# Einzelaufgaben

## Parse list

1. Schreibe eine Funktion list\_parse(), die eine Liste von Strings entgegennimmt, diese versucht als Float zu parsen und eine Liste von Floats zurückgibt. Es finden sich auch Strings in der Liste, die das Komma als Dezimaltrennzeichen nutzen, diese müssen auch berücksichtigt werden.
2. Alle nicht zu Float parsebaren Eingabestrings werden in einer Fehlerliste gesammelt. Diese muss die Originalstrings enthalten.
3. Siehe Datei list\_parse.py

Beispiel:

a = ["2", "a3", "0.2", "0,4", "a01", "3", ",", "a.a", "a,"]

Result:

b = [2.0, 0.2, 0.4, 3.0]

errorlist = ['a3', 'a01', ',', 'a.a', 'a,']

Hinweise: nutze Typehints, Try except, Doc-String

## Hot Days

1. Schreibe eine Funktion get\_hot\_work\_days(), die aus der Liste weekday\_temperatures eine neue Liste hot\_work\_days erstellt. In diese neue Liste sollen nur Tage kommen, die nicht am Wochenende sind und Temperaturen größer gleich 30 Grad Celcius hatten (siehe Datei hot\_days.py). Die Einträge in der neuen Liste sollen als Tupel dargestellt werden (Datum, Temperatur)

Ergebnis:

[('2019-07-15', 31), ('2019-07-16', 33), ('2019-07-19', 30), ('2019-07-23', 31)]

Hinweise: diese Aufgabe optional mit list comprehension lösen. Nutze Typehints

# Wetterdaten

Gegeben ist Datei wetterdaten.txt (siehe wetterdaten.txt), die für jede Kalenderwoche Temperaturdaten enthält. Der erste Wert in jeder Zeile ist die Kalenderwoche, zb. 12. Durch einen Doppelpunkt ist er abgetrennt von den Tageshöchstwerten.

**Beispiel für die Kalenderwoche 12 von Montag bis Sonntag:**

12: mo34 tue33 wed22 thu 34 fri33 sat33 sun29

Jeder Tag hat ein Kürzel, zb. mo für Monday, tue für Dienstag, gefolgt von einem Wert, der den Tageshöchstwert definiert

mo34 = Montag 34 Grad

tue33 = Dienstag 33 Grad

Der Wert für die Temperatur ist **immer** zweistellig, zb. 34 für 34 Grad, 09 für 9 Grad, 00 für 0 Grad, -03 für -3 Grad etc.

In der Datei sind Fehler enthalten: an manchen Tagen wird die Zahl nicht direkt an das Kürzel gesetzt, sondern durch ein Leerzeichen davon getrennt. Das passiert sporadisch und zufällig.

thu 34 = Donnerstag 34 Grad

Aufgabe:

Die Datei soll eingelesen und geparst werden. Aus den bereinigten Daten soll ein Dictionary erstellt werden. Schreibe dafür die nötigen Funktionen.

Die Kalenderwoche (als String) dient als Key des Dictionaries, eine Liste mit den numerischen Temperaturen für die Woche als value:

{'12': [34, 33, 22, 34, 33, 33, 29],

'13': [22, 33, 32, 32, 23, 33, 28],

'14': [23, 33, 26, 30, 29, 29, 29]}

Nutze Typehints, List-Comprehensions, doc-String und String-Funktionen (siehe übersicht String-Funktionen)

data\_file = 'wetterdaten.txt'

print(get\_data\_dict(data\_file))

Zuatz: wir können uns nicht verlassen, dass die Datei wetterdaten.txt heisst. Schreibe für den Fall, dass der Dateiname nicht stimmt, eine Exception. In diesem Fall gibt die Funktion ein leeres Dict zurück

data\_file = 'wetterddaten.txt'

pprint(get\_data\_dict(data\_file))

{}