

Range based for-Schleife in C++

Diese steht ab dem Sprachstandard C++11 zur Verfügung und erlaubt es, alle Elemente eines Containerobjekts (z.B. Instanz von `std::vector`, `std::list`, `std::array`, `std::initializer_list`, `std::string`) und auch eines C- Arrays **vorwärts** zu durchlaufen. Die `break` und `continue` Anweisungen sind auch hier im Schleifenkörper erlaubt.

Die Syntax ist:

```
for (variablendeklaration : Container-Instanz)  
    Anweisung;           // oder Block in {}
```

```
std::vector<int> v{1,2,3,4,5}; // ein Container mit den int 1-5  
for (int i : v)               // oder: for (auto i : v)  
    cout << i << ' ';       // gibt aus: 1 2 3 4 5
```

```
double u[] = {1.,2.1,-3.};    // ein C-Array  
for (auto x : u)              // x hat den Typ, den u speichert (double)  
    cout << x << ' ';       // gibt aus: 1 2.1 -3
```

```
for (auto x : u)              // x durchläuft alle Werte von u  
    x = 5;                    // das ändert die Index-Variable x, aber nicht u !!!
```

```
for (auto& x : u)             // x durchläuft alle Elemente von u per Referenz  
    x = 5;                    // ändert alle Elemente von u auf 5!!!!
```

```
for (const auto& x : u)       // x durchläuft Elemente von u per const Referenz  
    x = 5;                    // das ist ein Fehler, da x konstant ist
```

In C++17 gibt es auch die Form mit `Init`, z.B.

```
for (int n = 0; auto x : u)    // n zur Nummerierung  
    cout << "u[" << n++ << "] = " << x << '\n';
```