

Woche 01: Programmierung - Einführung in Jupyter Notebook

# Skript

Erarbeitet von  
Ludmila Himmelpach

Lernziele .....	1
Inhalt .....	2
Einstieg .....	2
Jupyter Notebook .....	2
Erstes Programm in Python .....	3
Zweites Programm in Python .....	4
Kommentare in Python .....	4
Run All .....	5
Aufgabenlösungen überprüfen .....	6
Take-Home Message .....	6
Quellen .....	6
Disclaimer .....	6

## Lernziele

- Die wichtigsten Funktionalitäten von Jupyter Notebook erklären können
- Im Jupyter Notebook ein einfaches Programm in Python schreiben und ausführen können
- Erklären können, was die print()-Anweisung in Python bewirkt
- Erklären können, was Kommentare sind und wie diese im Programmcode eingefügt werden

## Inhalt

### Einstieg

*Jupyter Notebook* ist die Programmierumgebung, die wir in unserem Kurs benutzen werden. Mit Jupyter Notebook kannst du deine Programme in Python schreiben und ausführen lassen. Außerdem kannst du damit alle Programme aus den Videos testen, die wir dir nach jedem Programmiervideo zur Verfügung stellen werden. Da Jupyter Notebook in einem Internetbrowser läuft, kannst du es sowohl auf deinem Computer zu Hause als auch auf deinem Laptop, Tablet oder sogar Smartphone unterwegs benutzen. Wir werden uns gleich gemeinsam die wichtigsten Funktionalitäten von Jupyter Notebook anschauen und das erste Programm in Python schreiben.

### Quelle [1]

#### Jupyter Notebook

Hier siehst du die Standardansicht einer Programmieraufgabe im Lern-Management-System von KI-Campus. Oben siehst du den Titel und die Beschreibung zu der Aufgabe. Außerdem siehst du, dass es sich um eine unbenotete Aufgabe handelt. Um die Aufgabe zu starten, musst du auf den unteren dunkellila Button mit der Aufschrift „Aufgabe starten“ klicken, der dich zum Jupyter Notebook weiterleitet. Da unsere Notebooks über das Jupyter Hub von KI-Campus ausgeführt werden, brauchst du dir keine Gedanken über die Installation von Compilern und Interpretern auf deinem Computer zu machen.

Ganz oben siehst du den Dateinamen von Jupyter Notebook. Darunter kommt das Menü und die Toolbar, in der du auf die wichtigsten Funktionen des Notebooks direkt zugreifen kannst. Unterhalb der Toolbar siehst du das eigentliche Notebook. Es sieht wie ein Blatt Papier aus, besteht aber aus Zellen. Im Jupyter Notebook gibt es zwei Arten von Zellen: die Textzellen und die Codezellen. Die Textzellen werden meistens dazu benutzt, um die Programmteile zu beschreiben. In den Notebooks wie diesem, die wir als Aufgabenblätter nutzen, findest du in den Textzellen die Aufgabenstellung. In den Codezellen wird das eigentliche Programm oder die Programmteile geschrieben, die du ausführen kannst.

Wir kommen zurück zur Toolbar. Du wirst nur einige Funktionen davon benötigen. Ganz links siehst du das Speichersymbol. Wenn du zum Beispiel ein Aufgabenblatt bearbeitest, solltest du am besten die Änderungen, die du an deinem Notebook vorgenommen hast, zwischendurch mal speichern, damit sie nicht verloren gehen. Alle deine Änderungen werden zwar automatisch gespeichert, aber um sicher zu gehen, empfehlen wir dir dein Notebook wenigstens vor dem Schließen einmal zu speichern.

Rechts vom Speichersymbol findest du einige Buttons, die zur Erstellung eines Notebooks nützlich sind. Da du in diesem Kurs keine Notebooks erstellen musst, überspringen wir diese und gehen direkt zum Button „Run“ über. Mit diesem Button kannst du eine Codezelle bzw. das Programm oder ein Programmteil in einer Codezelle, die gerade aktiv ist, ausführen

lassen. Durch das Klicken auf eine Codezelle wird diese aktiviert bzw. ausgewählt. Das machen wir jetzt mit der ersten Codezelle. Durch einen Klick auf den „Run“-Knopf führen wir diese aus. Es passiert nichts, weil unsere Codezelle noch keine einzige Anweisung enthält. Später in diesem Video siehst du, was passiert, wenn wir eine Codezeile mit Anweisungen ausführen.

Rechts neben dem „Run“-Button siehst du einen Knopf mit einem schwarzen Quadrat darauf. Wenn du deinen Mauszeiger auf diesen Knopf bewegst, siehst du die Erklärung zu seiner Funktion: „interrupt the kernel“. Du fragst dich jetzt wahrscheinlich, was eigentlich ein Kernel ist. Ein *Kernel* ist ein Programm, das die Programme bzw. die Programmteile in den Codezellen deines Jupyter Notebooks ausführt. Durch einen Klick auf den „interrupt the kernel“-Knopf kannst du die Ausführung deines Programms unterbrechen. Auf die Funktion weiterer Buttons kommen wir noch mal später zurück, nachdem wir etwas programmiert haben.

