

Programmieren für Data Science und AI

Übung 1

1. Python installieren

1. Laden Sie die offizielle Python Version von <https://www.python.org> herunter und installieren Sie sie.
2. Installieren Sie die zusätzlich benötigten Bibliotheken, indem Sie folgende Befehle im Befehlszeilenfenster (Command-line, Terminal, oder Shell) ausführen:

```
pip install numpy      (bzw. pip3 install numpy bei MacOS)
pip install pandas
pip install matplotlib
pip install seaborn
```

3. Installieren Sie Jupyter Lab

```
pip install jupyterlab
```

NB: Weitere Informationen zu JupyterLab und Jupyter Notebooks finden Sie auf <https://jupyter.org>

2. Testen Sie IPython im interaktiven Modus

2.1. Öffnen von IPython im interaktiven Modus

Öffnen Sie ein Befehlszeilenfenster (Command-line, Terminal, oder Shell).
Geben Sie im Befehlszeilenfenster

```
ipython
```

ein und drücken Sie dann die Eingabetaste (Return)

Sie sollten nun einen ähnlichen Text wie hier sehen. Die konkrete Ausgabe variiert je nach Plattform und IPython-Version, die Sie installiert haben.

```
Python 3.10.1 (v3.10.1:2cd268a3a9, Dec 6 2021, 14:28:59) [Clang 13.0.0 (clang-1300.0.29.3)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.31.0 -- An enhanced Interactive Python. Type '?' for help.
```

```
In [1]:
```

Der Text "In [1]:" ist eine Eingabeaufforderung, die anzeigt, dass IPython auf Ihre Eingabe wartet. Sie können ? für Hilfe eingeben oder mit der Eingabe von Snippets beginnen, wie Sie es gleich tun werden.

2.2. Auswertung von Ausdrücken (Snippets)

Geben Sie folgenden Ausdruck in der Eingabeaufforderung ein und beenden Sie die Eingabe mit der Eingabetaste (Return).

```
In [1]: 45 + 72  
Out[1]: 117
```

```
In [2]:
```

Nachdem Sie `45 + 72` eingegeben und die Eingabetaste gedrückt haben, liest IPython das Snippet, wertet es aus und gibt sein Ergebnis (117) in `Out[1]` aus. Dann zeigt IPython die Eingabeaufforderung `In [2]` an, um zu zeigen, dass es darauf wartet, dass Sie Ihr zweites Snippet eingeben. Für jedes neue Snippet erhöht IPython die Zahl in den eckigen Klammern um 1.

2.3. Den interaktiven Modus verlassen

Geben Sie den Befehl

```
exit
```

in der aktuellen Eingabeaufforderung `In [2]` ein und drücken Sie Eingabetaste (Return), um den Modus sofort zu verlassen.

Alternativ können Sie auch die Tastenfolge `<Ctrl> + d` (bzw. `<control> + d`) eingeben. Dadurch wird die Eingabeaufforderung "Wollen Sie den Modus wirklich verlassen ([y]/n)?" angezeigt. Die eckigen Klammern um `[y]` zeigen an, dass es sich um die Standardantwort handelt. Wenn Sie nun die Eingabetaste drücken, wird die Standardantwort gesendet und das Programm beendet.

Nur MacOS und Linux: Geben Sie `<Ctrl> + d` (oder `<control> + d`) zweimal ein, um das Programm sofort zu beenden.

3. Ausführen eines Python-Programms mit dem IPython-Interpreter

In diesem Abschnitt werden Sie ein Skript namens *RollDieDynamic.py* ausführen. Dazu müssen Sie zum aktuellen Zeitpunkt den Inhalt des Files nicht verstehen. Sie werden dieses Programm ev. später in der Vorlesung kennen lernen.

Die Erweiterung `.py` zeigt an, dass die Datei Python-Quellcode enthält. Das Skript *RollDieDynamic.py* simuliert das Rollen eines sechsseitigen Würfels. Es präsentiert eine farbenfrohe, animierte Visualisierung, welche die Anzahl der geworfenen Würfelseiten dynamisch, grafisch darstellt.

