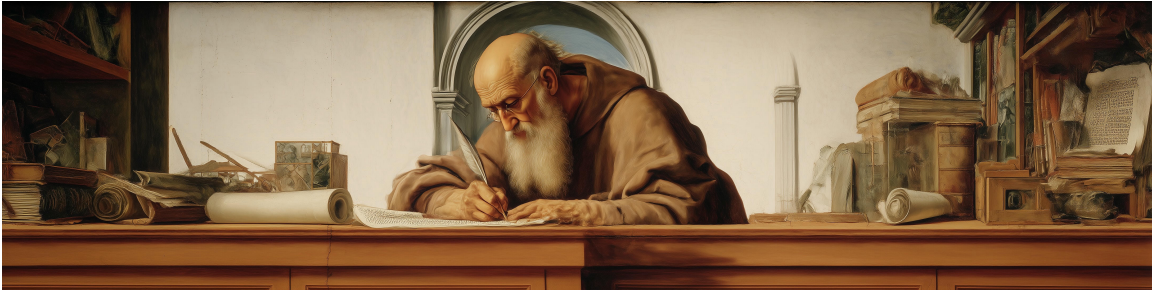


Jupyter Notebooks

Joern Ploennigs · AI4SC



I never wrote things down to remember; I always wrote things down so I could forget.

— Matthew McConaughey

In den letzten Jahren ist es immer beliebter geworden, Python über eine interaktive webbasierte Umgebung *Jupyter Notebook* zu verwenden. Notebooks bestehen aus *Code-Zellen* und *Markdown-Zellen* (Text-Zellen). Code-Zellen können Code enthalten, der mit einem einfachen Klick (oder Tastenkombinationen, z.B. Shift-Enter) ausgeführt werden kann. Der Code wird durch *ipython* ausgeführt, was eine Erweiterung von Python ist, die eine interaktive, schrittweise Programmierung erlaubt. Die Markdown-Zellen enthalten Markdown-Text und können durch einen ähnlichen Klick oder eine Tastenkombination gerendert werden. Der große Vorteil von Notebooks ist das sofortige Feedback: Man kann den Code ein paar Zeilen auf einmal schreiben, diese ausführen und sofort potenzielle Fehler korrigieren. Das macht Notebooks zu einer beliebten Methode für Data Science und Maschinellem Lernen in Python, da man leicht Code, Ausgaben und textliche Erklärungen in einem Notebook mischen kann, den Code dynamisch entwickeln kann und das Ergebnis in HTML-Seiten oder ein PDF-Dokument umwandeln kann, zur Dokumentation. Aber Notebooks sind nicht die Lösung für jedes Problem, so ist es nicht einfach Benutzerinteraktion zu realisieren. Notebooks erlauben auch das Ausführen von Zellen in beliebiger Reihenfolge, was zu Fehlern führen kann, die man in der traditionellen Codierung nicht sieht.

Oberfläche

Die Notebooks werden in einem Webbrowser angezeigt und laufen entweder auf einem lokalem Notebook-Server oder auf einem Server in der Cloud. Eine Ausführungsumgebung für Notebooks ist JupyterLab. Das ist ein Notebook-Server der vor allem erlaubt mit mehreren Notebooks komfortabel zu arbeiten.

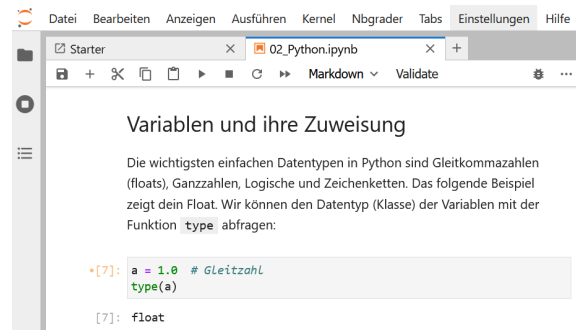


Abbildung 1: Oberfläche von Jupyter-Lab mit einem Jupyter-Notebook. Der Code befindet sich in der Code-Zelle, die Ausgabe ist darunter sichtbar.

Die Notebook-Zellen können entweder Code oder Markdown-Text enthalten. Die Abbildung zeigt drei Zellen in *Jupyterlab*. Die Zelle, die mit einer blauen Leiste links markiert ist, ist die aktive Zelle, in der man Code oder Text schreiben und bearbeiten kann. Dies ist derzeit eine Markdown-Zelle, wie im Dropdown-Menü oben rechts sichtbar (grün unterstrichen). Markdown ist eine vereinfachte Auszeichnungssprache, um Texte zu schreiben und zu formatieren. Hierbei verwendet man einfache Sonderzeichen, um Überschriften zu definieren oder Text zu formatieren. Ein Überblick über den Syntax findet sich hier. Als praktische Implikation kann man in Markdown Zellen auch HTML einfügen und so praktisch Webseiten erstellen.

