

1. Du kannst aus dem Umstand, dass die Elemente von Mengen einfach und ungeordnet auftreten, die nötigen Konsequenzen für die Darstellung von Mengen in Python ableiten.

2. Du bist in der Lage, Mengen mit

- `{<e1>, <e2>, <e3>, ...}`
- `set(<list>)`
- `set(<string>)`

aus einzelnen Elementen, aus Listen oder Zeichenketten zu erzeugen und kannst die resultierende Datenstruktur (bis auf Reihenfolge) beschreiben.

3. Du kannst die leere Menge mit `{}` oder `set()` erzeugen.

4. Du bist in der Lage, die folgenden Python-Ausdrücke für geeignete Mengen  $A$ ,  $B$ , und Elemente  $e$  auszuwerten.

- `A.union(B)` ( $A \cup B$ )
- `A.intersection(B)` ( $A \cap B$ )
- `A.difference(B)` ( $A \setminus B$ )
- `A.add(element)` ( $A \cup \{e\}$ )
- `A.discard(element)` ( $A \setminus \{e\}$ )
- `A.issubset(B)` (gilt  $A \subset B$  ?)
- `A.isdisjoint(B)` (gilt  $A \cap B = \{ \}$  ?)

5. Du kannst Python-Ausdrücke der Form `len(set)`, `sum(set)`, `sorted(M)` für geeignete Mengen  $M$  interpretieren.

6. Du kannst den Operator `in` im Zusammenhang mit Mengen interpretieren:

`5 in {1,2,3} => False`

`5 in {3,4,5} => True`

7. Du kannst mit einer `for`-Schleife über Mengen iterieren.