3.1. Wechseln Sie in den Ordner Uebung01

Wechseln Sie in der Kommandozeile in den Ordner *Uebung01* auf Ihrem System, falls Sie sich nicht schon da befinden.

Angenommen, Sie haben den Übungsordner im Ordner "*Documents/prog*" abgelegt, dann können Sie mit folgendem Befehl in diesen *Uebung01* Ordner wechseln:

- Geben Sie unter macOS/Linux `cd ~/Documents/prog/Uebung01` ein und drücken Sie dann die Eingabetaste.

- Unter Windows geben Sie `cd C:\Users\YourAccount\Documents\prog\Uebung01` ein, und drücken Sie dann die Eingabetaste. Ersetzen Sie dabei im oben genannten Pfad *YourAccount* mit ihrem konkreten Accountname.

3.2. Ausführen des Scripts

Um das Skript auszuführen, geben Sie den folgenden Befehl in die Befehlszeile ein und drücken Sie dann die Eingabetaste:

```
ipython RollDieDynamic.py 6000 1
```

Das Skript zeigt nach ein paar Sekunden ein Fenster mit der dynamischen Visualisierung an. Die Zahlen 6000 und 1 teilen diesem Skript mit, wie oft gewürfelt werden soll und wie viele Würfel jedes Mal zu werfen sind. In diesem Fall würfeln wir 1 Würfel 6000-mal.

Bei einem sechsseitigen Würfel sollten die Augenzahlen 1 bis 6 jeweils mit "gleicher Wahrscheinlichkeit" auftreten - die Wahrscheinlichkeit beträgt jeweils $1/6$ oder etwa 16,667%. Wenn wir einen Würfel 6000-mal würfeln, würden wir daher etwa eine Häufigkeit 1000 von jeder Seite erwarten. Wie das Werfen von Münzen ist auch das Würfeln zufällig, so dass es einige Seiten mit weniger als 1000, einige mit 1000 und einige mit mehr als 1000 geben könnte.

Experimentieren Sie mit dem Skript, indem Sie den Wert 1 in 100, 1000 und 10000 ändern, d.h. gleichzeitig 100, 1000, oder 10000 Würfel gleichzeitig werfen. Probieren Sie auch, die Anzahl Würfe (im Beispiel 6000) zu variieren.

Sie werden feststellen, dass sich mit zunehmender Anzahl der Würfelwürfe die Häufigkeit auf 16,667% annähert. Dies ist ein Phänomen des "Gesetzes der grossen Zahlen".

4. Schreiben und Ausführen von Code in einem Jupyter Notebook

Ein *Jupyter Notebook* läuft in einer interaktiven, browserbasierten Umgebung. Sie können in einem Jupyter Notebook Code schreiben und ausführen und den Code mit Text, Bildern und Video erweitern.

Jupyter Notebooks werden in der Data Science-Gemeinschaft im Besonderen und in der breiteren wissenschaftlichen Gemeinschaft im Allgemeinen in grossem Umfang verwendet. Sie sind das bevorzugte Mittel, um Python-basierte Datenanalysestudien durchzuführen und ihre Ergebnisse reproduzierbar zu kommunizieren.

