

Numerik I – PYTHON-Vorkurs – 0. Übungsblatt

Aufgabe 1: (Erste Schritte)

Befehle: `type`

Bestimmen Sie mit dem Befehl `type()` den Typ der folgenden Ausdrücke

- | | | |
|--------------|---------------------|---|
| (a) 7 | (d) 'Dagobert' | (g) <code>set([3, 1, 4, 1, 5, 9, 2])</code> |
| (b) 7. | (e) (1, 2, 3) | (h) <code>print('Donald')</code> |
| (c) $7 + 0j$ | (f) [1, 2, "Daisy"] | (i) <code>{0 : 1, 4 : 'D', -3 : [1, 2, 3]}</code> |

Aufgabe 2: („Taschen“rechner)

Überlegen Sie sich die Ergebnisse der folgenden mathematischen Ausdrücke und überprüfen Sie Ihre Ergebnisse mit dem Computer.

Hinweis: Sie können die Ausdrücke nicht direkt in die Eingabezeile tippen, sondern müssen sie zuerst in *Jupyter* Syntax umschreiben.

- | | | |
|-----------------------------------|------------------------------|--------------------------------------|
| (a) $(11, 9 + 8, 1) \cdot 6$ | (c) $(8 + 2i) \cdot (3 - i)$ | (e) $3 + 4 \bmod 5$ |
| (b) $\frac{5}{7} + \frac{2}{0,1}$ | (d) 2^3 | (f) $\frac{5(3-9)}{7+\frac{-15}{3}}$ |

Befehle: `np.cos`, `np.exp`, `import * as np`, `np.log`, `numpy`, `np.sqrt`

Laden Sie nun das Modul `numpy` als `np` und berechnen Sie die folgenden Ausdrücke:

- | | | |
|---------------------------|------------------|-----------------|
| (g) $\cos(\frac{\pi}{2})$ | (h) $e^{\ln(7)}$ | (i) $\sqrt{-1}$ |
|---------------------------|------------------|-----------------|

Verwenden Sie bei (i) sowohl den Potenzbefehl als auch die Funktion `np.sqrt`. Was fällt Ihnen auf? Welchen Typ haben die Ergebnisse in (i)?

Aufgabe 3: (Variablen)

Befehle: `print`

Befolgen Sie die Anweisungen des nachfolgenden Algorithmus um Ihre persönliche Glückszahl zu bestimmen. Beenden Sie jeden Befehl mit einem Semikolon, um die Ausgabe zu unterdrücken.

- Speichern Sie (immer als Zahlen) in der Variable `tag` Ihren Geburtstag, in der Variable `monat` Ihren Geburtsmonat, in der Variable `jahr` die letzten beiden Ziffern Ihres Geburtsjahres und in der Variable `glueckszahl` die Zahl 50.
- Speichern Sie in `monat`: Zweimal `monat` plus fünf.
- Speichern Sie in `jahr`: Zehn hoch vier mal `jahr` minus fünf mal `glueckszahl`.
- Speichern Sie nun in `tag`: Fünfzig mal `monat` plus `tag`.
- Speichern Sie zum Schluss `jahr` plus `tag` in der Variable `glueckszahl`.

In `glueckszahl` ist nun Ihre persönliche Glückszahl gespeichert. Geben Sie diese mit dem Befehl `print` aus. Wieso ist dies „Ihre“ Glückszahl?

Aufgabe 4: (Die Hilfe verwenden)

Befehle: selbst herausfinden

- (a) Erzeugen Sie 69 äquidistante Punkte in $[2, 20]$ mit dem Befehl `np.linspace`.
- (b) Wie lautet der Winkel von $3 + 4i$, d.h. wie lautet $\phi \in [-\pi, \pi]$ für $3 + 4i = r \cdot e^{i\phi}$?
- (c) Erzeugen Sie eine zufällige ganzzahlige Zahl aus dem Intervall $[13, 42]$.

Hinweis: Wenn Sie den gewünschten Befehl nicht kennen, gucken Sie ob ein vielleicht passender englischer Befehl existiert.

Aufgabe 5: (Slicing)

Erstellen Sie folgendes Objekt des Typs `list`:

`a = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]`

Greifen Sie nun mit Hilfe von *Slices* auf folgende Elemente zu:

- (a) Das dritte bis achte Element.
- (b) Die Elemente mit ungeradem Index.
- (c) Das erste, vierte, siebte und zehnte Element.
- (d) Alle Elemente ab dem zweiten.
- (e) Das elfte bis fünfte Element (also rückwärts).
- (f) Das vor-, viert-, sechst- und achtletzte Element.

Aufgabe 6: (*Arrays*)

Befehle: `array`, `cumsum`, `diag`, `ones`, `random`, `range`

Laden Sie als erstes das Modul `numpy` als `np`.

- (a) Erstellen Sie ein `array a` mit den Zahlen 1 bis 5. Verwenden Sie dabei den Befehl `range`.
- (b) Erstellen Sie nun mit dem Befehl `ones` ein `array b` der Dimension 5×1 gefüllt mit Dreien.
- (c) Erstellen Sie außerdem ein `array C` der Dimension 5×5 mit zufälligen natürlichen Zahlen aus $[0, 2]$ mit Hilfe des `np`-Untermoduls `random` (siehe Blatt 1).

