

List Comprehensions

List Comprehension bezeichnet eine vereinfachte Schreibweise zur Bearbeitung von Listen. Normalerweise verwendet man eine **for Schleife** um Listen zu verarbeiten.

```
>>> liste_a = range(10)
```

```
>>> liste_a  
range(0, 10)
```

```
>>> liste_b = []
```

```
>>> for x in liste_a:  
    liste_b.append(x+1)
```

```
>>> liste_b  
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

List Comprehensions

Das Schema von List Comprehension

Ausgabewert -- alle Eingabewerte aus der Quelle holen

[x+1 for x in liste_a]

```
>>> liste_a = range(10)    # Liste mit den Quelldaten
```

```
>>> liste_c = [x+1 for x in liste_a]
```

```
>>> liste_c  
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

List Comprehensions

Der Ausgabewert kann auch eine Liste oder ähnliches sein.

```
>>> liste_d = [[x, x+1] for x in liste_a]
```

```
>>> liste_d
```

```
[[0, 1], [1, 2], [2, 3], [3, 4], [4, 5], [5, 6], [6, 7],  
[7, 8], [8, 9], [9, 10]]
```

```
>>>
```

List Comprehensions

Probiert es selbst mal aus.

Erstellt eine Liste die Listen mit dem Durchmesser, dem Umfang und der Fläche von z.B. Drähten erzeugt.

$$\begin{aligned}\text{Umfang} &= d * 3.14 \\ \text{Fläche} &= d^{**2} * 3.14 / 4\end{aligned}$$

List Comprehensions

Die Lösung:

```
>>> liste_fläche = [[x, x * 3.14, x**2*3.14/4] for x in liste_a]
```

```
>>> liste_fläche
```

```
[[0, 0.0, 0.0], [1, 3.14, 0.785], [2, 6.28, 3.14], [3, 9.42,  
7.065], [4, 12.56, 12.56], [5, 15.7, 19.625], [6, 18.84, 28.26],  
[7, 21.98, 38.465], [8, 25.12, 50.24], [9, 28.26, 63.585]]
```

List Comprehensions

Auch Bedingungen sind in List Comprehensions möglich:

```
[x for x in range(10) if x % 2 == 0]
```

```
>>> [x for x in range(10) if x % 2 == 0]
```

```
[0, 2, 4, 6, 8]
```

```
>>>
```

List Comprehensions

Es ist auch ein **else** Zweig möglich. Allerdings muss die if-else-Bedingung dann in das Feld des Ausgabewertes geschrieben werden:

```
[x*x if x % 2 == 0 else x+1 for x in range(10)]
```

```
>>> [x*x if x % 2 == 0 else x+1 for x in range(10)]
```

```
[0, 2, 4, 4, 16, 6, 36, 8, 64, 10]
```

```
>>>
```

List Comprehensions

Nun habt Ihr eine ganze Menge zum Spielen.