

Schleifen

Spickzettel aus Jannicks C# Intensivkurs

Was sind Schleifen?

Eine Schleife besteht aus einem kleinen Code-Grundgerüst und bietet dir die Möglichkeit durch Arrays und Listen (also mehreren Objekten) durch zu iterieren (Code für jedes Element des Arrays oder der Liste auszuführen).

Wenn du beispielsweise 20 Dateien auf deinem Desktop liegen hast, die du alle löschen möchtest, dann wählst du diese 20 Dateien aus und eine Schleife wird ausgeführt die sagt: "Für jedes Element der 20 ausgewählten Dateien → Lösche das Element"

So wird also "Lösche das Element" 20x ausgeführt und am Ende sind alle Elemente gelöscht.

Welche Schleifen gibt es?

Es gibt mehrere Arten von Schleifen. Die wichtigsten beiden sind die `for`-Schleife und die `foreach`-Schleife (beide haben denselben Zweck).

Hinweis: Die `for`-Schleife verwendet einen **Index** (Position).

Hinweis: Die `foreach`-Schleife verwendet das **Objekt** selbst.

Praktische Beispiele

ForEach-Schleife

```
void Main() {  
  
    int[] numbers = new int[] {1,3,5,7}; // Array mit 4 Werten  
  
    foreach(int num in numbers){  
  
        Console.WriteLine(num); // Wird insgesamt 4x ausgeführt  
    }  
  
    // Finale Ausgabe in der Konsole: 1 3 5 7  
}
```

Wie du sehen kannst, verwenden wir die `foreach`-Schleife mit dem Keyword `foreach`, gefolgt von einer Definierung des Schleifen Algorithmus:

```
foreach(int num in numbers){  
    ...  
}
```

Da unser Array `numbers` vom Datentyp `integer` ist, ist gleichzeitig jedes Element in diesem Array also auch vom Datentyp `int`.

Aus diesem Grund geben wir an, dass wir für jedes Element im (`in`) Array `numbers`, dass wir kurzzeitig mit `num` und vom Datentyp `int` bezeichnen folgenden Code ausführen wollen:

```
Console.WriteLine(num);
```

Der Code wird also insgesamt 4x ausgeführt, weil es vier Elemente im Array gibt. **num steht dabei jedesmal für das derzeitige Element.**

For-Schleife

```
void Main() {  
  
    int[] numbers = new int[] {1,3,5,7}; // Array mit 4 Werten  
  
    for(int i=0; i < numbers.Length; i++){  
  
        Console.WriteLine(numbers[i]); // Wird insgesamt 4x ausgeführt  
    }  
  
    // Finale Ausgabe in der Konsole: 1 3 5 7  
}
```

Das Ergebnis der `for`-Schleife ist, wie du sehen kannst, identisch mit dem Ergebnis der `foreach`-Schleife.

Auch hier verwenden wir ein Schleifen Keyword (`for`) Keyword, gefolgt vom Schleifen Algorithmus. Die `for`-Schleife bezieht sich im Gegensatz zur `foreach`-Schleife nicht auf das Objekt im Array selber sondern auf einen Index (`i`).

Standardmäßig startet der index `i` bei 0 weil das erste Element im Array auf dem Index 0 liegt:

```
int i=0;
```

Als zweites wird definiert wie oft die Schleife laufen soll. Das könnte einmal oder auch hundert mal sein. Im Beispiel lassen wir die Schleife so oft laufen, wie wir Element im Array haben (z.B. 4 Elemente):

```
i < numbers.Length;
```

Als letztes müssen wir den Index `i` noch erhöhen, damit wir nicht für immer bei dem Index 0 hängen bleiben:

```
i++
```

Übersetzt sieht der Schleifen Algorithmus also so aus:

“Erhöhe `i` jedes mal um 1 für insgesamt 4 Elemente und führe für jedes Element folgenden Code aus”

Abschließend kannst du innerhalb des Schleifen Codes dann den Index `i` verwenden um dich auf das jeweilige Element zu beziehen:

```
Console.WriteLine(numbers[i]);
```