

## Die `while`-Schleife

```
while Bedingung:
    abhängige Anweisung;
    abhängige Anweisung;
    ...
keine abhängige Anweisung;
```

Der ganze `while`-Block wird solange wiederholt, wie die Bedingung wahr ist, unter Umständen also gar nicht, wenn die Bedingung schon am Anfang falsch ist. Als Bedingung kann jeder Ausdruck verwendet werden, der einen Wahrheitswert liefert:

```
while True:           die ewige Schleife
while False:         die niemals durchlaufene Schleife
while x <= 0:        x sollte irgendwann positiv werden, sonst endet das nie!
```

## Mit `break` aus der Schleife

Schleifen können auch mit der `break`-Anweisung verlassen werden. Meistens werden solche `break` Anweisungen in Verbindung mit einem `if` eingesetzt:

```
i = 1
j = 7
while True:           # scheinbar eine ewige Schleife
    print(i)
    i += 1
    if i == j:        # Abfrage: ist i gleich j
        break         # falls ja, beende die Schleife
```

Eine `break`-Anweisung beendet nur die innerste von mehreren ineinander geschachtelten Schleifen.

## Mit `continue` den Schleifendurchlauf abkürzen

Die `continue`-Anweisung innerhalb einer Schleife bricht den aktuellen Schleifendurchlauf ab und startet den nächsten Durchlauf, beendet aber nicht die Schleife selbst:

```
i = 0
while i < 20:
    i += 1          # das gleiche wie i = i+1
    if i % 3 == 0: # überspringe die Ausgabe für Vielfache von 3
        continue
    print(i)
```

Das Folgende funktioniert überhaupt nicht (warum?)

```
i = 1
while i < 20:
    if i % 3 == 0: # überspringe Vielfache von 3
        continue
    print(i)
    i += 1
```