

Ein bisschen Python
Episode 11
Boolesche Ausdrücke



Bedingungen in der Programmierung bedeutet, dass man eine Variable auf einen bestimmten Wert prüft, hier zwei Beispiele, die auch in den Show-Notes zu finden sind:

```
helligkeit < 20
stunde > 18
stunde < 8
```

Solche Vergleiche sind boolesche Ausdrücke.

Trifft eine Bedingung zu, z.B, wenn helligkeit einen Wert von 18 hat, dann wird der Ausdruck als wahr analysiert. Ist die Bedingung unzutreffend, dann ist der Wert des Ausdrucks falsch.

Jetzt eine kleine Audioübung, die Bedingung ist `stunde > 10`. Ich nenne jetzt Werte für `stunde`, sagen Sie, ob der Ausdruck `stunde > 10` für diesen Wert wahr ist oder falsch. Nach einigen Sekunden hören Sie einen Signalton, danach werde ich die Lösung sagen. Also noch einmal: Der Ausdruck ist `stunde > 10`.

```
stunde = 0           falsch, 0 ist nicht größer als 10
stunde = 15          wahr, 15 ist größer als 10
stunde = 20          wahr, 20 ist größer als 10
stunde = 10          falsch, 10 ist nicht größer als 10
```

Manchmal muss ein Anweisungsblock ausgeführt werden, wenn eine Bedingung nicht zutrifft. Man kann den Wahrheitswert eines Booleschen Ausdrucks negieren. D.h., ist der Ausdruck wahr, so ist seine Negation falsch, ist der Ausdruck falsch, so ist eine Negation wahr. Das Schlüsselwort zur Negation ist in Python `not`.

Ein Beispiel:

`not stunde > 10` ist für den Wert `stunde = 0` wahr, weil `0` nicht größer als `10` ist, der Ausdruck `stunde > 10` ist somit falsch. Die Negation des Ausdrucks ist somit wahr.

Die Wahrheitstabelle finden Sie in den Show-Notes:

A	nicht A
falsch	wahr
wahr	falsch

Ist der Boolesche Ausdruck A (z.B. `9 > 10`) falsch, dann ist nicht A wahr. Ist der Ausdruck A wahr, dann ist nicht A falsch.

Oft muss ein Anweisungsblock ausgeführt werden, wenn es Bedingungen für zwei Variablen gibt. Nehmen wir an, eine Lampe soll ab 19:00 angeschaltet werden, wenn der Helligkeitswert einen bestimmten Wert (z.B. 20) unterschreitet. Der Stundenanteil der Uhrzeit ist in der Variable `stunde` gespeichert. Der Helligkeitswert ist in der Variable `helligkeit` gespeichert. Der Boolesche Ausdruck A ist somit `stunde > 18` und der Boolesche Ausdruck B wäre z.B. `helligkeit < 20`. Die Lampe soll nur angeschaltet werden wenn sowohl der Ausdruck A (`stunde > 18`) und der Ausdruck B (`helligkeit < 20`) wahr sind. Verbindet man beide Ausdrücke durch die logische Verknüpfung UND, so wird dies erreicht. Diese ist

wahr, wenn beide mit UND verknüpften Aussagen wahr sind, sonst wird diese Verknüpfung als falsch ausgewertet.

In den Show-Notes finden Sie die entsprechende Wahrheitstabelle:

A	B	A und B
falsch	falsch	falsch
falsch	wahr	falsch
wahr	falsch	falsch
wahr	wahr	wahr

Die Bedingung für unseren Fall wäre also

$\text{stunde} > 18 \text{ and helligkeit} < 20$

Diese Bedingung wäre z.B. wahr für $\text{stunde} = 20$ und $\text{helligkeit} = 10$, weil beide Teilbedingungen als wahr ausgewertet werden.

Für $\text{stunde} = 20$ und $\text{helligkeit} = 25$ würde sie als falsch ausgewertet, da nur eine Teilbedingung als wahr ausgewertet werden kann.

Manchmal muss die der Wert einer Variable in einem bestimmten Bereich liegen. Nehmen wir an, eine Lampe soll zwischen 19:00 abends und 8:00 morgens angeschaltet werden. Es gibt als die Aussage A ($\text{stunde} > 18$) und die Aussage B ($\text{stunde} < 8$). Man benötigt also eine Verknüpfung, die als wahr ausgewertet wird, wenn mindestens eine der verknüpften Aussagen wahr ist. Diese Verknüpfung ist die ODER-Verknüpfung, deren Wahrheitstabelle in den Show-Notes zu finden sind:

A	B	A oder B
falsch	falsch	falsch
falsch	wahr	wahr
wahr	falsch	wahr
wahr	wahr	wahr

Die Bedingung würde für das Problem also lauten

$\text{stunde} > 18 \text{ or stunde} < 8$

Diese Bedingung ist für $\text{stunde} = 4$ wahr, weil eine der beiden Teilbedingungen wahr ist.

Verschiedene Bedingungen können auch komplex miteinander verknüpft werden. Ähnlich wie bei der Algebra, in der gilt: Punktrechnung vor Strichrechnung, gibt es auch in dem Fall eine Hierarchie, die eingehalten wird. Erst werden die Negationen vorgenommen, dann werden die mit and verknüpften Aussagen verknüpft, dann die mit or verknüpften. Es gilt also NAO: Negation vor And vor Or.

Also, keine Angst vor Boole und viel Spaß beim Python-Lernen.