

if & if-else-Anweisungen

Mit einer *if-Anweisung* kann man in Abhängigkeit von einer Bedingung steuern, ob eine Anweisung ausgeführt wird oder nicht, also:

- 1 | if (condition) statement
- 2 | if (condition) statement else statement

Hier wird *condition* in einen booleschen Wert konvertiert. In der ersten Form (ohne *else*) wird die Anweisung nur dann ausgeführt, wenn *condition* den Wert *true* hat. In der zweiten Form (mit *else*) wird die Anweisung ausgeführt, falls die Bedingung erfüllt ist und andernfalls die zweite.

- 1 | if (x < 1) a = 1;
- 2 | else a = 2;

Die Anweisung $a=1$ wird ausgeführt, wenn die Bedingung $x < 1$ erfüllt ist und andernfalls die Anweisung $a=2$. Eine Bedingung wird oft durch den Vergleich von zwei Ausdrücken mit einem der Operatoren $<$, $<=$, $>$, $>=$, $==$ oder $!=$ gebildet. Mit dem Operator $==$ prüft man die Gleichheit der beiden Operanden und mit $!=$ deren Ungleichheit. Dabei muss man darauf achten, dass man den Operator $==$ nicht mit dem Zuweisungsoperator $=$ verwechselt. Eine solche Verwechslung wird vom Computer nicht als Fehler bemängelt, sondern lediglich mit einem nicht funktionierendem Programm 🤪

Sollen mehrere Anweisungen in Abhängigkeit von einer Bedingung ausgeführt werden, müssen sie durch eine Verbundanweisung mit $\{$ und $\}$ zusammengefasst werden, wie im Beispiel.

- 1 | { statement-seq opt }

Eine Verbundanweisung wird auch als Block bezeichnet. In ihr können auch Variable definiert werden, die dann in einem Block lokal sind.

Verwendet man im *else-Zweig* einer *if-Anweisung* wieder eine *if-Anweisung*, erhält man eine verschachtelte *if-Anweisung*. Damit ist eine Mehrfachauswahl möglich, siehe Beispiel.

- 1 | if (x<0)
- 2 | {
- 3 | a = -1;
- 4 | x = -x;
- 5 | }
- 6 | else if (x>0) a = 1;
- 7 | else a = 0;

Bei der Ausführung dieser *if-Anweisung* wird zuerst geprüft, ob die Bedingung den Wert *true* hat. Trifft dies zu, dann wird die folgende Anweisung ausgeführt und andernfalls die *else* folgende.

Bei einer *if-Anweisung* ohne *else-Zweig* wird nichts gemacht, wenn die Bedingung nicht erfüllt ist.

Es empfiehlt sich so oft wie nur möglich *if-else-Anweisung* zu verwenden, da man einfach an Zeilen spart und bei einem *false* trotzdem etwas ausgeben kann.

Ergänzungen zu diesem Tutorial:

- [Die Grundlagen des Scriptens](#)

Mit Erlaubnis des Autors wurde dieses Tutorial auf <http://www.GTAVision.com> hochgeladen.

Weitere Tutorials rund um GTA findet ihr dort im Tutorialbereich.