Überlegen Sie sich welche Ausgaben folgende Befehle erzeugen und überprüfen Sie Ihre Überlegungen. Wieso erzeugen einige Eingaben eine Fehlermeldung?

- (d) `C.size`
- (e) `a.shape`
- (f) `b.shape`
- (g) `7 * a`
- (h) `a + b`
- (i) `a * C`
- (j) `a + C`
- (k) `np.exp(C)`
- (l) `np.array([a, b])`
- (m) `a + np.diag(C)`
- (n) `np.diag(a)`
- (o) `np.cumsum(a, 1)`

Aufgabe 7: (Zugriffe und Operationen)

Befehle: `imag`, `range`, `real`

Erstellen Sie ein Array `d` mit folgenden beiden Eigenschaften. Geben Sie das Array nicht direkt ein, sondern verwenden Sie Methoden wie `range`. Tippen Sie die Indizes in den folgenden Teilaufgaben nicht direkt ein, sondern verwenden Sie Slices, logische Ausdrücke und ggf. (selbstgeschriebene) Funktionen!

- `np.real(d)=array([-4., -3., -2., -1., 0., 1., 2., 3., 4., 5.])`
- `np.imag(d)=array([49., 36., 25., 16., 9., 4., 1., 0., 1., 4.])`

Greifen Sie nun auf folgende Elemente Ihres Arrays zu:

- Das erste, dritte, vierte und siebte Element (hier dürfen Sie doch direkt indizieren)
- Das zweite bis neunte Element
- Alle Elemente mit geradem Index
- Alle Elemente mit geradem Imaginärteil
- Alle Elemente deren Absolutbetrag des Realteils größer als 2 ist
- Alle Elemente deren Realteil durch 2 teilbar ist oder deren Imaginärteil durch 3 teilbar ist
- Führen Sie die folgenden Befehle aus. Was fällt Ihnen auf?

```
e=np.real(d); print(e)
e1 = e[range(1, 11, 2)]; print(e1)
e1[:] = 0; print(e, e1)
e2 = e[1 :: 2]; print(e2)
e2[:] = 0; print(e, e2)
```

Aufgabe 8: (Funktionen und Plots)

Befehle: `*`, `lambda*`, `*`, `legend`, `linspace`, `plot`

Laden Sie das Plot-Modul mit

```
import matplotlib.pyplot as plt;
```

- Definieren Sie eine Funktion $f(x) = 0,5 + 5 \cdot 10^{-2}x(x + 2,5)(x - 3)$
- Definieren Sie eine Funktion $g(x) = \sin(49x)$
- Plotten Sie beide Funktionen in einen Plot im Intervall $[-\pi, \pi]$ (verwenden Sie dazu den `linspace` Befehl). $f(x)$ soll in einer schwarzen, gestrichelten Linie und $g(x)$ in einer grünen, durchgezogenen Linie geplottet werden.
- Erstellen Sie zusätzlich eine aussagekräftige Legende mit Schatten die sich in der oberen linken Ecke befindet.

Es kann sein, dass Ihr Plot nicht so aussieht wie Sie es erwartet haben. Wieso ist das so und wie können Sie das Problem beheben?

Aufgabe 9: (*Arrays konstruieren*)

Befehle: concatenate, hstack, reshape, T, transpose, vstack

- (a) Erstellen Sie ein array **a** mit den Zahlen von 1 bis 9, ein array **b** mit drei Einsen und ein array **c** mit fünf Einsen.

Wichtig: Lösen Sie die folgenden Teilaufgaben nur mit Ihren Arrays **a**, **b** und **c**, als Ganzes und nicht nur Teile eines Arrays, und den oben genannten oder ähnlichen Befehlen. Jede Teilaufgabe soll ohne speichern von Zwischenergebnissen in nur einer Zeile gelöst werden!

Erstellen Sie folgende arrays:

(b) $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$

(c) $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 \end{pmatrix}$

(d) (1 4 7 2 5 8 3 6 9)

(e) $\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 3 & 1 \\ 1 & 4 & 5 & 6 & 1 \\ 1 & 7 & 8 & 9 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$

Aufgabe 10: (*Subplot & logarithmische Achse*)

Befehle: subplot

Definieren Sie die Funktionen

$$f(x) = \frac{\sin(20 \ln(x + 2))}{x + 2} \quad \text{und} \quad g(x) = e^{-4x}. \quad (1)$$

Zeichnen Sie einen normalen, also einem nicht interaktiven, Plot mit zwei Unterplots mit folgenden Eigenschaften:

- Linker Plot: Graphen von f in $[-1, 2]$ mit einer gestrichelten Linie der Stärke 4 in Rot.
- Rechter Plot mit logarithmischer y-Achse: Graphen von g in $[0, 10]$ in einer gepunkteten Linie in Magenta.
- Jeder Unterplot hat eine aussagekräftige Überschrift.

Diese Aufgaben werden nicht in den Programmierübungen besprochen.