

# Cheat-Sheet: Verzweigungen und Schleifen

Python | Klasse 10 - Gymnasium | Bedingungen,  
Wiederholungen, typische Muster

## Worum geht es?

Mit Verzweigungen trifft ein Programm Entscheidungen. Mit Schleifen wiederholt es Anweisungen.

```
if temperatur > 30:  
    print("heiss")
```

```
for zahl in range(3):  
    print(zahl)
```

## Bedingungen

Eine Bedingung ist ein Ausdruck, der **True** oder **False** ergibt.

```
alter = 16  
print(alter >= 18)    # False  
print(alter < 18)     # True
```

## Vergleichsoperatoren

```
==  gleich  
!=  ungleich  
<   kleiner  
>   groesser  
<=  kleiner oder gleich  
>=  groesser oder gleich
```

**Achtung: == vergleicht Werte. = weist einen Wert zu.**

## if

Der eingerückte Block wird nur ausgeführt, wenn die Bedingung wahr ist.

```
punkte = 82
```

```
if punkte >= 50:  
    print("bestanden")
```

## if - else

**else** wird ausgeführt, wenn die Bedingung falsch ist.

```
note = 5
```

```
if note <= 4:  
    print("bestanden")  
else:  
    print("nicht bestanden")
```

## if - elif - else

Mit **elif** prüft man weitere Fälle. Python nimmt den ersten passenden Fall.

```
punkte = 73
```

```
if punkte >= 90:  
    note = 1  
elif punkte >= 75:  
    note = 2  
elif punkte >= 60:  
    note = 3  
else:  
    note = 4
```

## Logische Operatoren

Mehrere Bedingungen können kombiniert werden.

```
alter = 16  
hat_ausweis = True
```

```
if alter >= 16 and hat_ausweis:  
    print("Einlass")
```

```
if alter < 6 or alter > 65:  
    print("Ermaessigung")
```

## Einrückung ist Pflicht

Blöcke erkennt Python an der Einrückung, meistens vier Leerzeichen.

```
if zahl > 0:  
    print("positiv")  
    print("fertig")
```

```
print("immer")
```

## for-Schleife

Eine **for**-Schleife wiederholt etwas für jedes Element einer Folge.

```
namen = ["Ali", "Mia", "Tom"]
```

```
for name in namen:  
    print(f"Hallo, {name}!")
```

## range()

**range()** erzeugt Zahlenfolgen. Der Stoppwert ist nicht dabei.

```
for i in range(5):  
    print(i)          # 0, 1, 2, 3, 4
```

```
for i in range(1, 6):  
    print(i)          # 1 bis 5
```

```
for i in range(2, 11, 2):  
    print(i)          # 2, 4, 6, 8, 10
```

## while-Schleife

Eine **while**-Schleife läuft, solange ihre Bedingung wahr ist.

```
zahl = 1
```

```
while zahl <= 5:  
    print(zahl)  
    zahl = zahl + 1
```

## Endlosschleifen vermeiden

Bei **while** muss sich die Bedingung irgendwann ändern.

```
zahl = 1
```

```
while zahl <= 5:
    print(zahl)
    # zahl wird erhoeht:
    zahl += 1
```

**Wenn zahl += 1 fehlt, bleibt zahl immer 1. Die Schleife endet nie.**

## break und continue

**break** beendet eine Schleife. **continue** springt zum nächsten Durchlauf.

```
for zahl in range(1, 8):
    if zahl == 5:
        break
    if zahl == 3:
        continue
    print(zahl)    # 1, 2, 4
```

## Typische Muster

Zählen, Summieren und Suchen kommen sehr häufig vor.

```
summe = 0
```

```
for zahl in range(1, 6):
    summe += zahl
```

```
print(summe)    # 15
```

## Typische Fehler

```
if punkte = 10:    # falsch
if punkte == 10:   # richtig
```

```
if punkte > 50     # Doppelpunkt fehlt
    print("ok")
```

```
while zahl < 10:
print(zahl)        # Einrueckung fehlt
```

## Merksatz

**if** entscheidet, welcher Code ausgeführt wird. **for** wiederholt über eine bekannte Folge. **while** wiederholt, solange eine Bedingung wahr ist.