

## Cleverer Algorithmen programmieren: Strings und Palindrome

15.01.2019

Ein String (oder Zeichenkette) ist eine endliche Folge von Zeichen (wie Buchstaben, Ziffern und Sonderzeichen). Strings sind ähnlich wie Listen von Zeichen (aber nicht gleich), die Syntax ist auch ähnlich. Hier sind einige Code-Beispiele mit Strings:

```
s = "Hallo Jakob"           einen String definieren (geht auch mit einfachen Anführungszeichen)
print s[1]                  zweites Zeichen des Strings (man zählt von null)
print s[-2]                 zweitletztes Zeichen des Strings
print len(a)                die Länge des Strings (die Anzahl der Zeichen)
t = s + "!!!"              zwei Strings verketten
print t                     ☆ zur Aufgabe 1
x = str(12 + 3)             eine Nummer in einen String konvertieren
a = int(x + x)              einen String in eine Nummer konvertieren
print a                     ☆ zur Aufgabe 1
```

```
print ord("B")              Jedes Zeichen hat eigenen ASCII-Code zwischen 0 und 255:
                             den ASCII-Code eines Zeichens finden
print chr(66)               das Zeichen mit bestimmten ASCII-Code finden
```

1) Was drucken die Befehle mit ☆ im Code oben aus? Versuche die Antwort zu finden, ohne den Code im Computer auszuführen. Überprüfe es danach mit Computer.

2) Drucke die ganze ASCII-Tabelle aus. Finde die ASCII-Codes der kleinen und grossen Buchstaben.

3) **Ein Palindrom ist ein String, der vorwärts wie rückwärts gelesen identisch ist.** Schreibe eine Funktion, die entscheidet, ob ein String ein Palindrom ist oder nicht.

```
palindrom('neben') = True
palindrom('ABBA') = True
palindrom('bababa') = False
```

4) Schreibe eine Funktion, die entscheidet, ob ein String ein Palindrom ist oder nicht, aber dabei Leerzeichen, Interpunktion, und Gross-/Kleinschreibung **ignoriert**.

```
satzpalindrom('Sie esse Eis.') = True
satzpalindrom('Sie isst Eis.') = False
satzpalindrom("Regal mit Sirup pur ist im Lager.") = True
satzpalindrom('Trug Tim eine so helle Hose nie mit Gurt?') = True
satzpalindrom('Dieser Satz will ein Palindrom sein, ist aber nicht.') = False
```

5) Schreibe eine Funktion, die einen String erhält und den grössten Substring des Strings findet, der ein Palindrom ist.

```
subpalindrom('mama') = 'mam' (oder auch 'ama')
subpalindrom('aaabbaabbaba') = 'abbaabba'
```