

Bedingungen in Python

Jacques Mock Schindler

25.09.2024

Bedingungen

Es gibt Situationen, in denen soll einzelne Anweisungen eines Programmes nur dann ausgeführt werden, wenn eine bestimmte Bedingung zutrifft. Die allgemeine Struktur einer solchen wenn-dann-sonst-Konstruktion sieht folgendermassen aus:

WENN Bedingung DANN

Anweisungen, die ausgeführt werden, wenn die Bedingung wahr ist

SONST

Anweisungen, die ausgeführt werden, wenn die Bedingung falsch ist

ENDE WENN

In Python wird das als

```
1 \ControlFlowTok{if}\NormalTok{ x }\OperatorTok{==} \VariableTok{True}
2 \NormalTok{    ...}
3 \ControlFlowTok{else}\NormalTok{:}
4 \NormalTok{    ...}
```

dargestellt. Dabei können nach dem `if` beliebige Bedingungen formuliert werden. Die dazu erforderlichen Vergleichsoperatoren sind

| Operator | Bedeutung |
|--------------------|----------------|
| <code>==</code> | ist gleich |
| <code>!=</code> | nicht gleich |
| <code><=</code> | kleiner gleich |
| <code>>=</code> | grösser gleich |

Mehrere Bedingungen können dabei mit `und` bzw. `oder` verknüpft werden. Die Verknüpfung der Bedingungen erfolgen nach den Regeln der formalen Logik.

Aussagenlogik (Verknüpfung von Bedingungen)

Computer verarbeiten Daten. Daten werden im Computer als Nullen und Einsen oder als "Strom" und "kein Strom" dargestellt. Mit diesen zwei Zuständen lässt sich alles darstellen. Weil alles mit zwei Zuständen dargestellt werden kann ist diese Form der Darstellung *digital*.

Im folgenden sollen die logischen Operationen

- logisches Nicht (*not*)
- logisches Und (*and*)
- logisches Oder (*or*)
- logisches Exklusiv-Oder (*xor*)

betrachtet werden.

Die Negation

Die logische Negation wird als *nicht*-Operation bezeichnet.

Wenn es nur zwei Werte - 0 und 1 - gibt, dann ist

- der Wert, der *nicht* 0 ist, 1 und
- der Wert, der *nicht* 1 ist, 0.

Nicht wird mit dem Symbol \neg oder $\bar{}$ (Strich über dem verneinten Zeichen) dargestellt.

Man kann damit schreiben

$$\neg 1 = 0$$

oder

$$\bar{1} = 0$$

Die Verneinung kann in einer sogenannten Wahrheitstabelle dargestellt werden.

| x | \bar{x} |
|-----|-----------|
| 0 | 1 |
| 1 | 0 |

Mit Wahrheitstabellen lassen sich auch andere logische Operationen definieren.

Logisches Und (*and*)

Das logische Und (*and*) wird mit dem Symbol \wedge dargestellt.

Die Wahrheitstabelle dafür sieht folgendermassen aus:

| x | y | $x \wedge y$ |
|-----|-----|--------------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Logisches Oder (*or*)

Das logische Oder (*or*) wird mit dem Symbol \vee dargestellt.

Die Wahrheitstabelle für das logische Oder sieht folgendermassen aus:

| x | y | $x \vee y$ |
|-----|-----|------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Logisches exklusiv-Oder (*xor*)

Das logische exklusiv-Oder (*xor*) wird mit dem Symbol $\dot{\vee}$ dargestellt.

Die Wahrheitstabelle für das logische exklusiv-Oder sieht folgendermassen aus:

| x | y | $x \dot{\vee} y$ |
|-----|-----|------------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Rangfolge logischer Operatoren

Analog zu den arithmetischen Operationen haben auch die logischen Operatoren eine Rangfolge. Am stärksten bindet dabei *not* ($\neg x$ oder \bar{x}) gefolgt von *and* ($x \wedge y$) und *or* ($x \vee y$).

Arbeitsblatt

Hier finden Sie ein Arbeitsblatt für die Arbeit mit Bedingungen in Python.