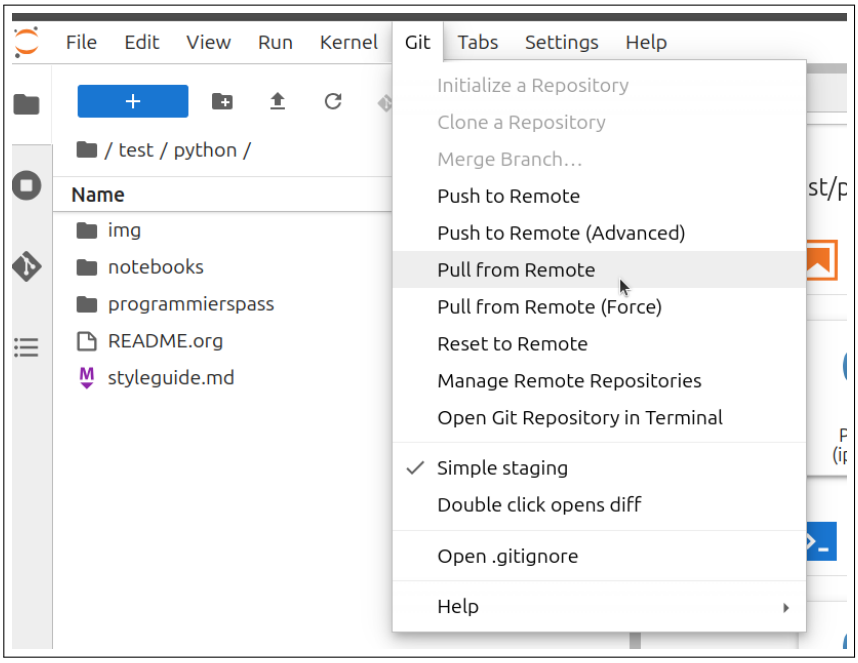

Abbildung 10: Ausführen von `git pull` im JupyterHub.

1.3 Online: Google Colab

Alternativ können Sie auch Google Colab nutzen. Ähnlich wie beim JupyterHub muss hierfür lediglich die entsprechende Webseite aufgerufen werden, und es ist ein Login mit einem Google-Konto erforderlich. Eine kurze Einführung in Google Colab finden Sie im entsprechenden Einführungsnotebook.

2 Jupyter Notebooks herunterladen

Bevor die Notebooks bearbeitet werden können, müssen sie zunächst heruntergeladen werden. Im Rahmen des Kurses wird üblicherweise ein Link zu einem GitLab-Repository bereitgestellt, in dem alle Notebooks sowie weitere Kursmaterialien zentral gesammelt sind.

2.1 Einzelne Notebooks herunterladen

Rufen Sie zunächst das GitLab-Repository auf, aus dem Sie das gewünschte Notebook herunterladen möchten. Im Beispiel verwenden wir das Repository des Python-Seminars. Anschließend klicken Sie auf das entsprechende Notebook (siehe Abbildung 11).

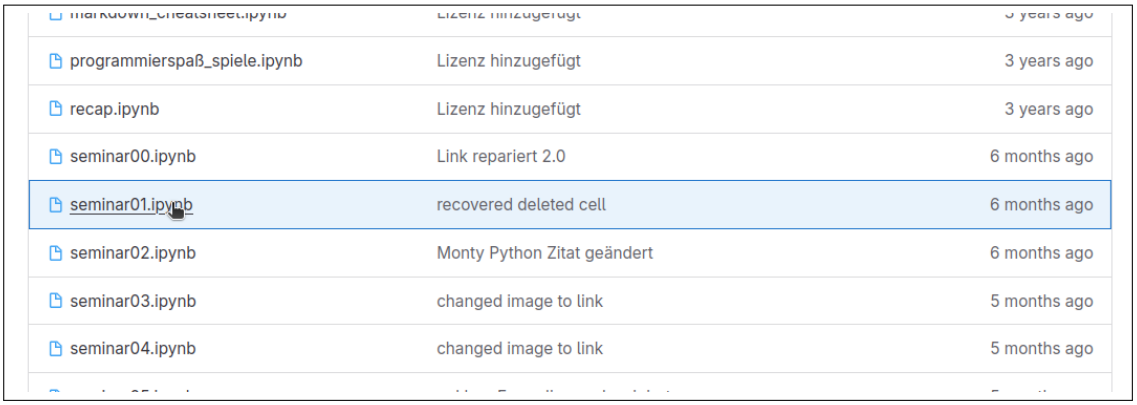

Abbildung 11: Auswählen des gewünschten Notebooks.