Zellen können mit dem „Ausführen“-Dreieck ► in der Mitte der Iconleiste (grün unterstrichen) ausgeführt werden. Einfacher ist es sich anzugewöhnen die Zellen mit Shift-Enter zu schließen. Dadurch wird enthaltender Code ausgeführt oder der Markdown-Text gerendert.

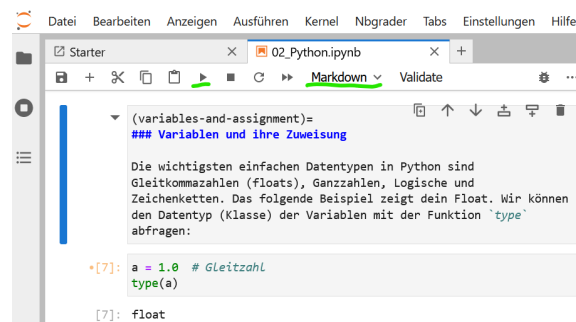


Abbildung 2: Das Hinzufügen von Markdown zu Notebooks. Man kann den Zellentyp aus dem Dropdown-Menü oben rechts auswählen und anschließend das Markdown mit dem Pfeil-Symbol ► rendern (beide unterstrichen), oder einfacher mit Shift-Enter.

Unten sehen wir eine Code-Zelle, die bereits ausgeführt wurde. Code-Zellen können an den Klammern links erkannt werden; die Zahl in der Klammer [7] bedeutet, dass diese Zelle im aktuellen Kernel als siebte Ausführung ausgeführt wurde. Wenn der Code irgendwelche Ausgaben erzeugt, sind diese auch darunter sichtbar. Das visuelle Layout und die Menüs sind etwas anders, wenn Notebooks außerhalb von *Jupyterlab* verwendet werden.

Jupyter Notebooks und Python installieren

Möchte man Jupyter Notebooks lokal ausführen, so muss man zuerst Python installieren. Python ist sehr beliebt um einfache Skripte für Betriebssystem (OS)-bezogene Aufgaben zu schreiben, deshalb kann bereits eine Version von Python installiert sein. Das sollte man prüfen, um Probleme zu vermeiden.

Für Maschinelles Lernen ist es empfohlen, einen guten Paketmanager für Python zu installieren, der beim Installieren genutzter Pakete auch Abhängigkeiten auflöst. Hier sind im Bereich des Maschinellen Lernens *Anaconda* oder *Miniconda* sehr beliebt. Anaconda ist ein Paketmanager für Python, der insbesondere beliebte Bibliotheken und viele andere verwandte Tools enthält. Sie können den Anaconda-Installer herunterladen und auf Ihrem Computer ausführen; alternativ ist auch Miniconda eine gute, schlankere Option. Wählen Sie die Standardeinstellungen, es sei denn, Sie wissen, was Sie tun.

