

Übung 1.1

Lernziele

Die Studierenden...

- wissen, wie man Arrays in Python aus Listen erstellt
- wissen, was formatted String sind und wie man sie benutzt
- können Arrays indizieren
- können Arrays mittels Konsole ausgeben (`print()`) und mit `matplotlib.pyplot` visualisieren (`plt.imshow()`)
- können verschiedene Parameter der Visualisierung einstellen (`cmap`, `vmin`, `vmax`)
- können mit doppelten for-Schleifen durch Arrays iterieren, um einzelne Pixel individuell zu adressieren
- können Funktionen schreiben und in anderen Skripten importieren

Aufgaben

1. Erstellen Sie folgendes 4x4 Array und speichern Sie es in der Variable `my_array`:

| | | | |
|---|---|----|----|
| 1 | 2 | 3 | 4 |
| 2 | 4 | 6 | 8 |
| 3 | 6 | 9 | 12 |
| 4 | 8 | 12 | 16 |

2. Visualisieren Sie das Array und geben Sie für die Visualisierung einen minimalen Wert von 0 und einen maximalen Wert von 16 an.
3. Schneiden Sie aus `my_array` eine 2x2 Matrix aus der Mitte aus und speichern Sie den Ausschnitt in einer neuen Variable `my_array_2`. Benutzen Sie dabei `my_array[Anfang:Ende+1 , Anfang:Ende+1]`.
4. Schreiben Sie eine Funktion, die die Werte eines Arrays quadriert und lagern Sie die Funktion in eine externe .py Datei aus. Geben Sie `my_array` in die Funktion und speichern Sie das Ergebnis in einer neuen Variable `my_array_3`. Visualisieren Sie `my_array_3` und benutzen Sie einen minimalen Wert von 0 und einen maximalen Wert von 255. Vergleichen Sie die Visualisierung von `my_array` und `my_array_3`. Fällt Ihnen etwas auf?
5. Schreiben Sie eine Funktion, die durch ein Array iteriert und jeden Pixel-Wert ausgibt, der grösser als die Hälfte des maximalen Array-Wertes ist. Lagern Sie die Funktion aus und geben Sie `my_array` und `my_array_3` in die Funktion. Wieso werden unterschiedlich viele Werte ausgegeben?