

Handout: Logische Operatoren in Python

Logische Operatoren werden verwendet, um mehrere Bedingungen zu verknüpfen und komplexere Entscheidungen zu treffen.

Übersicht: Logische Operatoren

| Operator | Beschreibung | Beispiel | Ergebnis |
|----------|-----------------------------------------------|---------------|----------|
| and | Wahr, wenn beide Bedingungen wahr sind | True and True | True |
| or | Wahr, wenn eine Bedingung wahr ist | True or False | True |
| not | Negiert (kehrt um) den Wahrheitswert | not True | False |

Verwendung in Bedingungen

and

```
x = 10
y = 20
if x == 10 and y == 20:
    print("Beide Bedingungen sind wahr.")
```

or

```
if x == 10 or y == 10:
    print("Mindestens eine Bedingung ist wahr.")
```

not

```
anwesend = True
if not anwesend:
    print("Nicht anwesend.")
else:
    print("Anwesend.")
```

Kombination mit Vergleichsoperatoren

Logische Operatoren werden oft mit Vergleichsoperatoren kombiniert:

```
note = 2
alter = 17

if note <= 2 and alter < 18:
    print("Gute Leistung für einen Jugendlichen.")
```

! Priorität (Reihenfolge der Auswertung)

- Klammern `()` haben die höchste Priorität
- Danach folgen Vergleichsoperatoren (`==`, `!=`, `<`, `>`, etc.)
- Dann `not`, `and`, `or` in dieser Reihenfolge

☑ Tipps

- Verwende Klammern zur besseren Lesbarkeit.
- `and` ist strenger als `or`.
- `not` kehrt den Wahrheitswert um.

📄 Weitere Informationen

- [Python Docs: Booleans](#)