

# Recap

- Warum nutzen wir Schleifen?
- Welche Schleifenarten gibt es?
  - Worin unterscheiden sich diese?
- 09 – Hausi.py
- Fragen zu 09 – Praxisbeispiel.py ?

# Schleifen können noch mehr!

- Schleifen frühzeitig beenden mit **break**
- Beispiel:

```
for i in range(10):  
    if i == 5:  
        break  
    print(i)
```

- Schleifendurchläufe überspringen mit **continue**
- Beispiel:

```
for i in range(10):  
    if i == 5:  
        continue  
    print(i)
```

# Schleifen können noch mehr!

- Code ausführen, wenn die Schleife fertig ist mit **else**
- Beispiel:

```
for i in range(5):  
    print(i)  
else:  
    print("Endlich fertig!")
```

- Achtung: Der else-Block wird nicht ausgeführt, wenn die Schleife mit `break` beendet wurde!

# Schleifen in Schleifen: Nested Loops

- Grundsatz: Führe für jeden Schleifendurchlauf der *äußeren* Schleife die komplette (!) *innere* Schleife aus (sofern nicht explizit ausgeschlossen)

- Beispiel:

```
adj = ["red", "big", "tasty"]
```

```
fruits = ["apple", "banana", "cherry"]
```

```
for i in adj:
```

```
    for j in fruits:
```

```
        print(i, j)
```

<u>print(i, j)</u>	<u>i</u>	<u>j</u>	
red apple	red	apple	
red banana		banana	
red cherry		cherry	<i>innere Schleife beendet</i>
big apple	big	apple	
big banana		banana	
big cherry		cherry	<i>innere Schleife beendet</i>
tasty apple	tasty	apple	
tasty banana		banana	
tasty cherry		cherry	<i>innere &amp; äußere Schleife beendet</i>

# Übung

- Siehe 10 – Übung.py

# Hausaufgabe

- Python bietet für Listen eine sehr nützliche Funktion an: `enumerate()`
- Hausaufgabe: Finde mehr über diese Funktion raus!
- Anlaufstellen: offizielle Python Dokumentation, w3schools, die Suchmaschine deiner Wahl...
- Folgende Sachen gilt es herauszufinden:
  - Was macht die `enumerate()`-Funktion?
  - Welche Parameter akzeptiert sie?
  - Was gibt sie aus?
  - BONUS: Was sind mögliche Use Cases?