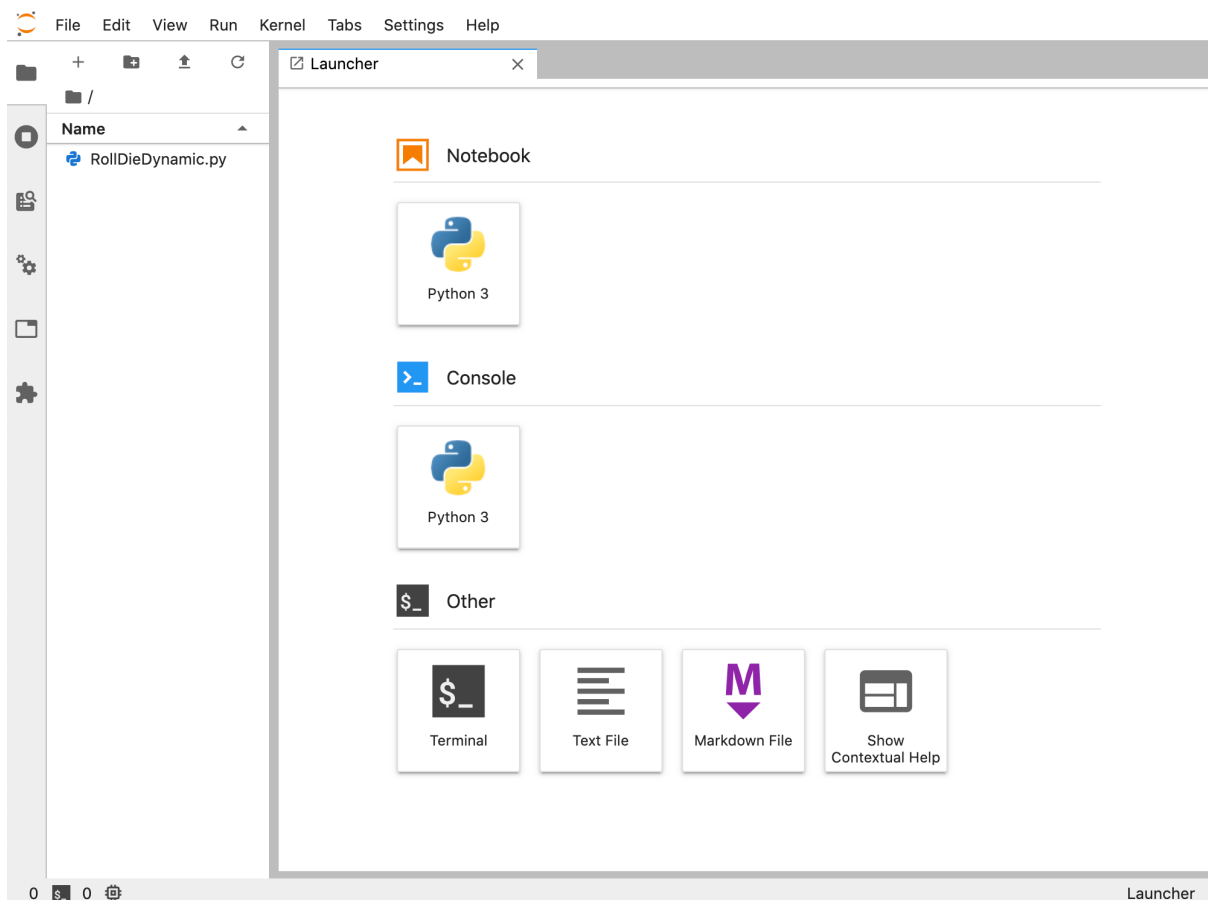
In diesem Abschnitt verwenden Sie die *JupyterLab*-Schnittstelle, mit der Sie Ihre Notebook-Dateien und andere Dateien, die Ihre Notebooks verwenden (wie Bilder und Videos), verwalten können. Wie Sie sehen werden, ist es mit *JupyterLab* auch bequem, Code zu schreiben, auszuführen, die Ergebnisse zu sehen, den Code zu modifizieren und erneut auszuführen.

4.1. Öffnen von JupyterLab in Ihrem Browser

Um JupyterLab zu öffnen, wechseln Sie in den Ordner *Uebung01* in Ihrem Befehlszeilenfenster (Command-line, Terminal, oder Shell). Geben Sie dann den folgenden Befehl ein und drücken Sie dann die Eingabetaste (Return):

```
jupyter lab
```

Dadurch wird der Jupyter Notebook-Server auf Ihrem Computer ausgeführt und JupyterLab in Ihrem Standard-Webbrowser geöffnet, wobei der Inhalt des Ordners *Uebung01* im File Browser auf der linken Seite von JupyterLab angezeigt wird.



Dateien im File Browser können Sie doppelklicken, um Sie zu öffnen. Sie werden dann als separate Registerkarte angezeigt.