Auf der geöffneten Seite befindet sich oben rechts ein *Download*-Button, über den das Notebook heruntergeladen werden kann, wie in Abbildung 12 dargestellt. Das Notebook befindet sich dann in ihrem *Download*-Ordner. Wir empfehlen, für jede Lehrveranstaltung einen eigenen Ordner anzulegen und die heruntergeladenen Notebooks dort abzulegen.

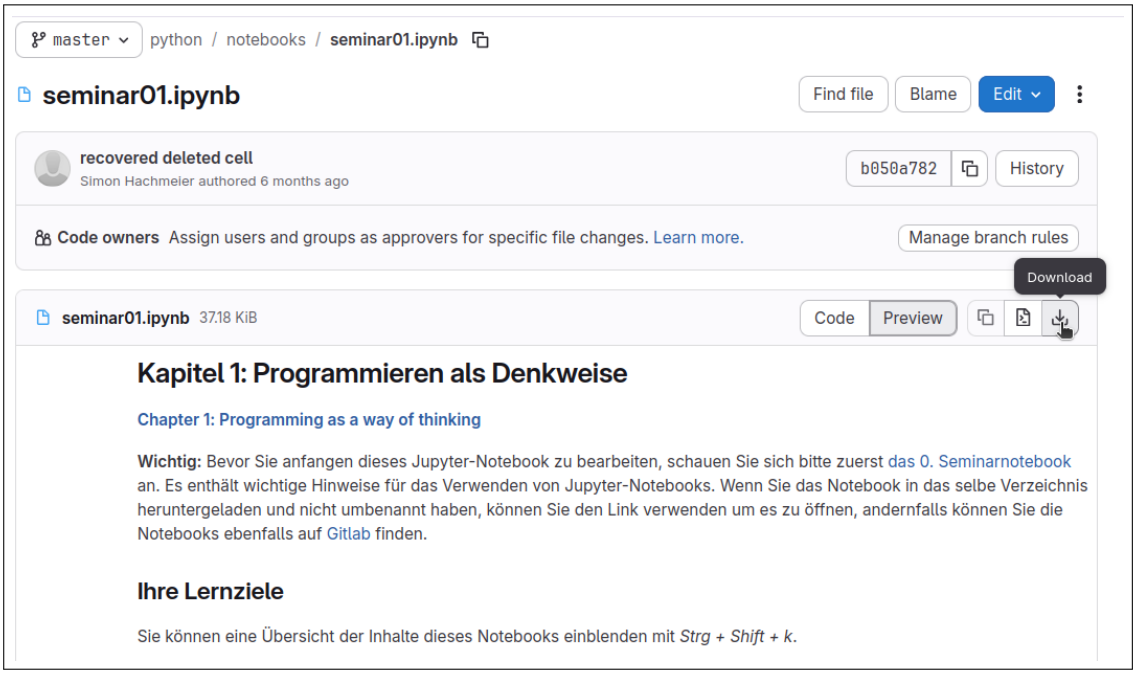

Abbildung 12: Herunterladen eines Notebooks über den *Download*-Button oben rechts. Das gezeigte Beispiel stammt aus dem ersten Seminarnotebook des Python-Seminars.

2.2 Nutzung von Git

Zunächst muss Git installiert werden. Anschließend kann das Repository wie in der Git-Anleitung beschrieben geklont werden (siehe Abbildung 13). **Achtung:** Das lokale Repository sollte regelmäßig aktualisiert werden, da sich die Inhalte im Verlauf der Veranstaltung ändern können. Dies erfolgt über den Befehl `git pull`, wie in der oben verlinkten Anleitung beschrieben.

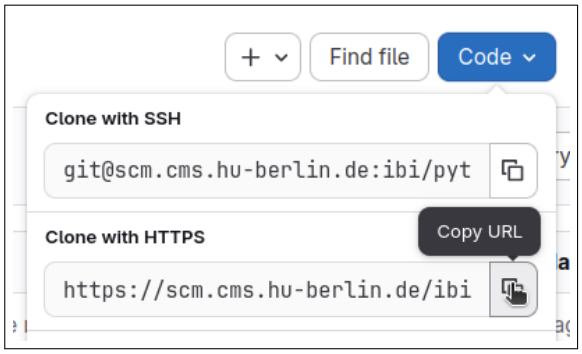

Abbildung 13: Clonen eines Repository aus GitLab.