

Software Development

Software-Lösungen. Einfach. Clever.



Inhalt

- for
- for -> rückwärts
- Verschachtelte for-Schleife
- foreach
- while
- do
- break

Schleifen

Software-Lösungen. Einfach. Clever.

Schleifen

for

Wird so lange durchlaufen, bis Abbruchbedingung *false* ist

C #

```
for (zähler=Anfanswert; Abbruchbedingung; zähler=neuerWert)
{
    Anweisungen;
}
```

C #

```
for (int i = 1; i <= 5; i++)
{
    Console.WriteLine($"i={i}");
}
```

Ausgabe

```
i=1
i=2
i=3
i=4
i=5
```

Schleifen

for -> rückwärts

Eine Schleife kann auch rückwärts laufen

C #

```
for (int i = 7; i > 2; i--)  
{  
    Console.WriteLine($"i={i}");  
}
```

Ausgabe

```
i=7  
i=6  
i=5  
i=4  
i=3
```

Schleifen

Verschachtelte for-Schleife

Schleifen können ineinander verschachtelt werden

C #

```
for (int i = 1; i <= 2; i++)  
{  
    for (int j = 1; j <= 5; j++)  
    {  
        Console.WriteLine($"i={i}, j={j}");  
    }  
}
```

Ausgabe

```
i=1, j=1  
i=1, j=2  
i=1, j=3  
i=1, j=4  
i=1, j=5  
i=2, j=1  
i=2, j=2  
i=2, j=3  
i=2, j=4  
i=2, j=5
```

Schleifen

foreach

- Ist nur auf Array-Typen anwendbar
- Hat keine Schleifenvariable
- Erlaubt während des Durchlaufs keine Veränderung des Arrays

C #

```
int[] myArray = new int[] { 1, 2, 3, 4, 5 };  
foreach (int i in myArray)  
{  
    Console.WriteLine($"i={i}");  
}
```

Ausgabe

```
i=1  
i=2  
i=3  
i=4  
i=5
```

Schleifen

while

Abbruchbedingung am Schleifenanfang (kopfgesteuert)

C #

```
while (Abbruchbedingung)  
{  
    Anweisungen;  
}
```

C #

```
int i = 1;  
while (i <= 4)  
{  
    Console.WriteLine($"i={i}");  
    i++;  
}
```

Ausgabe

```
i=1  
i=2  
i=3  
i=4
```


Schleifen

do

Abbruchbedingung am Schleifenende (fussgesteuert)

C #

```
do
{
    Anweisungen;
}
while (Abbruchbedingung)
```

C #

```
int i = 1;
do
{
    Console.WriteLine($"i={i}");
    i++;
}
while (i <= 4);
```

Ausgabe

```
i=1
i=2
i=3
i=4
```

Schleifen

break

Mit **break** vorzeitig die Schleife verlassen

C #

```
int i = 1;
do
{
    Console.WriteLine($"i={i}");
    if (i == 8)
    {
        break; // Abbruch testen
    }
    i++;
}
while (i <= 10);
```

Ausgabe

```
i=1
i=2
i=3
i=4
i=5
i=6
i=7
i=8
```

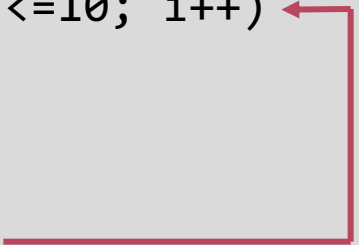
Schleifen

continue

continue übergibt den Verlauf an die nächste Iterations-Anweisung

C #

```
for (int i = 1; i <=10; i++)  
{  
    if (i%2 == 0)  
    {  
        continue;  
    }  
    Console.WriteLine($"i={i}");  
}
```



Ausgabe

```
i=1  
i=3  
i=5  
i=7  
i=9
```

Übungen

Uebungen/Kurseinheit 4/Übung Schleifen.pdf



Software-Lösungen. Einfach. Clever.