## Erstes Programm in Python

Jetzt wollen wir aber gemeinsam unser erstes Programm in Python schreiben. In der ersten Aufgabe sollen wir ein Programm schreiben, das die Begrüßung „Hallo zusammen!“ ausgibt. Um einen Text in Python ausgeben zu können, wird die Anweisung `print()` benutzt.

Wenn wir in den Codezellen Python-Befehle eingeben, ist es wichtig, dass diese direkt am Anfang einer Zeile, ohne Leerzeichen davor beginnen. Das Einrücken von Anweisungen hat in Python eine eigene Bedeutung. Vielleicht erinnerst du dich an das Beispiel im Einführungsvideo. Da steht die `sleep()`-Anweisung im Funktionscode eingerückt. Die Leerzeichen, die nicht am Anfang einer Zeile stehen, ignoriert Python, also verursachen sie keine Probleme.

Um unsere Begrüßung auf der Ausgabefläche anzeigen zu lassen, müssen wir sie in die runden Klammern der `print()`-Anweisung schreiben. Dabei werden die Texte in Anführungszeichen geschrieben. Alles, was innerhalb der Anführungszeichen geschrieben wird, wird als Text erkannt. In Python stehen uns zwei Optionen zur Verfügung. Wir können entweder die doppelten („...“) oder die einfachen Anführungszeichen ('...') benutzen. Welche Vorteile uns das bietet, sehen wir noch später in diesem Video. Zuerst setzen wir unseren Text in die doppelten Anführungszeichen.

```
print("Hallo zusammen!")
```

Jetzt können wir unser Programm ausführen lassen, um zu sehen, ob es unsere Begrüßung ausgibt. Dafür klicken wir auf den Ausführungsknopf oben. Es sieht gut aus! In der Ausgabe unter der Codezelle sehen wir unsere Begrüßung „Hallo zusammen!“. Damit haben wir erfolgreich das erste Programm in Python geschrieben.

## Zweites Programm in Python

In der zweiten Aufgabe soll unser Programm den Text 'Willkommen bei der Veranstaltung „KI für alle“!' ausgeben. Dazu schreiben wir diesen Text wieder in der Klammer der `print()`-Anweisung.

```
print("Willkommen bei der Veranstaltung "KI für alle"!")
```

Wir führen unser Programm direkt aus. Wir bekommen leider eine Fehlermeldung. Genaugenommen handelt es sich um einen Syntaxfehler.

Diesen können wir eigentlich direkt an der Farbmarkierung in der Codezelle erkennen. In der ersten Aufgabe wird der gesamte Ausgabertext zusammen mit den Anführungszeichen rot (im Skript grün) angezeigt. In unserem zweiten Programm wird nur ein Teil des Ausgabertextes rot (im Skript grün) markiert, und zwar vom ersten bis zum zweiten Anführungszeichen. Danach kommt ein Teil, der nicht mehr als Text erkannt wird. Und am Ende wird wieder ein Teil des Ausgabertextes als Text erkannt, weil er von Anführungszeichen umschlossen ist.

Die Schwierigkeit besteht darin, dass wir in der zweiten Aufgabe Anführungszeichen im Text verwenden müssen. Um unseren gesamten Ausgabertext als Text von Python erkennen zu lassen, müssen wir zwei unterschiedliche Arten von Anführungszeichen verwenden. Ich habe schon gesagt, dass Python sowohl die einfachen als auch die zweifachen Anführungszeichen akzeptiert. Die doppelten Anführungszeichen nehmen wir für den Titel unserer Veranstaltung. Mit den einfachen Anführungszeichen umschließen wir unseren Ausgabertext. D. h. wir zeigen Python, wo unser Ausgabertext anfängt und endet.

```
print('Willkommen bei der Veranstaltung "KI für alle"!')
```

Jetzt können wir unser Programm ausführen lassen. Es funktioniert! In der Ausgabe unter der Codezelle wird unser Text richtig angezeigt. Somit ist unser zweites Programm in Python fertig.

## Kommentare in Python

In der dritten Aufgabe müssen wir unsere Programme aus den ersten beiden Aufgaben kommentieren. Dazu kopieren wir die Programmcodes bzw. Programmtexte in die Codezellen dieser Aufgabe und ergänzen sie mit Kommentaren. Ein *Kommentar* ist eine Art Notiz, die man im Programmcode für sich und andere schreibt. Ein Kommentar kann ein Programm oder einzelne Programmteile beschreiben, hat aber keine Auswirkung auf die Ausführung des Programms.

