

IF-Else-Statement

Syntax:

```
if (BEDINGUNG) { ANWEISUNGSBLOCK 1 }  
else { ANWEISUNGSBLOCK 2 }
```

```
1 #include <iostream>  
2  
3 int main()  
4 {  
5     std::cout << "Bitte eine ganze Zahl eingeben: ";  
6     int a;  
7     std::cin >> a;  
8  
9     if(a > 0)  
10    {  
11        std::cout << "Diese Zahl ist positiv." << std::endl;  
12    } else  
13    {  
14        std::cout << "Diese Zahl ist nicht positiv." << std::endl;  
15    }  
16  
17    return 0;  
18 }
```

Vergleichsoperatoren

Operator	Syntax
Prüfen auf Gleichheit	<code>a == b</code>
Prüfen auf Ungleichheit	<code>a != b</code>
Prüfen, ob <code>a</code> echt größer als <code>b</code> ist	<code>a > b</code>
Prüfen, ob <code>a</code> echt kleiner als <code>b</code> ist	<code>a < b</code>
Prüfen, ob <code>a</code> größer oder gleich <code>b</code> ist	<code>a >= b</code>
Prüfen, ob <code>a</code> kleiner oder gleich <code>b</code> ist	<code>a <= b</code>

Table: Vergleichoperatoren

while-Schleifen

Syntax

```
while (BEDINGUNG)  
{  
    ANWEISUNGSBLOCK  
}
```

Solange die BEDINGUNG erfüllt ist, wird der ANWEISUNGSBLOCK wiederholt.

Beispiel:

```
1 int i = -5;  
2  
3 while(i < 10)  
4 {  
5     std::cout << i << std::endl;  
6     ++i;  
7 }
```

for-Schleifen

Syntax:

```
for ( INITIALISIERUNG; BEDINGUNG; STEP )  
{  
    ANWEISUNGSBLOCK  
}
```

Führe erst die INITIALISIERUNG aus. Solange die BEDINGUNG erfüllt ist, führe erst den ANWEISUNGSBLOCK und dann den STEP aus.

Beispiel:

```
1 for(int i = -5; i < 10 ; ++i)  
2 {  
3     std::cout << i << std::endl;  
4 }
```