

## Ein bisschen Python

### Episode 9

#### *if-Anweisung 2: Vergleiche und Syntax der if-Anweisung*



In der letzten Episode sprachen wir über bedingte Anweisungen und Verzweigungen. In dieser Episode sollen Sie lernen, mit welchen Operatoren Sie in if-Anweisungen logische Aussagen formulieren können.

Bei einer if-Anweisung prüfen Sie immer eine logische Aussage auf ihren Wahrheitsgehalt. Dafür müssen Sie eine logische Aussage jedoch erst einmal formulieren. Das machen Sie, indem Sie bestimmte Variablen auf einen Wert prüfen. In den Episoden-Notes und hier in der Episoden-Beschreibung finden Sie eine kleine Liste. Gehen wir sie einmal durch:

Die arithmetischen Vergleiche kleiner als bzw. größer als können Sie direkt aus der Mathematik verwenden.

Der Vergleichsoperator für den Test auf Gleichheit (z.B. „a ist gleich b“) ist ein zweifaches Ist-Gleich-Symbol.

Der Vergleichsoperator für den Test auf Ungleichheit (z.B. „a ist ungleich b“) ist ein Ausrufezeichen gefolgt von einem Gleich-Symbol. Diese Kombination hat Ähnlichkeit mit einem durchgestrichenen Ist-Gleich-Symbol.

Die Vergleichsoperatoren für größer gleich und kleiner gleich werden so geschrieben wie man sie spricht: beim größer-gleich erst ein größer-als-Zeichen gefolgt von einem Gleichheitssymbol und beim kleiner-gleich erst ein kleiner-als-Zeichen gefolgt von einem Gleichheitssymbol.

Natürlich können Sie nicht nur zwei Variablen miteinander vergleichen, sondern auch eine Variable mit einem bestimmten Wert. Denken Sie an unser Beispiel aus Episode 7, in der wir geprüft haben, ob der Messwert für Helligkeit unter einem bestimmten Wert liegt. Da haben wir geprüft: `helligkeit < 25`

Solche Vergleiche sind boolsche Ausdrücke, benannt nach dem irischen Mathematiker, Philosoph und Logiker George Boole, der im 19. Jahrhundert lebte. Solche Vergleiche können entweder wahr oder falsch sein. Wahr und falsch ist in dem Fall nicht wertend gemeint, sondern besagt einfach nur, was das Ergebnis des Vergleichs ist. Im Weiteren benutzen wir die englischen Begriffe „true“ und „false“.

Wie werden nun if-Anweisungen in Python realisiert? Dazu verwenden wir das Schlüsselwort `if` (deutsch wenn) gefolgt von dem boolschen Ausdruck, hier also unseres Vergleichs. Nach dem Ausdruck wird dann ein Doppelpunkt gesetzt. Jetzt folgt der Anweisungsblock, der ausgeführt wird, wenn die boolsche Aussage richtig ist. Anweisungsblock bedeutet, dass ein Befehl ausgeführt wird oder eine Reihe von Befehlen. Woher weiß der Python-Interpreter nun, wo der Anweisungsblock anfängt und wo er aufhört? Dazu müssen wir Einrückungen verwenden. Die Einrückungen müssen alle die gleiche Tiefe haben. Die meisten Programmierer verwenden dafür die Tabulator-Taste. Sie können aber auch die Leertaste nehmen.

Bei bedingten Anweisungen reicht es, nach dem Anweisungsblock, der bei einer bestimmten Bedingung ausgeführt wird, mit dem Programm einfach fortzufahren. Wichtig ist hier: Die Einrückungen sind eine Stufe zurückzunehmen. Sehen Sie sich dazu das Beispiel in den Episode-Notes und der Episoden-Beschreibung an.

Verzweigungen haben ja einen zweiten Strang. Wie wird dieser nun realisiert? Zunächst kommt nach der if-Anweisung ein Doppelpunkt, der true-Anweisungsblock folgt einfach eingerückt. Nach dem true-Anweisungsblock kommt das Schlüsselwort `else` (sonst) gefolgt

von einem Doppelpunkt. Der false-Anwendungsblock folgt in einer neuen Zeile genauso eingerückt wie der true-Anwendungsblock. Sehen Sie sich dazu unser Beispiel in den Episode-Notes bzw. in der Episoden-Beschreibung an.

Viel Spaß beim Python-Lernen.