

Übungsblatt Nr. 4 zur Vorlesung „Einführung in das Rechnergestützte Arbeiten“

Die Übungen sollen auf dem Jupyter-Server der Fakultät bearbeitet werden. Voraussetzung dafür ist nur ein Webbrowser und ein Poolraum-Account.

Der Jupyter-Server ist unter <https://jupytermachine.etp.kit.edu> zu erreichen.

1 Jupyter

Bevor wir uns Python zuwenden ist es hilfreich zuerst die neue Arbeitsumgebung kennenzulernen

- Öffnen Sie mit Ihrem Webbrowser den Link zum Jupyter-Server und melden Sie sich mit Ihrem KIT-Account an. Wenn Sie sich zum ersten Mal verbinden sollten nun **Server Options** abgefragt werden. Wählen Sie **Python** und starten Sie Jupyter.
- Ignorieren Sie vorerst das Hauptfenster mit dem **Launcher**. Daneben sollte das Fenster des **File Browser** geöffnet sein, ist dies nicht der Fall öffnen Sie diesen im Menü auf der linken Seite. Erstellen Sie einen neues Verzeichnis **ERA** und wechseln Sie dahin.
- Auf der Webseite zur Veranstaltung finden Sie ein Jupyter-Notebook mit eine Python-Einführung. Transferieren Sie diese Datei mit **Upload Files** in das **ERA**-Verzeichnis auf dem Jupyter-Server. Öffnen Sie diese Datei öffnet sich ein neues Tab mit der Einführung. Hier können Sie bei der Bearbeitung des Blattes auch immer einzelne Themen nachlesen.
(Aktivieren Sie auf der linken Seite **Table of Contents** wird der **File Browser** durch ein interaktives Inhaltsverzeichnis ersetzt).
- Erstellen Sie nun mit dem Launcher ein erstes Notebook. Dieses Notebook erhält den Namen **Untitled.ipynb** und wird automatisch gespeichert. Benennen Sie es in **blatt4.ipynb** um.
(Beachten Sie, dass im Jupyter-Webinterface die rechte Maustaste häufig ein hilfreiches Kontext-Menü öffnet.)
- Erstellen Sie eine Markdown-Zelle mit der **Überschrift** „Blatt 4“ und eine weitere mit der Unterüberschrift „Aufgabe 2“. (Hilfe zur Markdown-Syntax finden Sie auch unter **Help-> Markdown Reference**

2 Python

Jetzt kommen die ersten Schritte in Python. Die Zellen sollten dabei **Code** sein.

- Weisen Sie in einer Zelle der Variable **a** einen Integer-Wert zu, und geben danach mit **print()** die Zeichenkette “->” sowie das Ergebnis von **5*a** aus.
Wiederholen Sie dies für eine Fließkommazahl und eine Zeichenkette.
- Verwenden Sie in der nächstes Zelle die Funktion für die Bildschirmeingabe. Geben Sie dabei den Text „Eine Zahl: “ aus und weisen Sie die Eingabe der Variable **x** zu. Welchen Typ hat die Variable **x** nach der Eingabe?
- Wandeln Sie die Variable **x** nun in den Datentyp **float** um und testen Sie erneut den Variablentyp.

3 Ein erstes Python-Skript (Teil 1)

Legen Sie ein neues Notebook an und benennen Sie es in „aufgabe3.ipynb“ um.

1. Binden Sie das Modul `math` ein und testen Sie durch die Ausgabe der Variable `math.pi` ob dies funktioniert hat.
2. Lassen Sie eine Zahl „Summationsgrenze:“ eingeben und weisen Sie die Eingabe der Variable `n_max` zu. Wandeln Sie den Wert vor der Zuweisung gleich in den Datentyp `int` um. Geben Sie danach den Inhalt der Variable aus.
Was passiert wenn Sie als Zahl den Text „Hallo“ eingeben?
3. Jetzt kommt eine erste bedingte Anweisung: Testen Sie ob `n_max > 12` gilt, und sollte dies wahr sein, geben Sie die Meldung `“n_max darf nicht größer als 12 sein!”` aus.

`math` bietet eine Funktion zur Berechnung von $\sin(x)$, hier soll dies aber jetzt selbst implementiert werden. Dabei wird die Taylorreihe genutzt:

$$\sin(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

4. Schreiben Sie zuerst ein `for`-Schleife, in der `n` von 0 bis einschliesslich `n_max` durchlaufen wird und geben Sie dabei in jedem Durchlauf `n` aus.
5. Berechnen $y = \sin(\pi/2)$ in dem Sie vor der Schleife $x = \pi/2$ und $y = 0$ setzen und dann innerhalb der Schleife nacheinander die Terme der Taylorreihe zu y addieren. Lassen Sie dabei in der Schleife auch immer den neuen Wert von y ausgeben. (Die Funktion `math.factorial()` berechnet die Fakultät.)
6. Wiederholen Sie dies, lassen Sie die Schleife nun aber umgekehrt, d.h. von `n_max` nach 0 laufen. Ändert sich das Ergebnis? *Hinweis:* Schrittweiten können auch negativ sein.

4 Eigene Funktionen

Diese Berechnung so immer wieder in ein Programm einzubauen ist umständlich.

Definieren Sie eine Funktion `my_sin(x)` die auf diese Weise $\sin(x)$ berechnet und zurück gibt. Setzen Sie dabei in der Funktion `n_max` fest auf den Wert 12. Testen Sie die Funktion in dem Sie `my_sin(x)` für verschiedene Werte berechnen.

Eigene Funktionen werden in Python üblicherweise gleich am Anfang des Skripts nach den `import`-Anweisungen definiert. Verschieben Sie die Zelle mit der Funktionsdefinition entsprechend an den Anfang des Notebooks.