

Collections und Funktionen

Claas de Boer, Tilman Hinnerichs

26.11.2020

Python-Grundlagen

Was zuletzt geschah...

- Listen und wo sie zu finden sind (`help(...)`, `[]`, `L.append(...)`)
- for-Schleifen (`for i in range(10):`)
- Standarddatentypen und Umwandlung dieser (`int("42")`)
- Eingabe und Ausgabe (`input(SZahlen, bitte:)`)
- Strings (`äbc"+ "def", 'abc'[:-1]`)

1. Übung macht den Meister

2. Dictionaries

Übung macht den Meister

Schreibe ein Programm, welches

1. alle geraden Zahlen einer Zahlenliste in einer zweiten Liste speichert und stoppt, sobald eine 237 vorkommt
2. alle Elemente einer Liste von Strings zusammenfügt und ausgibt
3. als Input den Namen und Heimatplaneten des Nutzers abfragt und diesen nett grüßt
4. die Seitenlänge h und zugehörige Höhe h_c eines Dreiecks einliest und den Flächeninhalt ausgibt
5. die Quersumme einer Zahl berechnet

Dictionaries

Nachtrag zu Listen: List comprehensions

```
1 L = []  
2 for num in list_of_numbers:  
3     if num%2 == 0:  
4         L.append(num+1)
```

```
1 L = [num+1 for num in list_of_numbers if num%2==0]
```

Tupel und Mengen

tuple Listen mit festgesetzter Länge

- kein **append**, etc.
- normaler Zugriff und Veränderung einzelner Elemente wie bei lists

set mathematische Menge

- keine Duplikate
- ungeordnet

```
1 >>> a=(1,2)
2 >>> a[1]
3 2
4 >>> s = {1,2,3,1}
5 >>> s
6 {1,2,3}
7 >>> s[0]
8 Error
9 >>> {1,2,3} & {2,3,4}
10 {2,3}
11 >>> {1,2,3} | {2,3,4}
12 {1,2,3,4}
13 >>> set(range(20))
14 ...
```


Dictionaries sind Mengen von key-value-pairs mit folgenden Eigenschaften

- unordered
- changeable
- ohne Duplikate (bezogen auf keys)

Dictionaries

```
1 thisdict = {  
2     "brand": "Ford",  
3     "model": "Mustang",  
4     "year": 1964  
5 }  
6 print(thisdict)  
7 print(thisdict['brand'])  
8  
9 thisdict['brand']='Trabant'  
10 thisdict['color'] # ERROR  
11 thisdict.get('color')
```

Mengen erlauben keine Duplikate:

```
1 thisdict = {  
2     "brand": "Ford",  
3     "model": "Mustang",  
4     "year": 1964,  
5     "year": 2020  
6 }  
7 print(thisdict)
```

Wie man über dicts iteriert

dicts haben Methoden wie

- `d.items()`
- `d.keys()`
- `d.values()`

```
1 for key, value in d.items():  
2     d[key] = value+1  
3  
4 # dict comprehensions  
5 d = {key: value+1 for (key,value) in d.items()}
```

Rückblick: Typübersicht

Name	Funktion
<code>int</code>	Ganzzahl "beliebiger" Größe
<code>float</code>	Kommazahl "beliebiger" Größe
<code>str</code>	Zeichenkette
<code>bool</code>	Wahrheitswert (True , False)
<code>list</code>	gewöhnliche Liste
<code>tuple</code>	unveränderbares n-Tupel
<code>set</code>	(mathematische) Menge von Objekten
<code>dict</code>	Hash-Map

Schreibe ein Python Programm, welches

1. zwei dicts zusammenfügt
2. ein dict nach den values sortiert
3. alle geraden Einträge eines dicts löscht
4. den größten und kleinsten value einer Liste ausgibt
5. den größten und kleinsten value eines dicts ausgibt
6. ein dict nach values größer 170 filtert
7. welches die Überschneidungen zweier dicts ausgibt