Python cheat sheet 3

1 Numpy

Ein wichtiges `package` (Paket bzw. Bibliothek) heißt `numpy`. Diese Bibliothek muss vor der ersten Benutzung heruntergeladen werden. Sie bietet die Möglichkeit, zufällige Entscheidungen zu treffen und schöne rechteckige Tabellen zu zeichnen (und SEHR viel mehr). Um `numpy` zu benutzen, beginnt man das Programm normalerweise mit:

```
import numpy as np
```

Danach startet man ein Befehl, der aus der `numpy`-Bibliothek stammt, immer mit dem Kürzel `np`. Ein paar Beispiele, die mit zufälligen Entscheidungen zu tun haben:

`np.random.random()`	*gibt eine zufällige Zahl zwischen 0 und 1 (z.B 0.276543) zurück*
`np.random.randint(5, 10)`	*gibt eine zufällige ganze Zahl zwischen (in diesem Fall) 5 und 9 zurück*
`np.random.choice(['Katze', 'Hund'])`	*gibt ein zufälliges Element dieser Liste zurück*

Und ein paar Beispiele mit Tabellen:

`np.full((3, 3), 'X')`	*gibt eine 3x3 Tabelle voll mit 'X'-Zeichen zurück*
`board = np.zeros((8, 8))`	*gibt eine 8x8 Tabelle voll mit 0-Zeichen zurück*
`board[0][2] = 3`	*ändert das Element in der ersten Zeile und dritter Spalte zu 3*

2 If-Sätze

Was unter einem `if`-Satz geschrieben ist, passiert, falls die Bedingung erfüllt ist. Falls nicht, sind die Bedingungen der `elif`-Sätze zu überprüfen. Falls immer noch keine Bedingung erfüllt ist, passiert, was unter `else`-Satz steht. Man kann beliebig viele `elif`-Sätze benutzen - oder gar keinen. Ein `else`-Satz ist auch nicht notwendig. Ein Beispiel:

```
if regen_wahrsch < 0.1:
    print("Heute kein Regen!")
elif 0.1 <= regen_wahrsch < 0.4:
    print("Regen ist unwahrscheinlich.")
elif 0.4 <= regen_wahrsch < 0.8:
    print("Regen ist ziemlich wahrscheinlich.")
else:
    print("Nimm einen Regenschirm mit! Es wird definitiv regnen!")
```

Weitere Beispiele für mögliche Bedingungen:

`if my_number <= 3:`	*falls die Variable my_number kleiner oder gleich 3 ist*
`if 5 > 7:`	*falls nummer 5 größer als 7 ist (passiert natürlich nie)*
`if username != 'Hana':`	*falls die Variable username nicht gleich 'Hana' ist*
`if geld == 0:`	*falls die Variable geld genau gleich 0 ist*
`if stop_game:`	*falls die Variable stop_game gleich `True` ist*

3 Erinnerung - Input und Drucken

Input ermöglicht den Benutzer, mit dem Programm zu interagieren. Manchmal ist es nützlich, die Angabe in eine Zahl zu verändern (z.B. um mit der Zahl später zu rechnen). Dabei ist wichtig, dass bei input es immer als ein String (Text) abgespeichert wird.

```
name = input("Was ist dein Name? ")       speichert die Eingabe als die Variable name
alter = int(input("Wie alt bist du? ")) speichert die Eingabe als eine ganze Zahl unter die Variable alter
print("Mein Name ist "+name+".")          ermöglicht die Werte der Variablen in einem Text zu benutzen
```

4 Schleifen

Oft benutzt man eine `for`-Schleife. Hier kann man drei Parameter auswählen: `range(start, stop, step)`, also Start, Ende und Schrittweite. Dabei wird die letzte ausgegebene Zahl strikt kleiner als die `stop`-Zahl beachtet. Um rückwärts zu gehen, muss man -1 als Schritt angeben.

```
for x in range(9):                    -> 0, 1, 2, 3, 4, 5, 6, 7, 8
for reihe in range(-2, 8, 2):         -> -2, 0, 2, 4, 6
for variable in range(100, 95, -1):   -> 100, 99, 98, 97, 96
```

Wenn man nicht weiß, wie häufig etwas zu wiederholen ist, kann es nützlich sein, stattdessen eine `while`-Schleife zu benutzen. Hier wird die Schleife beliebig lang wiederholt und endet erst wenn eine bestimmte Bedingung erfüllt wird - In diesem Beispiel so lang, bis der Benutzer ein Passwort mit mehr als 5 Zeichen schreibt und der Befehl `break` erreicht ist.

```
while True:
    passwort = input("Bitte schreiben Sie Ihr Passwort: ")
    if len(passwort) < 5:
        print("Das Passwort muss mindestens 5 Zeichen haben!")
    else:
        break
```

5 Erinnerung - Funktionen

Eine Funktion (`function`) ist ein Stück Code, den man an beliebig vielen Stellen benutzen kann. Es kann eine oder mehrere Variablen ((`arguments`)) in Klammern haben und es kann mit dem Befehl `return` etwas zurückgeben. Muss aber nicht. Eine super einfache Funktion wäre z.B.:

```
def print_dreimal_hallo():
    print("Hallo!")
    print("Hallo zum zweiten mal!")
    print("Hallo zum dritten mal!")
```

Wenn man diese Funktion schließlich benutzen will, schreibt man an der Stelle:

```
print_dreimal_hallo()
```

Und eine etwas kompliziertere Funktion mit einer Variable und zwei möglichen Zurückgaben:

```
def ist_durch_2_teilbar(nummer):
    """
    Überpruft, ob die angegebene Nummer durch 2 teilbar ist.
    """
    if nummer%2 == 0:
        return True
    else:
        return False
```

Diese Funktion kann man wieder sehr einfach benutzen, z.B. für die Zahl 17:

```
ist_durch_2_teilbar(17)
```