



# Mengen

# Mengen



## Mengen / Sets

- nicht sequentiell
- mutabel
- nur immutable Elemente

# Mengen



## Mengen / Sets

- { }
- Erzeugung auch mit set() aus iterable

```
>>> set( ( 1, 2, 3 ) )  
{1, 2, 3}
```

- Darstellung mit { }

# Mengen



## Ausnahme leere Menge

```
>>> { 1, 2, 3 }  
{1, 2, 3}
```

```
>>> type( {} )  
<class 'dict'>
```

- nur mit `set()`

# Mengen



## Operationen

- `len()`
- `in`
- `not in`
- `|`

alle wie bei Dictionary

# Mengen



## Operationen

- $<$
- $<=$
- $>$
- $>=$

```
>>> { 1,2 } < { 1,2,3 }  
True
```

```
>>> { 1,2 } < { 1,2 }  
False
```

```
>>> { 1,2 } <= { 1,2 }  
True
```

## Teilmenge / Obermenge

# Mengen



## Operationen

- & Schnittmenge
- - Differenz
- ^ symmetrische Differenz

```
>>> { 1,2 } & { 2,3 }  
{2}
```

```
>>> { 1,2 } - { 2,3 }  
{1}
```

```
>>> { 1,2 } ^ { 2,3 }  
{1, 3}
```

# Mengen



immutable Version:

frozenset

- alle bisherigen Operationen
- auch gemischt

```
>>> frozenset( (1, 2) )  
frozenset({1, 2})
```



# Mengen



iterabel

- Reihenfolge nicht definiert

```
1 menge = {1, 100, "a", 0.5}
2 for element in menge:
3     print(element)
```

Ausgabe:

```
a
1
100
0.5
```

# Mengen



## Methoden

- `s.issubset(t)`  $\leq$
- `s.issuperset(t)`  $\geq$
- `s.union(t)`  $|$
- `s.intersection(t)`  $\&$
- `s.difference(t)`  $-$
- `s.symmetric_difference(t)`  $\wedge$

# Mengen



## Methoden

- `s.isdisjoint(t)`
- einzige Mengenoperation ohne Operator

# Mengen



Methoden nur für mutable

- `s.add(e)`
- `s.discard(e)`
- `s.remove(e)` Fehler, wenn nicht vorhanden

# Mengen



## Aufgabe

- Erzeugen Sie die Menge  $\{ 1, 2, 3, 4 \}$ .
- Fügen Sie die Elemente 5 und 6 hinzu.
- Entfernen Sie die Elemente 3 und 5.

# Mengen



immutable Mengen

frozenset

```
>>> frozenset( [1,2,3] )
frozenset({1, 2, 3})

>>> frozenset()
frozenset()

>>> frozenset(1)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'int' object is not iterable
```

# Mengen



- Welcher Fehler wird ausgelöst, wenn man `add()` auf ein frozenset anwenden will?

# Mengen



- Welcher Fehler wird ausgelöst, wenn man `add()` auf ein `frozenset` anwenden will?
- Kann man eine Menge als Schlüssel in einem Dictionary verwenden?



# Mengen



Schreiben Sie eine Funktion `mengenrelationen(m1,m2)`, die

- als Argumente zwei Mengen nimmt
- als Rückgabewert ein Dictionary hat, in dem "<", "<=", ">", "&" als Schlüssel vorkommen und als Wert der Wahrheitswert für die jeweilige Operation zwischen den beiden Mengen

```
{ "<" : True, "<=" : True, ">" : False, "&" : {1,2} }
```