

Prä- und Postinkrement

Sie kennen bereits Anweisungen der Form `x++` als Abkürzung für die Zuweisung `x=x+1`. Man nennt `++` auch den Inkrement-Operator, denn Inkrementieren ist der Fachbegriff für schrittweises Erhöhen. Bei einfachen Zahlen heißt das nichts anderes als 1 addieren.

Bis jetzt haben wir noch nicht ausgenutzt, dass so eine Anweisung auch einen Wert ergibt. Das heißt, sie ist nicht nur eine Zuweisung, sondern auch ein Ausdruck. Das Ergebnis hängt davon ab, ob der Operator `++` vor oder nach der Variablen steht:

- `x++` gibt zuerst den Wert von `x` zurück und erhöht ihn dann um 1.
- `++x` erhöht zuerst `x` um 1 und gibt dann das Ergebnis zurück.

Beispiel 1:

```
int x = 5;
cout << x++ << endl; // Gibt 5 aus.
cout << x << endl;   // Gibt 6 aus.
```

Beispiel 2:

```
int x = 5;
cout << ++x << endl; // Gibt 6 aus.
cout << x << endl;   // Gibt 6 aus.
```

Prä- und Postdekrement

Das soeben Erklärte gilt analog für das Dekrementieren, also für das schrittweise Herunterzählen, mit dem Operator `--`.

Aufgabe

Welche Zahlen geben die beiden Schleifen im folgenden Programm auf dem Bildschirm aus?

```
#include <iostream>
using namespace std;

int main()
{
    int i = 3;
    do
    {
        cout << i-- << ' ';
    } while (i > 0);

    cout << endl; // neue Zeile auf dem Bildschirm

    i = 3;
    do
    {
        cout << --i << ' ';
    } while (i > 0);
}
```

3 2 1
2 1 0