Und noch ein Iterable: Mengen (Sets)

- Ursprung: Mathematik (-> Mengenlehre)
- Mengen haben keine Ordnung
  - Und daher auch keinen Index!
- Jedes Element ist einzigartig (keine Duplikate)
- Mengen sind veränderbar (hinzufügen & entfernen)
- Erlaubte Datentypen: Alles außer Listen, Mengen (& Dictionaries)
  - Generell: Keine Iterables mit veränderbaren Werten
- Syntax: geschweifte Klammern

```
my_set = {1, 2.0, "abc", (100, True)}
```

Sets erlauben keine Duplikate

- Duplikate werden automatisch entfernt, keine Fehlermeldung

```
my_set = {1, 2, 3, 3, 2, 3}
print(my_set)        # Ausgabe: {1, 2, 3}
```

- Anwendungsfall: Prüfen, ob Liste Duplikate hat:

```
x = [1, 2, 3, 1]

if len(x) == len(set(x)):
    print("Keine Duplikate")
else:
    print("Liste hat Duplikate")
```

Leere Sets erstellen? Tricky

```
my_set = {}
print(type(my_set))          # Ausgabe: <class 'dict‘>

# so geht’s richtig:
my_set = set()
print(type(my_set))          # Ausgabe: <class 'set'>
```

Sets verändern: Elemente hinzufügen

- Es gibt keinen Index!
- Es gibt keine Ordnung!
- Neue Elemente hinzufügen:

```
my_set = {1, 2, 3}

my_set.add(4)           # Einzelne Elemente mit .add() hinzufügen
my_set.update([5, 1])   # Mehrere Elemente mit .update() hinzufügen
print(my_set)           # Ausgabe: {1, 2, 3, 4, 5}
```

Sets verändern: Elemente entfernen

- Verschiedene Wege zum Entfernen:

my_set = {1, 2, 3, 4, 5}

- Entfernen mit .discard()

```
my_set.discard(3)        # Element 3 wird entfernt
my_set.discard(6)        # Nichts passiert!
```

- Entfernen mit .remove()

```
my_set.remove(2)         # Element 2 wird entfernt
my_set.remove(6)         # Fehlermeldung! KeyError
```

- Fazit: .discard() nutzen (oder .remove() inkl. vorherigem Check)

Sets verändern: Elemente entfernen

- Auch hier gibt es

.pop() # Entfernt *zufälliges* Element (weil es keinen Index gibt)

.clear() # Entfernt alle Elemente

Anwendungsfälle von Sets

- Meistens: Prüfen, ob es Duplikate in einer Liste gibt
- Generell: Überall, wo man keine Duplikate haben will
  - Inventar
  - Playlisten
  - ToDo-Listen
  - Teams
  - etc.

Übungen

- Siehe 13 – Hausi.py