

Recap

- Was sind Listen?
- Welche der folgenden Aussagen sind wahr?
 1. Die Länge einer Liste ist nicht veränderbar
 2. Einer Liste, die nur aus Integer-Werten besteht, dürfen keine String-Werte hinzugefügt werden
 3. Die Länge einer Liste wird in Zentimeter angegeben
 4. Listen dürfen keine Listen beinhalten
 5. Für die Liste `x = [1, 2, 3]` gibt `print(x[1])` den Wert 2 aus
 6. Listenelemente lassen sich mit der Methode `.add()` hinzufügen
 7. Die Methode `.pop()` entfernt das erste Element
- Mehr: 08 – Recap.py

Hausaufgaben

- 07 – Hausi.py

Wie werden Listen genutzt?

- Sehr flexibel: Veränderbar, Erlauben Duplikate, verschiedene Datentypen
- Eigene Ordnung: Index
- Beispiel: Platzierungen

```
league_rankings = ["Manchester City", "Manchester United", "FC Liverpool"]
```

```
print(league_rankings[0])      # Gibt ersten Platz aus
```

```
print(league_rankings[-1])     # Gibt letzten Platz aus
```

- Beispiel: Zahlen-“Übersetzer“

```
nr_translator = ["Zero", "One", "Two"]
```

```
print(nr_translator[0])        # Ausgabe: Zero
```

```
print(nr_translator[1])        # Ausgabe: One
```

Wie werden Listen genutzt?

- Generell: Überall, wo mehrere Werte in einer Variable gespeichert werden

```
countries = ["Germany", "Spain", "France"]
```

```
random_numbers = [432, 93, 2452]
```

```
grades_of_student = [1, 3, 1, 1, 5]
```

Übung

- Funktion, die 2 Elemente aus einer Liste vertauscht
- Siehe 08 – Übung.py

Listen in Listen

- Bekannt: Elemente einer Liste können Listen sein
- „Multidimensionale“ Listen? Eher eine Art Tabelle
 - außer Listen in Listen in Listen in ... -> sowas vermeiden!
- Beispiel:

`my_list = [[23, 91], [45, 45, 45], [76, 23]]`

	A	B	C
1	23	45	76
2	91	45	23
3		45	

Listen in Listen

```
my_list = [[23, 91], [45, 45, 45], [76, 23]]
```

	A	B	C
1	23	45	76
2	91	45	23
3		45	

my_list[0] ist die **erste Spalte** (A)

my_list[1] ist die **zweite Spalte** (B)

my_list[2] ist die **dritte Spalte** (C)

my_list[0][0] ist das **erste Element** der **ersten Spalte**: 23

my_list[0][1] ist das **zweite Element** der **ersten Spalte**: 91

my_list[2][0] ist das **erste Element** der **dritten Spalte**: 76

Listen in Listen - Fazit

- `my_list = [[23, 91], [45, 45, 45], [76, 23]]`

`my_list[0]` # [23, 91]

`my_list[1]` # [45, 45, 45]

`my_list[2]` # [76, 23]

`my_list[0][0]` # 23

`my_list[0][1]` # 91

usw...

Listen in Listen - Fazit

- Andere Syntax möglich:

```
my_list = [[23, 91], [45, 45, 45], [76, 23]]
```

```
my_list = [[23, 91],  
           [45, 45, 45],  
           [76, 23]]
```

- Achtet auf eckige Klammern und Kommas!

Hausaufgabe

- Entwickle eine Funktion, die für ein Tic Tac Toe-Spiel den Gewinner ermittelt
- Ein Spiel ist als Liste in Listen dargestellt, z.B.

```
# Spieler 1 ist "x"  
# Spieler 2 ist "o"  
# leeres Feld ist ""  
game = [["x", "o", ""],  
        ["o", "x", ""],  
        ["o", "x", "x"]]
```

- Hier gewinnt Spieler 1
- Regeln: <https://de.wikipedia.org/wiki/Tic-Tac-Toe>