# Übung 1.1

## Lernziele:

* Anaconda + Environment kennenlernen
* Variablen kennenlernen (int, list, array)
* print und plt.imshow() als Ausgabe-Möglichkeit kennenlernen
* Was formatted Strings sind und wie man sie benutzt
* Array indizieren und erste Punktoperation
* Array visualisieren und Visualisierungs-Einstellungen kennenlernen
* For-Schleifen und if-Bedingung

## Schritte:

* Anaconda starten
* Environment erstellen
* Spyder installieren
* Spyder Fenster erklären
* print(“Hello world”)
* Liste erstellen
* Variable anschauen
* Element in Liste addressieren
* Numpy und Matplotlib installieren
* Array erstellen
* Print array
* Print array shape
* Print array.min() und array.max()
* Array indizieren
* Array mit plt.imshow() anzeigen
* Cmap definieren
* vmin und vmax definieren
* Array einmal mit Skalar addieren und einmal multiplizieren und plotten
* Funktion zum Subtrahieren schreiben und auslagern in neue Datei
* Schleife schreiben, um auf Array zuzugreifen
* Ausgabe des Wertes, wenn der Wert grösser als ... ist

## Aufgaben

* 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 |

* Visualisieren Sie das Array und geben Sie für die Visualisierung einen minimalen Wert von 0 und einen maximalen Wert von 16 an.
* Schneiden Sie aus “my\_array” eine 2x2 Matrix aus der Mitte aus. Benutzen Sie dabei my\_array[Anfang:**Ende+1** , Anfang:**Ende+1**]. Speichern Sie die den Ausschnitt in einer neuen Variable “my\_array\_2”

Nach den Indizes fragen

* Schreiben Sie eine Funktion, die die Werte von “my\_array” quadriert und lagern Sie die Funktion in eine externe .py Datei. Das Ergebnis dieser Funktion soll in einer neuen Variable “my\_array\_3” gespeichert und visualisiert werden. Benutzen Sie dabei 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?

Mit einer nicht-linearen Operation (quadrieren) kann man Werte “filtern” bzw. hervorheben

* 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.

# Übung 1.2

## Lernziele

* Graubild einlesen
* Linienprofil plotten
* Histogramm plotten
* Berechnung von Dateigrössen

## Schritte

* Graubild einlesen
* Datentyp ausgeben lassen
* Datei-Grösse berechnen
* Histogramm plotten
* Skimage installieren
* Linienprofil plotten

## Aufgaben

* Schreiben Sie eine Funktion, die ein Bild als Input nimmt und in der Konsole folgende Informationen mit **einem formatted String** ausgibt:
  + Dimensionen des Bildes
  + Minimaler Pixelwert
  + Maximaler Pixelwert
  + Datentyp der Pixel
* Lesen Sie das Bild “cameraman.bmp” ein und geben Sie es in ihre soeben geschriebene Funktion ein.
* Visualisieren Sie das Bild und schätzen Sie ein, welche Pixelwerte am meisten vorkommen sollten
* Plotten Sie das Histogramm. Welche Pixelwerte treten tatsächlich am meisten auf?

## Bonus

* Die Funktion skimage.measure.profile\_line() zur Visualisierung des Intensitäts-Profils nimmt 2 Punkte mit ihren x- und y-Koordinaten als Input, was nicht sehr intuitiv ist. Schreiben Sie eine Funktion, die ein Bild einliest und dem Benutzer anzeigt. Wenn der Benutzer in dem Bild an 2 Stellen mit der Maus klickt, soll eine Linie dazwischen gezogen und das Intensitäts-Profil entlang dieser Linie angezeigt werden.

Tipp: Der Befehl, um die Koordinaten auszulesen lautet:

plt.ginput(*n*=number\_of\_clicks, *timeout*=-1)

Recherchieren Sie im Internet, wie die Funktion zu benutzen ist.

* Strukturierung:
  + Bild einlesen
  + Bild anzeigen
  + clicks = plt.ginput(*n*=number\_of\_clicks, *timeout*=-1)
  + Koordinaten aus clicks extrahieren
  + Bei den Koordinaten Punkte einzeichnen lassen mit plt.scatter()
  + Zwischen den Punkten eine Linie zeichnen mit plt.plot()
  + Die Abbildung mit plt.draw() aktualisieren
  + Die Koordinaten in skimage.measure.profile\_line() eingeben und in neuer Abbildung mit plt.plot()visualisieren