

Bildverarbeitung Praktikum

Übung 0

Dr. Christina Gillmann

April 15, 2021

1 Installieren von Python

Python ist eine interpretierte, höhere Programmiersprache. Der Charme dieser Programmiersprache liegt in einem sehr kurzen und gut lesbaren Programmierstil. Auch wenn Python dadurch schlecht wartbar bei großen Projekten ist, ist es eine sehr gute Basis für das Bildverarbeitungspraktikum, da ihnen viel Overhead erspart bleibt. Python hat eine starke Community, weshalb sie i.d.R. zu jedem Problem eine Lösung online finden werden.

Installieren sie Python auf ihrem Rechner. Welche Entwicklungsumgebung sie benutzen ist im Prinzip freigestellt. Empfohlen ist <https://www.spyder-ide.org/>.

Zusätzlich zu Python benötigen sie für dieses Praktikum einige Bibliotheken. Diese können sie durch die Anaconda Shell installieren. Sie benötigen vorerst **numpy** und **matplotlib**.

Die Befehle dazu sind folgende:

```
1 conda install -c anaconda numpy
```

```
1 conda install -c conda-forge matplotlib
```

Nach dem Installieren der beiden Bibliotheken sollten sie in der Lage sein, folgenden Code in Spyder auszuführen. Hierzu müssen sie eine neue Datei anlegen und den Code einfügen. Sie benötigen noch ein Bild, welches im gleichen Ornder abgelegt wird wie der Quellcode (eventuell müssen sie den Namen des Bildes in Zeile 5 anpassen).

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 from PIL import Image
4
5 img = Image.open( 'test.png' )
6
7 data = np.asarray( img, dtype='uint8' )
8
```

```
9 imgplot = plt.imshow(data)
```

2 Erste Schritte in Python

Wenn sie bis jetzt eher C++ oder Java gewohnt sind, kann Python am Anfang etwas befremdlich sein. Betrachten sie folgenden Code und führen sie ihn aus, indem sie eine neue Quellcode Datei anlegen in welcher der folgende Code enthalten ist.

Sie können ein beliebiges Bild für diesen Code verwenden, dass hier als X.png bezeichnet wird.

```
1 matplotlib inline
2 import matplotlib.pyplot as plt
3 import matplotlib.image as mpimg
4 import numpy as np
5 def imag_X(img, n, m=1):
6     if n == 1:
7         tiled_img = img
8     else:
9         lst_imgs = []
10        for i in range(n):
11            lst_imgs.append(img)
12        tiled_img = np.concatenate(lst_imgs, axis=1 )
13    if m > 1:
14        lst_imgs = []
15        for i in range(m):
16            lst_imgs.append(tiled_img)
17        tiled_img = np.concatenate(lst_imgs, axis=0 )
18
19    return tiled_img
20 basic_pattern = mpimg.imread('X.png')
21 decorators_img = imag_X(basic_pattern, 3, 3)
22 plt.axis("off")
23 plt.imshow(decorators_img)
```

```
1 matplotlib inline
2 import matplotlib.pyplot as plt
3 import matplotlib.image as mpimg
4 import numpy as np
5
6 basic_pattern = mpimg.imread('X.png')
7 cropped = basic_pattern[90:150, 50:120]
8 plt.axis("off")
9 plt.imshow(cropped)
```

- a) Was passiert in den beiden Quellcodes?
- b) Welche Parameter können sie jeweils verändern und wie ändert sich dann der Output?
- c) Welche Parameter können in den Quellcodes zu Problemen führen?