

Datentypen

<code>bool</code>	Datentyp für Wahrheitswerte
Literal: <code>true</code> , <code>false</code>	
<pre>for (bool another = true; another;) { int a; std::cin >> a; if (a < 3) // Note: a < 3 is of type bool another = false; }</pre>	

Operatoren

<code>&&</code>	Logisches UND
Präzedenz: 6 und Assoziativität: links	
Kurzschluss-Auswertung: <code>&&</code> wertet <i>immer</i> den <i>linken</i> Operanden zuerst aus. Ergibt dieser <code>false</code> , so wird der <i>rechte</i> Operand <i>nicht mehr</i> ausgewertet.	
<pre>if (3 > 2 && 10 > 11) // no short circuit evaluation std::cout << "Of course not!\n"; int a = 3; if (false && ++a < 2) // short circuit evaluation std::cout << "Of course not!\n"; std::cout << a << "\n"; // Output: 3</pre>	

<code> </code>	Logisches ODER
-----------------	----------------

(...)

Programmier-Befehle - Woche 03

(...)

Präzedenz: 5 und Assoziativität: links

Kurzschluss-Auswertung: `||` wertet auch *immer* den *linken* Operanden zuerst aus. Ergibt dieser `true`, so wird der *rechte* Operand *nicht mehr* ausgewertet.

```
if (8 < 3 || -2 > -5)    // no short circuit evaluation
    std::cout << "Yes!\n";

int a = 3;
if (3 < 8 || ++a < 2)    // short circuit evaluation
    std::cout << "Yes!\n";

std::cout << a << "\n"; // Output: 3
```

!

Logisches NICHT

Präzedenz: 16 und Assoziativität: rechts

```
int a,b,c;
std::cin >> a >> b >> c;

if ( ! (a < b && c < b) )
    std::cout << "b is not max!" << "\n";
```

<

strikt kleiner

Präzedenz: 11 und Assoziativität: links

Sonst gibt es noch:

- > strikt grösser
- <= kleiner gleich
- >= grösser gleich

(...)

(...)

```
int a,b;
std::cin >> a >> b;

if (a < b)
    std::cout << "a smaller than b" << "\n";
```

==

exakt gleich

Präzedenz: 10 und Assoziativität: links

Sonst gibt es noch:

!= **ungleich**

```
int a,b;
std::cin >> a >> b;

if (a == b)
    std::cout << "a is equal to b" << "\n";
```

Generell

if-else

bedingtes Ausführen von Code

Der **else**-Teil ist optional.

```
int a,b;
std::cin >> a >> b;

// if a < b then output "Ja"; otherwise output "Nein"
if (a < b) {
    std::cout << "Ja\n";
} else {
    std::cout << "Nein\n";
}
```

Programmier-Befehle - Woche 03

<code>assert</code>	sofortiges Stoppen des Programms zu Testzwecken
Erfordert: <code>#include <cassert></code>	
<pre>int a,b; std::cin >> a >> b; assert(b != 0); // prevent division by 0 std::cout << a / b << "\n";</pre>	

Schleifen - Teil 1

<code>for (...) {...}</code>	for-Schleife
Wenn man eine leere Condition als Abbruchbedingung angibt, so wird diese als <code>true</code> interpretiert.	
<pre>unsigned int n; std::cin >> n; unsigned int sum = 0; for (unsigned int i = 1; i <= n; ++i) { sum += i; std::cout << "Intermediate result: " << sum << "\n"; }</pre>	