Collections

Claas de Boer, Tilman Hinnerichs

26.11.2020 und 03.12.2020

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]`,)

Gliederung

Übung macht den Meister

Einige Aufgaben™4.0

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

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

```
>>> a=(1,2)
>>> a[1]
2
>>> s = {1,2,3,1}
>>> s
{1,2,3}
>>> s[0]
Error
>>> {1,2,3} & {2,3,4}
{2,3}
>>> {1,2,3} | {2,3,4}
{1,2,3,4}
>>> set(range(20))
...
```

Einige Aufgaben™5.0

Schreibe ein Programm, welches

1. die Liste der Quadrate der Zahlen von 1 bis 20 erzeugt, wenn diese Zahlen kleiner als 5 oder größer als 14 sind (mit List comprehensions)
2. eine gegebene, **geordnete** Liste von Zahlen von Duplikaten befreit

Bonus: eine gegebene, **ungeordnete** Liste von Zahlen von Duplikaten befreit

Dictionaries

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

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

Dictionaries

```
thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
}
print(thisdict)
print(thisdict['brand'])

thisdict['brand']='Trabant'
thisdict['color] # ERROR
thisdict.get('color')
```

Mengen erlauben keine Duplikate:

```
thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964,
    "year": 2020
}
print(thisdict)
```

Wie man über dicts iteriert

dicts haben Methoden wie

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

```
for key, value in d.items():
    d[key] = value+1

# dict comprehensions
d = {key: value+1 for (key,value) in d.items()}
```

Rückblick: Typübersicht

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

Einige Aufgaben™6

Schreibe ein Python Programm, welches

1. zwei gegebene dicts zusammenfügt
2. welches die Überschneidungen zweier gegebener dicts ausgibt
3. ein gegebenes dict nach den values sortiert
4. alle geraden Einträge eines gegebenen dicts löscht
5. den größten und kleinsten value einer gegebenen Liste ausgibt
6. den größten und kleinsten value eines gegebenen dicts ausgibt
7. ein gegebenes dict nach values größer 170 filtert (Nur Paare mit $value > 170$ verbleiben)