



Übungsblatt 2

Willkommen zum Praktikum zu Programmieren in Python.

Aufgabe 1. Extrahieren Sie aus einer Liste der Zahlen von 0 bis 100: die ersten 10 Zahlen, die letzten 10 Zahlen, jede 10.te Zahl (beginnend mit 0), die mittlere Zahl, jede dritte Zahl außer von den ersten vier und von den letzten fünf, jede dritte Zahl aber davon nicht die ersten vier und nicht die letzten fünf.

Aufgabe 2. Weisen Sie der Variablen `name` Ihren Vornamen zu. Ersetzen Sie den zweiten Buchstaben Ihres Vornamens durch ein großes “X”. Verwenden Sie nur Zuweisung, Slicing und Verkettung.

Aufgabe 3. Was ist der Wert von `lis` und `unter` nach Ausführung der folgenden Anweisungen?

```
lis = [1, 2, 3]
lis[len(lis):] = [4]
unter = [5, 6]
lis[1] = unter
unter[0] = 7
```

Zeichnen Sie die resultierenden Strukturen.

Aufgabe 4. Lesen Sie in der Dokumentation nach was die eingebaute Funktion `dir` macht. Welche Methoden haben Listen? Probieren Sie sie aus.

Aufgabe 5. Erstellen Sie ein Dictionary `de2en` für die Übersetzung der Zahlen von eins, zwei und drei ins englische. Verwenden Sie als Schlüssel sowohl die Zahlen als auch die Ziffern. Welche Methoden haben Dictionaries? Probieren Sie sie aus.

Aufgabe 6. Schreiben Sie ein Programm `imagemanip`, das Graustufenbilder manipuliert. Sie verwenden die Python-Bibliothek `PIL.Image` (Hinweis 1), um einfach Graustufenbilder im PGM-Format einzulesen, zu manipulieren und zu schreiben. Es soll möglich sein Bilder aufzuhellen, zu spreizen und zu binarisieren. Das Programm akzeptiert immer vier Parameter: Die Methode, eine Zahl, die Eingabedatei und die Ausgabedatei. Zum Beispiel wendet

```
python3 imagemanip.py gamma 1.4 bilder/a.pgm out.pgm
```

die Methode `gamma` mit Parameter `1.4` auf die Bilddatei `a.pgm` an und erzeugt die Bilddatei `out.pgm`. Im Folgenden steht v für den Helligkeitswert eines Pixels und $V (= 255)$ für den maximal zulässigen Helligkeitswert. Folgende Methoden sollen unterstützt werden:

- `heller`, Parameter $-100 \leq p \leq 100$, `int`: Der Helligkeitswert v jedes Pixels wird um p Prozent des maximalen Helligkeitswerts erhöht: $v \leftarrow v + p\% \cdot V$
- `gamma`, Parameter $0.0 \leq \gamma \leq 10.0$, `float`: Der Helligkeitswert v jedes Pixels wird auf den Bereich $[0, 1]$ normiert mit γ (Gamma) potenziert: $v \leftarrow \lfloor V \cdot (\frac{v}{V})^\gamma \rfloor$
- `spreizen`, Parameter $0 \leq p \leq 100$, `int`: Sei a minimaler und b maximaler Helligkeitswert der Pixel im aktuellen Bild. Bei $p = 100$ werden die Helligkeitswerte der Pixel

gleichmäßig im Bereich $[0, V]$ verteilt: $v \leftarrow V \cdot \frac{v-a}{b-a}$. Falls $p < 100$, dann werden die Grenzen prozentual zwischen 0 und a beziehungsweise b und V verschoben. Die neue Untergrenze ist dann $\lfloor a \cdot (100-p)\% \rfloor$ statt 0 und die neue Obergrenze $b + \lfloor (V-b) \cdot p\% \rfloor$ statt V .

- binarisieren, Parameter $0 \leq s \leq V$, `int`: Wenn der Helligkeitswert v jedes Pixels größer oder gleich s ist wird er auf V gesetzt, ansonsten auf 0: $v \leftarrow \lceil (sgn(v-s) + 1)/2 \rceil$. Im manipulierten Bild gibt es also nur noch schwarze oder weiße Pixel.



In obigen Bildern wurde von links nach rechts auf das Bild `lena.pgm` die folgenden Methoden angewendet: `helliger 20`, `gamma 1.4`, `spreizen 100`, `binarisieren 128`. Testen Sie Ihr Programm mit den Bildern in `bilder`.

Hinweis 1. Wir verwenden die Python Image Library (PIL), um Bilder einzulesen und zu speichern wie folgt:

```
import PIL.Image as Image
img = Image.open("bilder/lena.pgm")
cols, rows = img.size # 512, 512
v = img.getpixel((cols-1, rows-1)) # hole Wert eines Pixel (0..255)
img.putpixel((0,0), v) # setze Pixel von rechts unten nach links oben
img.save("/tmp/komisch.pgm")
```

Das Portable Gray Map (PGM) Format erlaubt es plattformübergreifend Graustufenbilder abzulegen (siehe netpbm.sourceforge.net/doc/pgm.html). Sie erhalten für ein Bild mit Breite b und Höhe h ein Image-Objekt `img`, das in dem Tupel `img.size == (b, h)` sich die Breite und Höhe merkt. V steht für den maximalen Helligkeitswert eines Pixels im konkreten Bild. Bei uns ist V immer 255. Auf die Graustufenwerte kann man lesend mit `img.getpixel((x, y))` zugreifen. Beachten Sie, dass der erste Parameter ein Tupel ist. Schreiben können Sie den Grauwert v eines Pixels mit `img.putpixel((x, y), v)`. Sie können eine Bilddatei im PGM-Format¹ einlesen mit `Image.open(name)`, wobei der Name die Endung `.pgm` haben sollte. Schreiben können Sie ein manipuliertes Image-Objekt `img` mit `img.save(name)`, wobei jetzt der Name die Endung `.pgm` haben muss.

Sie können aus Bildern in anderen Formaten PGM-Bilder machen mit zum Beispiel
`$ convert bild.jpg bild.pgm`.

moodle Python_Barth

¹Leider ist es ein anderes Format als das was Sie im 2. Semester in OOP verwendet haben, also die alten Dateien nicht wiederverwenden!