# 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 addressieren
* 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 |

1. Visualisieren Sie das Array und geben Sie für die Visualisierung einen minimalen Wert von 0 und einen maximalen Wert von 16 an.
2. 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**].
3. 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?
4. 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?