

PEP 8 - Style Guide

Einleitung

PEP 8 (Python Enhancement Proposal Nr. 8) ist der offizielle Style-Guide für Python und dient als Empfehlung, um den Code einheitlich und gut lesbar zu gestalten. Gerade in größeren Projekten oder im Teamalltag ist eine klare und konsistente Struktur essenziell, da Code von mehreren Personen gelesen, überprüft und erweitert wird. PEP 8 gibt dafür konkrete Richtlinien vor – von der Einrückung und Zeilenlänge bis hin zu Namenskonventionen und Kommentarstrukturen. Indem wir uns an diesen Leitfaden halten, stellen wir sicher, dass unsere Python-Programme verständlich, wartbar und professionell wirken.

Benennung von Variablen, Klassen und Dateien

1. Variablen und Funktionen:

- Schreibweise: *snake_case*
- Alle Buchstaben klein, Wörter werden durch Unterstriche getrennt.
- Beispiele:

python

Kopieren

Bearbeiten

```
variable_name = 10
```


```
def berechne_summe(a, b):  
    return a + b
```

2. Konstanten:

- Schreibweise: *UPPER_CASE_WITH_UNDERSCORES*
- Wird häufig für feste Werte genutzt, die sich während der Laufzeit nicht ändern.
- Beispiel:

python

 Kopieren

 Bearbeiten


```
MAX_VERSUCHE = 3
TIMEOUT_IN_SEKUNDEN = 30
```

3. Klassen:

- Schreibweise: *CamelCase* (Erstes Zeichen jedes Wortes groß)
- Wenn ein Klassenname aus mehreren Wörtern besteht, werden diese ohne Unterstrich direkt aneinandergereiht, jeweils mit großem Anfangsbuchstaben.
- Beispiel:

python

 Kopieren

 Bearbeiten

```
class MeinFahrzeug:
    def __init__(self, name):
        self.name = name

    def starte(self):
        print(f"{self.name} startet.")
```

4. Dateien und Module:

- Schreibweise: *klein*, ggf. mit Unterstrich, wenn mehrere Wörter nötig sind.
- Beispiel:
 - mein_module.py
 - utils.py


Empfehlungen für arithmetische Operationen

1. Leerzeichen um Operatoren

- Setze in der Regel **ein Leerzeichen** vor und nach Binär-Operatoren (z.B. +, -, *, /, ==, <=, etc.).
- Beispiel:

python

 Kopieren

 Bearbeiten


```
summe = a + b
differenz = x - y
produkt = m * n
vergleich = zahl == 42
```

2. Aussagekräftige Namen

- Gib Variablen zu Rechenoperationen möglichst eindeutige Namen. Das macht deinen Code verständlicher und leichter wartbar.
- Beispiel:

python

 Kopieren

 Bearbeiten

```
artikel_preis = 19.99
anzahl_artikel = 5
gesamtkosten = artikel_preis * anzahl_artikel
```

3. Klammern und Zeilenumbrüche

- Wenn eine Rechenoperation zu lang wird (z.B. wegen der 79-Zeichen-Regel), nutze **implizite Zeilenfortsetzung** in Klammern.
- Beispiele:

python

 Kopieren

 Bearbeiten

```
# Vorzugsweise Zeilenumbruch innerhalb von Klammern
gesamtkosten = (
    artikel_preis * anzahl_artikel
    - rabattsumme
    + versandkosten
)

# Oder nach einem Operator
gesamtkosten = artikel_preis * anzahl_artikel \
    - rabattsumme \
    + versandkosten
```