Wenn Sie Ihren Browser versehentlich geschlossen haben, können Sie JupyterLab erneut öffnen, indem Sie die folgende Adresse in Ihren Webbrowser eingeben

```
http://localhost:8888/lab
```

4.2. Ein neues Jupyter Notebook erstellen

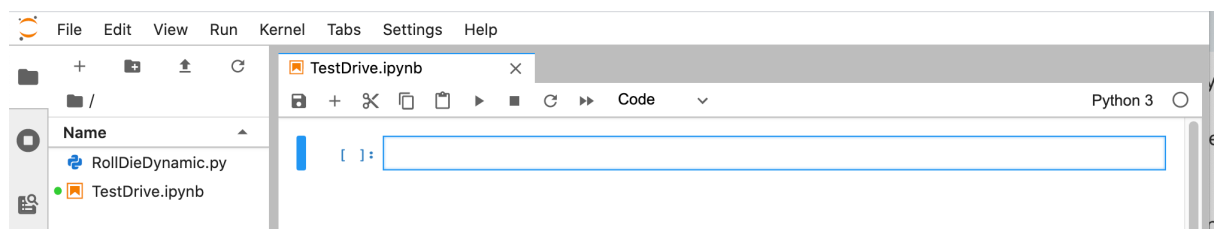
Klicken Sie in der Registerkarte *Launcher* auf die Schaltfläche Python 3, um ein neues Jupyter Notebook mit dem Namen *Untitled.ipynb* zu erstellen. Die Dateierweiterung *.ipynb* ist eine Abkürzung für IPython Notebook. In diesem Jupyter Notebook können Sie dann Python 3 Code eingeben und ausführen.

4.3. Jupyter Notebook umbenennen

Benennen Sie *Untitled.ipynb* in *TestDrive.ipynb* um:

Klicken Sie mit der rechten Maustaste auf die Registerkarte *Untitled.ipynb* und wählen Sie Notebook umbenennen. Ändern Sie den Namen in *TestDrive.ipynb*.

Der obere Teil von JupyterLab sollte nun wie folgt aussehen:



4.4. Auswerten eines Ausdrucks

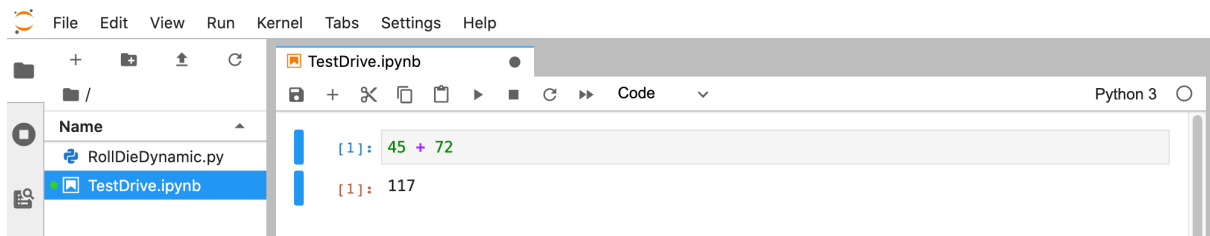
Die Arbeitseinheit in einem Jupyter Notebook ist eine Zelle, in die Sie Code-Schnipsel eingeben können. Standardmässig enthält ein neues Notizbuch eine Zelle - das Rechteck im *TestDrive.ipynb* Notizbuch - aber Sie können weitere hinzufügen.

Links neben der Zelle befindet sich die Notation `[]:`, in der das Jupyter-Notizbuch die Nummer des Codeausschnitts der Zelle anzeigt, nachdem Sie die Zelle ausgeführt haben.

