

CODE-REVIEW BEI CODE SOLUTIONS



Code-Autor



Code-Reviewer



Code-Reviews gehören bei der CodeSolutions Neumarkt GmbH zum Alltag, um die Qualität des Codes sicherzustellen und Probleme frühzeitig zu erkennen. Sie fördern die Zusammenarbeit und garantieren, dass nur qualitativ hochwertiger Code in Produktion geht.

Sie schauen zwei Ihrer Kollegen beim Code-Review über die Schulter.



Code-Autor

```
# Dieses Programm berechnet die
# Summe der ganzen Zahlen von 1
# bis 20
summe = 0
zahl = 1

while zahl <= 20:
    summe += zahl
    zahl += 1

# Ausgabe
print(f"Die Summe der ganzen
Zahlen von 1 bis 20 ist:
{summe}")
```

Code-Reviewer



```
/*
Verbesserungsvorschlag:
Dieser Vorgang ist prädestiniert für eine
sequenzbasierte Schleife, da sie hier
übersichtlicher und sinnvoller ist.
Mein Vorschlag ist, eine For-Schleife in
Verbindung mit der range-Funktion zu
verwenden:
*/

for zahl in range(1, 21):
    summe += zahl
```

Verbesserungsvorschlag des Reviewers

- Beraten Sie sich mit Ihrem Sitznachbarn / Ihrer Sitznachbarin:
 - Was könnte eine **sequenzbasierte** Schleife sein?
 - Welche Bedeutung könnte die **range-Funktion** im Schleifenkopf für eine Bedeutung haben?
 - Recherchieren Sie im Internet zur Bedeutung der **range-Funktion**.



5 Minuten

```
for zahl in range(1, 21):  
    summe += zahl
```



Code-Autor

```
# Dieses Programm berechnet die
# Summe der ganzen Zahlen von 1
# bis 20
summe = 0
zahl = 1

while zahl <= 20:
    summe += zahl
    zahl += 1

# Ausgabe
print(f"Die Summe der ganzen
Zahlen von 1 bis 20 ist:
{summe}")
```

Code-Reviewer

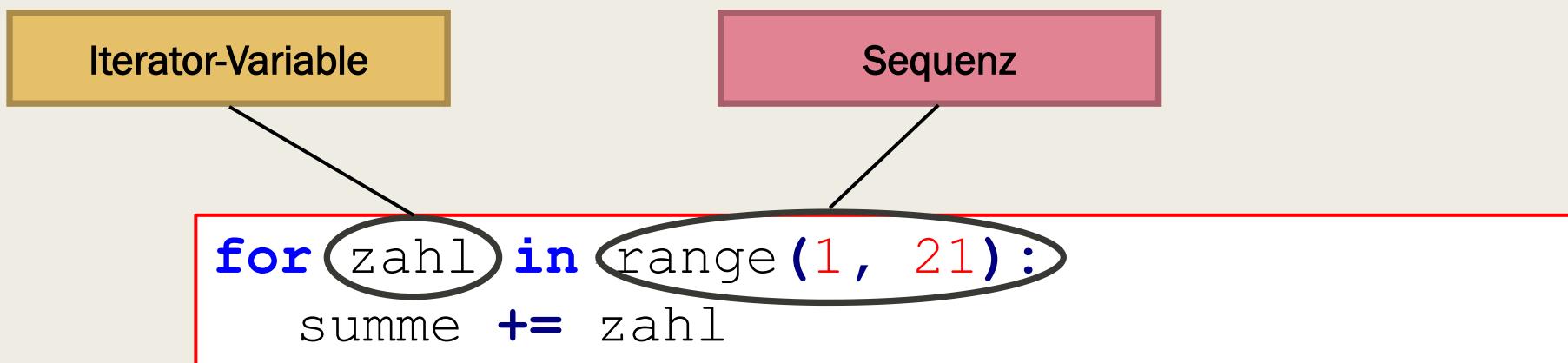


```
/*
Verbesserungsvorschlag:
Dieser Vorgang ist prädestiniert für eine
sequenzbasierte Schleife, da sie hier
übersichtlicher und sinnvoller ist.
Mein Vorschlag ist, eine For-Schleife in
Verbindung mit der range-Funktion zu
verwenden:
*/

for zahl in range(1, 21):
    summe += zahl
```

Die For-Schleife

- Bestandteile der For-Schleife



Die For-Schleife

1 2 3 ... 20

Sequenz

?

```
for zahl in range(1, 21):  
    summe += zahl
```

Die For-Schleife

Iterator-Variable	Schleifen durchlauf	Inhalt von zahl	Inhalt von summe
	1		

for zahl **in** range(1, 21):
 summe += zahl

Die For-Schleife

Iterator-Variable	Schleifen durchlauf	Inhalt von zahl	Inhalt von summe
	1	1	1

`for zahl in range(1, 21):
 summe += zahl`

Die For-Schleife

Iterator-Variable	Schleifen durchlauf	Inhalt von zahl	Inhalt von summe
	1	1	1
	2	2	3

`for zahl in range(1, 21):
 summe += zahl`

Die For-Schleife

Iterator-Variable	Schleifen durchlauf	Inhalt von zahl	Inhalt von summe
	1	1	1
	2	2	3
	3	3	6

```
for zahl in range(1, 21):
    summe += zahl
```

Die For-Schleife

Iterator-Variable

Schleifen durchlauf	Inhalt von zahl	Inhalt von summe
1	1	1
2	2	3
3	3	6
4	4	10

```
for zahl in range(1, 21):  
    summe += zahl
```

While-Schleife vs. For-Schleife

- Die **while**-Schleife
 - *Bedingungsabhängig*
 - Wiederholt sich, solange eine Bedingung wahr ist.
 - Anzahl der Durchläufe unbekannt
 - *Benutzerdefinierte Bedingung*
 - Bedingung wird vor jedem Durchlauf überprüft.
 - *Manuelle Aktualisierung*
 - Bedingung und Update werden separat behandelt.

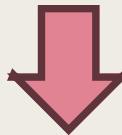
```
i = 0
while i < 5:
    print(i)
    i += 1
```

While-Schleife vs. For-Schleife

- Die **For-Schleife**
 - *Sequenzbasiert*
 - Iteriert über Elemente einer Sequenz.
 - *Feste Anzahl von Durchläufen*
 - Definierte Anzahl von Iterationen.
 - Anzahl der Durchläufe bekannt.
 - *Automatische Aktualisierung*
 - Die Variable wird automatisch aktualisiert.



0 1 2 3 4



```
for i in range(5):  
    print(i)
```

Übungsaufgabe



- Erstellen Sie ein Python-Skript, das mithilfe einer for-Schleife den Durchschnitt aller ganzen Zahlen von 1 bis 30 (inklusiv) bestimmt.
- Verwenden Sie hierfür die **range-Funktion** mit geeigneten Grenzen.
- Verzichten Sie auf die Verwendung der **sum-Funktion**.



10 Minuten



Einzelarbeit