

# Python für Linguisten

Dozentin: Wiebke Petersen & Co-Dozentin: Esther Seyffarth

5. Foliensatz  
sequentielle Datentypen, Dictionaries

# Sequentielle Datentypen

- Tupel, Listen und Strings sind sequentielle Datentypen.
- Tupel: (a,b,c,d), Liste: [a,b,c,d], String: "abcd"
- Für alle sequentiellen Datentypen funktioniert:
  - Indexing (`seq_data[2]`)
  - Slicing (`seq_data[2:4]`)
  - Konkatenation (`seq_data + seq_data`)
  - Wiederholung (`seq_data * 3`)
  - Längenbestimmung (`len(seq_data)`)
  - Test auf Enthaltensein (`'a' in seq_data`).
  - Iteration in einer for-Schleife (`for 'a' in seq_data`).

# veränderliche / unveränderliche Datentypen

- Tupel und Strings sind unveränderliche Datentypen
- Listen sind veränderlich, sie können verändert werden durch:
  - Zuweisung neuer Elemente: `my_list[1] = 'x'`
  - Verwendung spezieller Methoden auf Listen (`help(list)`):
    - `append`: `my_list.append('y')`
    - `pop`: `my_list.pop()`
    - `reverse`: `my_list.reverse()`
    - `sort`: `my_list.sort()`
    - `extend`, `insert`, `remove`
- Vorsicht: Bei der Veränderung einer Liste ändert sich nicht ihr Speicherort, dies hat Auswirkungen auf die Evaluation von Variablen:

```
>>> x = 3
>>> y = x
>>> x = 5
>>> y
3
```

```
>>> x = [1,2,3,4]
>>> y = x
>>> x.reverse()
>>> y
[4, 3, 2, 1]
```

# Listen-Abstraktion (list comprehension)

- Elegante Methode zur Erzeugung von Listen:  
`[c for c in 'dies ist ein Satz']`
- Funktioniert auch mit zusätzlicher Bedingung:  
`[c for c in 'dies ist ein Satz' if c in 'aeiou']`
- Syntax ähnlich zur impliziten Mengendefinition in der Mathematik:  
`[x**2 for x in range(10) if x % 3 == 0]`  
 $\{x^2 \mid x \in [0, \dots, 10] \text{ and } x \bmod 3 = 0\}$

# Dictionaries

- Dictionaries bestehen aus Schlüssel-Wert-Paaren:  

```
>>> De_Nl = {"Haus": "huis", "Tasche": "tas", "Fahrrad": "fiets"}
```
- Die Datentypen der Werte sind beliebig. Die Schlüssel müssen unveränderlich sein.
- Der Zugriff auf die Elemente der Dictionaries erfolgt über die Schlüssel:

```
>>> De_Nl['Haus']  
"huis"
```

```
for x in De_Nl:  
    print(x)
```

```
Haus  
Tasche  
Fahrrad
```

```
for x in De_Nl:  
    print(De_Nl[x])
```

```
huis  
tas  
fiets
```

# Dictionaries in Listen umwandeln und umgekehrt

- Dictionary in Liste:

- `>>> De_Nl.items()`  
`dict_items([('Tasche', 'tas'), ('Fahrrad', 'fiets'), ('Haus', 'huis')])`

- `>>> De_Nl.keys()`  
`dict_keys(['Tasche', 'Fahrrad', 'Haus'])`

- `>>> De_Nl.values()`  
`dict_values(['tas', 'fiets', 'huis'])`

- **Achtung:** Dies sind keine Python-Listen, wie wir sie kennen. Wir können auf diese Objekte aber jeweils die Funktion `list(...)` anwenden, um sie zu einer Liste umzuwandeln.

- Liste in Dictionary:

- `frucht = ['Apfel', 'Orange', 'Zitrone']`  
`farbe = ['gruen', 'orange', 'gelb']`  
`ff = dict(zip(frucht, farbe))`

# Übungsaufgaben (einfacher)

- Erstellen Sie ein dictionary, das deutsche Wörter ihren englischen Übersetzungen zuordnet.
- Versuchen Sie, die Einträge ihres Dictionarys alphabetisch sortiert auszugeben.
- Schreiben Sie ein Programm, das nach einem Wort in Deutsch fragt und die Übersetzung liefert.
- Schreiben Sie ein Programm, das sowohl Wörter von Deutsch nach Englisch als auch umgekehrt übersetzt.
- Erstellen Sie ein dictionary, das deutsche Wörter ihren französischen Übersetzungen zuordnet.
- Schreiben Sie ein Programm, das die Übersetzung zwischen allen drei Sprachen anbietet (verwenden sie nur die beiden dictionaries).

# Übungsaufgabe (schwieriger)

- Das CMU Pronouncing Dictionary ist ein maschinenlesbares Wörterbuch von 134.000 englischen Wörtern, denen ihre phonologische Transkription zugeordnet wird.
- Ihre Aufgabe ist es, ein Programm zu schreiben, das einem Benutzer bei Eingabe eines englischen Wortes die Transkription des Wortes ausgibt.
- Das Dictionary können Sie sich unter `http://svn.code.sf.net/p/cmuspinx/code/trunk/cmudict/cmudict.0.7a` herunterladen.
- Ihr Programm soll die heruntergeladene Datei öffnen und die Einträge der Datei (ein Wort pro Zeile) in einem Python-Dictionary speichern.
- Bei der Benutzereingabe soll das eingegebene Wort aus dem Python-Dictionary herausgesucht und der entsprechende Wert zurückgegeben werden.
- Achtung: Die Datei beginnt mit einem Header, den Sie nicht verwerten können. Die Kommentarzeilen sind durch die Zeichensequenz `;;;` markiert. Filtern Sie diese Zeilen beim Einlesen der Datei heraus.
- Wenn Sie zunächst nur Teile der Datei verarbeiten, z.B. die ersten 100 Zeilen, ist das Programm einfacher zu testen.