

03c_NumPy_Miniuebungen

0.1 Mini-Aufgaben zur Überprüfung des Verständnis: ndarrays

Legen Sie die folgenden Matrizen mittels `arange()` und `reshape()` an:

$$A = \begin{pmatrix} 1 & 3 & 5 \\ 7 & 9 & 11 \\ 13 & 15 & 17 \end{pmatrix} \quad B = \begin{pmatrix} 17 & 15 & 13 \\ 11 & 9 & 7 \\ 5 & 3 & 1 \end{pmatrix}$$

Hinweis: Dies ist jeweils mit einem Einzeiler möglich.

```
[1]: import numpy as np

# ...
```

Erreichen Sie dasselbe Ergebnis ohne Verwendung von `reshape()`. Hierzu sollten zwei Codezeilen pro Matrix ausreichen.

```
[2]: import numpy as np

# ...
```

Schreiben Sie Code, der mittels `diag()` für eine beliebige Länge der Hauptdiagonale Matrizen der folgenden Form erzeugt (nicht angegebene Einträge sind Nullen):

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}$$

Hinweis: Matrizen können in NumPy mit `+` addiert werden.

```
[3]: import numpy as np

n = 5

# ...
```

Erzeugen Sie mit `linspace()` jeweils mit einem Einzeiler die folgenden Vektoren: * Zahlen von 1 bis 10×10 äquidistante Punkte im Intervall $[0, \pi]$

Hinweis: NumPy stellt die Konstante π natürlich auch bereit. Finden Sie heraus wie, anstatt das Paket math aus der Standardbibliothek zu verwenden.

```
[4]: import numpy as np  
  
# ...
```

0.2 Impressum

0.2.1 Programmierkurs Python, Dominik Götde <https://www.ians.uni-stuttgart.de>,
Universität Stuttgart

Version vom April 2023

Lizenziert unter der Creative Commons Namensnennung 4.0 International Lizenz



Veröffentlicht auf <https://zoerr.de>, (alle Rechte am Logo vorbehalten)



Gefördert durch die Stiftung Innovation in der Hochschullehre. (alle Rechte am Logo vorbehalten)



Gefördert mit Mitteln der Deutschen Forschungsgemeinschaft (EXC 2075 - 390740016) im Rahmen der Exzellenzstrategie.