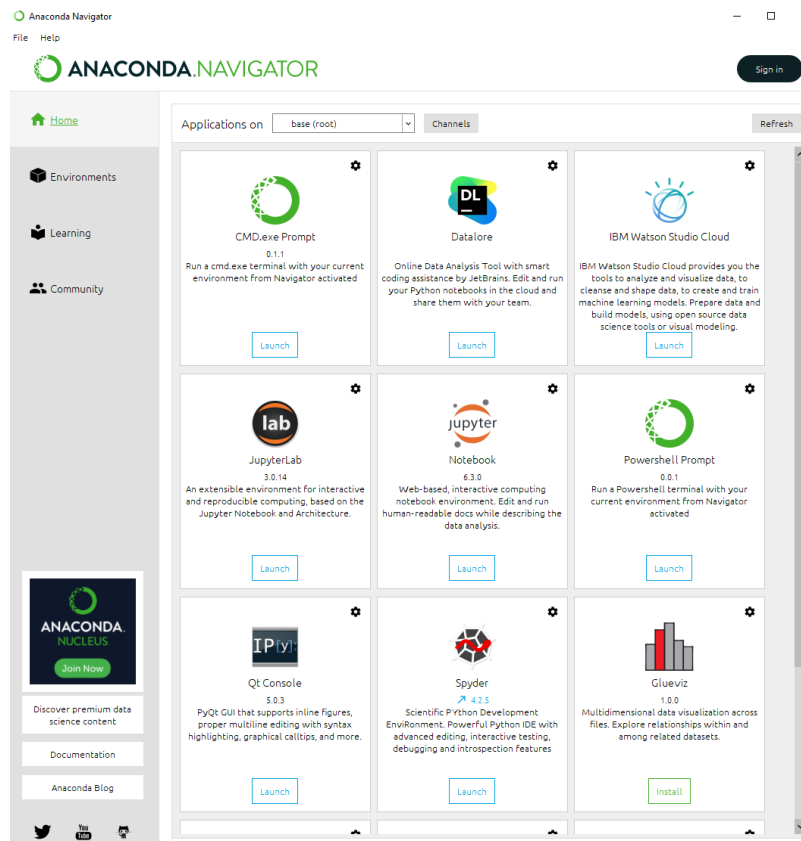


Abbildung 3: Anaconda Navigator in Windows

Auf Windows und Mac OS installiert Anaconda den *Anaconda Navigator*, eine grafische Benutzeroberfläche, die es Ihnen ermöglicht, auf die wichtigsten mit Anaconda verbundenen Programme zuzugreifen. Dazu gehören *JupyterLab* und *Jupyter Notebooks* für interaktives Codieren, der Texteditor *Spyder* und eine Anaconda-fähige Befehlszeilenschnittstelle. Letztere ist nützlich, wenn Sie Python-Programme von der Befehlszeile ausführen möchten.

Anaconda ermöglicht es auch, verschiedene Versionen von Python und anderen Paketen gleichzeitig über *Umgebungen* installiert zu halten. Sie können sich Umgebungen als verschiedene Ordner vorstellen, die verschiedene Versionen der Software enthalten. Wenn es jedoch das erste Mal ist, dass Sie Anaconda auf Ihrem Computer installieren, müssen Sie keine Umgebungen einrichten.

Python kann auf verschiedene Arten verwendet werden. Der traditionelle Weg besteht darin, einen Texteditor oder eine IDE zu verwenden, um ein Python-Programm zu schreiben (typischerweise mit der Dateierweiterung *.py*) und das Programm von der Befehlszeile aus auszuführen. Moderne IDEs unterstützen auch die Ausführung von Code mit einem einzigen Klick. Die Befehlszeile ist der bevorzugte Weg, um Python-Code (oder jeden anderen Programmcode) für Aufgaben auszuführen, die lange dauern oder im Hintergrund ohne Benutzerinteraktion laufen. Wenn Sie Anaconda verwenden, sollten Sie das entsprechende Anaconda-Befehlsfenster verwenden, um Python-Programme auszuführen.

Abrufen von Hausaufgaben

Die Hausaufgaben können auf der JupyterLab-Plattform heruntergeladen, bearbeitet und abgegeben werden. Hierfür dient der Menüpunkt „Nbgrader“. Die Option „Assignments“ ist die wichtigste, da sie die Seite mit den Aufgaben anzeigt. Die Option „Courses“ zeigt alle Kurse an denen man Teilnimmt. Die Option „Formgrader“ nutzen Lehrende, um die Aufgaben der Studenten einzusehen.

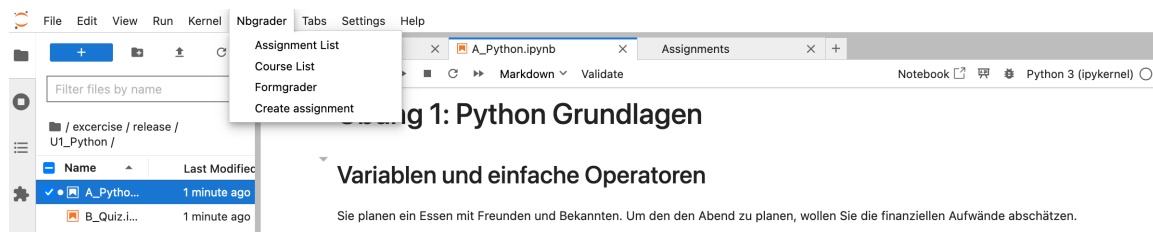


Abbildung 4: Nbgrader menu

Wählt man im „Nbgrader“ Menu die Option „Assignments“ so öffnet sich die Ansicht der aktuellen Aufgaben. Im Beispiel wird eine Aufgabe angezeigt. Durch betätigen von „fetch“ wird die ausgewählte Aufgabe heruntergeladen.



Abbildung 5: Nbgrader: Available assignments

Sie erscheint dadurch zum einen in der Liste der „Downloaded Assignments“. Durch Klick auf den blauen Namen der Aufgabe wird der Unterordner der Aufgabe geöffnet und angezeigt. Hier werden dann alle Notebooks und zugehörigen Dateien angezeigt.