In der Praxis wird am Anfang jedes Programms ein Kommentar geschrieben, der die Aufgabe des Programms kurz beschreibt. Auch während des Schreibens eines Programms sind die Kommentare für einzelne Programmteile hilfreich, zum Beispiel, wenn man diese später

bearbeiten möchte. Wir werden die Kommentare manchmal dazu nutzen, um einige Hinweise und Hilfestellungen zu den Programmieraufgaben zu geben.

In Python werden Kommentare durch das Raute-Zeichen (#) eingeleitet. Alles, was nach dem Raute-Zeichen in der Zeile steht, wird bei der Ausführung des Programms ignoriert. Deswegen müssen die Kommentare am Ende der Zeile stehen.

Wir können unseren zweiten `print()`-Befehl wie folgt kommentieren: „# Nutze einfache Anführungszeichen, um den Text zu umschließen“. Da die Programmzeile schon relativ lang ist, schreiben wir unseren Kommentar in die Zeile davor.

```
# Nutze einfache Anführungszeichen, um den Text zu umschließen  
print('Willkommen bei der Veranstaltung "KI für alle"!')
```

Wenn wir jetzt das Programm ausführen, bekommen wir die gleiche Ausgabe wie in Aufgabe 2 und keinen Syntax-Fehler, weil wir den Text unseres Kommentars nicht ordentlich mit den Anführungszeichen umschlossen haben. Das ist passiert, weil der Compiler bzw. Interpreter die Kommentare komplett ignoriert.

Möchte man Kommentare einfügen, die über mehrere Zeilen gehen, setzt man drei einfache ('''') oder drei doppelte (""") Anführungszeichen jeweils vor und nach dem Kommentar.

Damit können wir zum Beispiel unser gesamtes Programm aus Aufgabe 1 kommentieren.

```
'''Das Programm gibt  
eine Begrüßung aus'''  
print("Hallo zusammen!")
```

Wir können das Programm ausführen lassen und sehen, dass sich nichts an der Ausgabe des Programms geändert hat.

## Run All

Wenn du in deinem Notebook oder in einem Notebook zu einem Video alle Codezellen von oben nach unten auf einmal ausführen möchtest, kannst du das durch einen Klick auf den Knopf mit zwei schwarzen Dreiecken machen. Dabei wird der Kernel neu gestartet und auch alle Änderungen – aber nicht der Code – gehen verloren, die durch die Ausführung deines Programms davor gemacht wurden. Wenn du nur alle Codezellen des Notebooks ausführen möchtest, ohne den Kernel neu zu starten, musst du in das Menü gehen, dann in den „Cell“-Bereich wechseln und „Run All“ auswählen.

## Aufgabenlösungen überprüfen

Jetzt bist du mit den wichtigsten Funktionalitäten des Jupyter Notebooks vertraut und kannst die Programmieraufgaben lösen. Aber wie kannst du selbst überprüfen, ob deine Lösungen richtig sind? Bei einigen Aufgaben siehst du das direkt an der Ausgabe. Bei den anderen Aufgaben ist es nicht so einfach. Deswegen findest du nach jedem Jupyter Notebook, das Programmieraufgaben für dich enthält, noch eine Übung im Lern-Management-System auf KI-Campus. Diese Übungen bestehen in der Regel aus Multiple-Choice-Aufgaben, die in direktem Bezug zu den Programmieraufgaben stehen. Du kannst diese Aufgaben entweder direkt im Anschluss an die Programmieraufgaben machen oder parallel dazu. Wenn du mal gar nicht weißt, wie du eine Programmieraufgabe lösen sollst, können dir die Multiple-Choice-Aufgaben sogar dabei helfen, auf die Lösung zu kommen.

## Take-Home Message

In diesem Video hast du die wichtigsten Funktionalitäten des Jupyter Notebooks kennen gelernt. Du weißt jetzt, wie du ein einfaches Programm in Python schreiben kannst, das einen Text ausgibt. Du weißt auch, wie du deine Programme bzw. Programmteile in Jupyter Notebook ausführen kannst. Außerdem weißt du, wie du deine Programme in den Codezellen mit sinnvollen Kommentaren ergänzen kannst.

## Quellen

Quelle [1] Kluver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., ... & Willing, C. (2016). *Jupyter Notebooks – a publishing format for reproducible computational workflows* (Vol. 2016, pp. 87-90).

## Disclaimer

Transkript zu dem Video „Woche 01: Programmierung - Einführung in Jupyter Notebook“, Ludmila Himmelpach.

Dieses Transkript wurde im Rahmen des Projekts ai4all des Heine Center for Artificial Intelligence and Data Science (HeiCAD) an der Heinrich-Heine-Universität Düsseldorf unter der Creative Commons Lizenz [CC-BY](https://creativecommons.org/licenses/by/4.0/) 4.0 veröffentlicht. Ausgenommen von der Lizenz sind die verwendeten Logos, alle in den Quellen ausgewiesenen Fremdmaterialien sowie alle als Quellen gekennzeichneten Elemente.