

# List Comprehensions und Dict Comprehensions in Python

---

## Einführung

List Comprehensions und Dict Comprehensions sind leistungsstarke Konstrukte in Python, die es ermöglichen, Listen und Dictionaries auf kompakte und elegante Weise zu erstellen. Sie bieten eine alternative Schreibweise zu traditionellen Schleifen und bieten oft eine bessere Lesbarkeit und kürzere Codezeilen.

## List Comprehensions

List Comprehensions ermöglichen es, Listen auf eine einzige Zeile Code zu erstellen, indem eine kompakte Syntax verwendet wird.

### Syntax

```
liste = [ausdruck for element in iterable if bedingung]
```

### Beispiel

```
quadratzahlen = [x ** 2 for x in range(5) if x % 2 == 0]  
print(quadratzahlen) # Ausgabe: [0, 4, 16]
```

## Dict Comprehensions

Dict Comprehensions funktionieren ähnlich wie List Comprehensions, ermöglichen jedoch die Erstellung von Dictionaries.

### Syntax

```
dictionary = {schlüssel: wert for element in iterable if bedingung}
```

### Beispiel

```
wochenendtage = {'Montag': 'Arbeit', 'Dienstag': 'Arbeit', 'Mittwoch': 'Arbeit',  
                 'Donnerstag': 'Arbeit',  
                 'Freitag': 'Arbeit', 'Samstag': 'Wochenende', 'Sonntag':  
                 'Wochenende'}  
umgekehrte_wochenendtage = {tag: typ for tag, typ in wochenendtage.items() if typ  
                             == 'Wochenende'}
```

```
print(umgekehrte_wochenendtage) # Ausgabe: {'Samstag': 'Wochenende', 'Sonntag':  
'Wochenende'}
```

## Verwendungszwecke

- List Comprehensions und Dict Comprehensions sind nützlich, um Listen und Dictionaries schnell und lesbar zu erstellen.
- Sie sind ideal, wenn Sie eine kompakte Schreibweise für das Erstellen von Datenstrukturen benötigen.