Klicken Sie in die Zelle und geben Sie dann den folgenden Ausdruck ein:

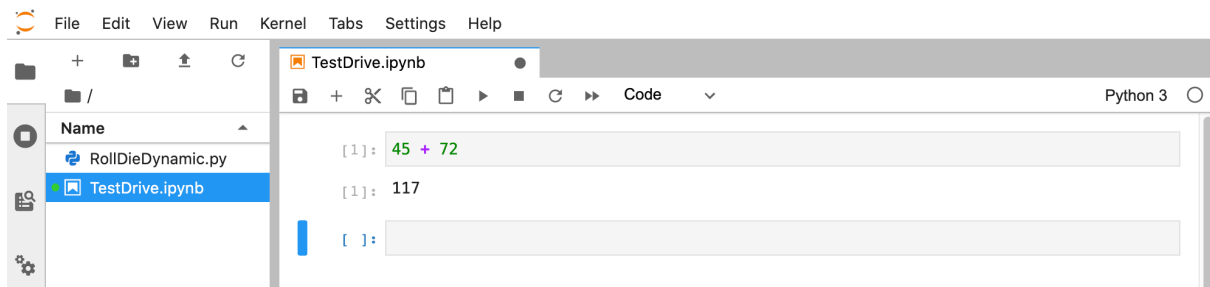
`45 + 72`

Um den Code der aktuellen Zelle auszuführen, geben Sie **Ctrl+Eingabe** (control + enter) ein. JupyterLab führt den Code in IPython aus und zeigt dann die Ergebnisse unterhalb der Zelle an:



4.5. Hinzufügen und Ausführen einer weiteren Zelle

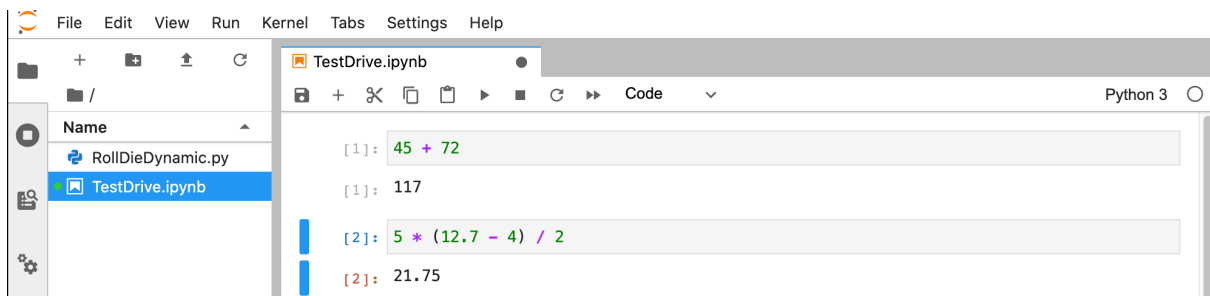
Lassen Sie uns einen weiteren Ausdruck auswerten. Klicken Sie zunächst auf die Schaltfläche + in der Symbolleiste über der ersten Zelle des Notizbuchs. Dadurch wird eine neue Zelle unter der aktuellen Zelle hinzugefügt:



Klicken Sie in die neue Zelle und geben Sie dann den Ausdruck

$5 * (12.7 - 4) / 2$

ein und führen Sie die Zelle aus, indem Sie diesmal **Shift+Eingabe** (shift + enter) eingeben:



Auch diesmal wird das Snippet ausgeführt und das Resultat angezeigt. Zusätzlich wird aber auch eine neue, leere Zelle eingefügt.

4.6. Jupyter Notebook speichern

Wenn Ihr Notizbuch ungespeicherte Änderungen hat, ändert sich das X rechts vom Namen der Registerkarte des Notizbuchs zu ●. Um das Notizbuch zu speichern, wählen Sie entweder in JupyterLab das Menü Datei (nicht oben im Browserfenster) und dann Notizbuch speichern, oder klicken Sie auf das Disketten Symbol in der Symbolleiste.

4.7. Erforschen Sie JupyterLab

Machen Sie sich mit JupyterLab vertraut. Im Menü *Hilfe* finden Sie den Zugang zur JupyterLab Dokumentation (via Menü-Eintrag *JupyterLab Reference*), sowie zu weiteren Dokumentationen und Informationen.

4.8. JupyterLab schliessen

Wenn Sie mit JupyterLab fertig sind, können Sie den Browser, oder die Browser-Registerkarte schliessen und dann im Befehlszeilenfenster (Command-line, Terminal, oder Shell), von der aus Sie JupyterLab gestartet haben, zweimal <Ctrl> + c eingeben.