

Schleifen

Schleifen

Schleifen wiederholen Code, der in ihnen enthalten ist. Jeder Durchlauf einer Schleife heißt "Iteration". Dabei gibt es zwei grundsätzliche Möglichkeiten: Schleifen prüfen mit jedem Durchlauf, ob eine Bedingung erfüllt/nicht erfüllt ist und brechen gegebenenfalls ab, oder sie zählen bis zu einem bekannten Wert und brechen dann ab. Die erste Möglichkeit wird durch eine While-Schleife realisiert.

While-Schleifen

In dem folgenden Beispiel ist eine Endlosschleife dabei. Das bedeutet, dass das Programm nie normal beendet wird. Du musst daher das Programm selbst beenden, in dem du auf den Stop-Knopf drückst, der neben dem Code oder in der oberen Menüleiste erscheint.

```
import time
i=0
while i < 10:
    i+=1 # Hier wird der Wert der Variablen i um 1 erhöht. Man kann das auch so schreiben: i = i
    + 1
    print(i)
print("Die Schleife ist durchgelaufen.")

# Dieser Code wird unendlich lange ausgeführt, da die Bedingung immer wahr ist.
while True:
    print("Schleifendurchlauf")
    time.sleep(0.1)
```

Schreibe in die Bedingung der zweiten `while`-Schleife andere Bedingungen, die immer wahr sind. Z. B. `1==1` oder `2+3==5`. Das Gegenteil von `True` ist übrigens `False`. Jede Bedingung, die überprüft wird, endet immer in entweder `True` oder `False`.

`while`-Schleifen eignen sich für die wiederholte Überprüfung von Bedingungen, bis die Abbruchbedingung erreicht ist und der Code hinter der Schleife ausgeführt wird. Beispiel:

```

i_am_bored = False
counter = 0
while not i_am_bored:
    counter += 1
    print(counter)
    if counter > 30:
        i_am_bored = True
print("I am bored!")

```

Hier ist eine verbesserte Version der Passwortabfrage aus dem Kapitel Bedingungen. Hier wird die Passwortabfrage in einer Schleife durchgeführt, solange, bis das Passwort richtig ist.

```

def password_correct(passwd):
    if passwd == "1234":
        return True
    else:
        return False

while not password_correct(input ("Passwort eingeben: ")):
    # überprüfe Passwort
    print("Du kommst hier nicht rein.")

print("Nun geht es los. ")

```

Ein realistischeres Beispiel wäre der Algorithmus zur Berechnung des größten gemeinsamen Teilers:

```

x,y = 1741446609,671084182
while x!=y:
    if x>y:
        x=x-y
    elif y>x:
        y=y-x
print(x)

```

For-Schleifen

Bei For-Schleifen ist beim ersten Durchgang schon klar, wie viele Iterationen es geben wird. Im ersten Beispiel wird von 0 bis 9 gezählt. Das erledigt die Funktion `range()`. In den anderen Beispielen wird bei jedem Durchgang ein Element einer Liste ausgegeben und beim letzten Beispiel

werden die Buchstaben im String `s` einzeln ausgegeben.

```
for i in range (10):
    print(i)
    pass
print("Die Schleife ist durchgelaufen.")

fruit = ["Apples", "Bananas", "Raspberries"]
for x in fruit:
    print(x)

## Iteration über einen String
s = "Theodor-Heuss-Schule"
for i in s:
    print(i)
```

Revision #14

Created 29 August 2022 16:10:43 by Admin

Updated 26 November 2025 08:04:41 by Marcus Jacobs