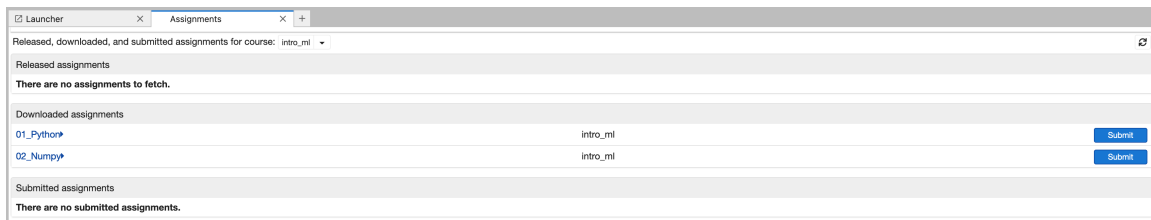


Abbildung 6: Nbgrader: Downloaded Assignments

Bearbeiten von Hausaufgaben

Lesen sie zuerst die Aufgabenstellung durch. In den Code-Zellen finden Sie dann einen Abschnitt `YOUR CODE HERE` oder `YOUR ANSWER HERE`. Löschen sie auch die ggf. folgende Zeile `raise NotImplementedError()`. Wenn Sie die Zelle erfolgreich bearbeitet haben, können Sie diese mit `Shift+Enter` ausführen. Achten Sie darauf, dass die Ausführung erfolgreich war.

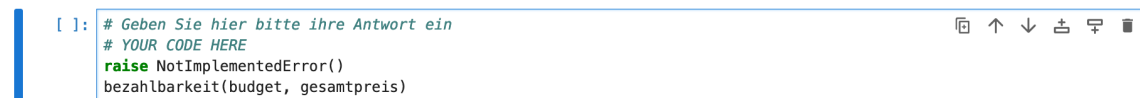


Abbildung 7: Nbgrader menu

Auf jede zu ergänzende Code-Zelle folgt normalerweise eine Test-Zelle. Diese können Sie nicht bearbeiten.



Abbildung 8: Nbgrader menu

Bevor Studierende ihre Aufgabe einreichen, kann die Funktion des Notebooks überprüft werden. Hierfür sollte es zum einen mit einem neuen Python-Kernel vollständig ausgeführt werden. In dem Notebook sollte dafür das  Icon in der Iconbar von dem Notebook mit dem Titel **Restart the kernel and run all cells** aufgerufen werden oder im Menü die Option **Kernel>Restart Kernel and Run all Cells...** ausgeführt werden. Dadurch wird das Notebook einmal komplett neu ausgeführt und Ihr könnt sicherstellen, dass keine Fehler durch fehlende temporäre Variablen bei der Bewertung auftreten.

Einreichen von Hausaufgaben

Vor dem Einreichen der Aufgaben sollte jedes Notebook validiert werden. Hierfür kann die Schaltfläche im Menü betätigt werden. Ist die Überprüfung erfolgreich, erscheint die Nachricht: „Success! Your notebook passes all the tests.“; wenn die Validierung nicht erfolgreich ist, wird ein Bericht mit den Fehlern angezeigt.



Abbildung 9: Nbgrader menu

Sobald alle Notebooks validiert wurden, kann die Schaltfläche „Submit“ betätigt werden. Dadurch wird die Aufgabe eingereicht und in die Liste der eingereichten Aufgaben aufgenommen.

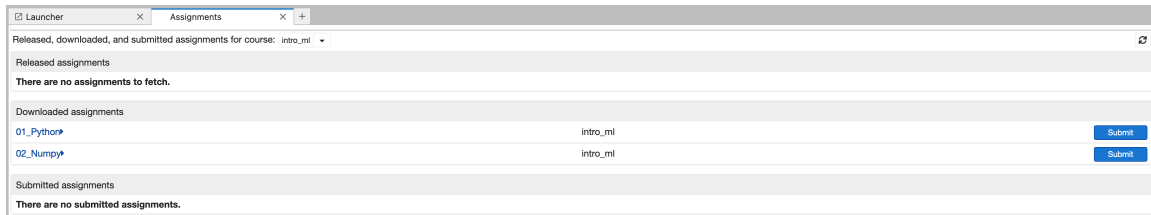


Abbildung 10: Nbgrader menu

Auch nach der Abgabe kann eine Aufgabe noch bearbeitet und über „Submit“ erneut abgegeben werden. Alle Versionen einer Abgabe werden in der Liste der abgegebenen Aufgaben angezeigt. Wenn die Lehrkraft die Aufgabe abholt, erhält sie automatisch die neueste Version der Abgabe.

Das Feedback und die Bewertung für die Aufgabe kann über „Feedback abrufen“ eingesehen werden, sobald sie veröffentlicht wurde.

Server herunterfahren

Die JupyterLab Umgebung sollte ordnungsgemäß beendet werden, wenn die Arbeit abgeschlossen worden ist. Hierfür sollte man über die Option „Hub Control Panel“ (File -> Hub Control Panel) in die Serverkontrollansicht wechseln. Hier kann der Server mit „Stop my Server“ beendet werden.

Bibliographie