**L3\_2\_2 While-Schleife**

In manchen Fällen muss ein Programm immer wieder die gleichen Anweisungen viele Male hintereinander ausführen, bis ein bestimmtes Ergebnis eintritt. Wie oft man diese Anweisungen wiederholen muss, hängt dabei von einer Bedingung ab, welche am Anfang, bevor die Anweisungen ausgeführt werden, überprüft wird. Solche Wiederholungen nennt man auch Schleifen.

Ist die Anzahl der Schleifendurchläufe vor dem Beginn des Programms nicht exakt bekannt, benutzt man eine **while-Schleife**.

**Beispiel 1:**

Das folgende Programm simuliert einen Würfelvorgang und würfelt solange, bis eine 6 gewürfelt wurde:

1 from random import \*

2

3 zufallszahl = 0

4 while zufallszahl != 6:

5 zufallszahl = randint(1, 6)

6 print ("Wurf:", zufallszahl)

**Datei:** L3\_2\_2\_a\_while\_Schleife.py

*Hinweis:* *In diesem Programm kommen zwei Anweisungen vor, die benötigt werden, weil mit Zufallszahlen gearbeitet werden soll (nicht abiturrelevant!):   
In Zeile 1 wird ein Modul in das Programm eingebunden, das Zufallszahlenfunktionalität zur Verfügung stellt, in Zeile 4 wird dann eine Funktion des Moduls genutzt, die Funktion randint(1, 6), die eine ganzzahlige Zufallszahl aus dem Bereich 1 bis 6 erzeugt.*

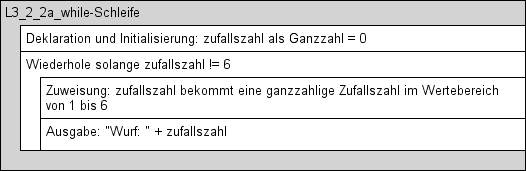
**Erklärung des Programmablaufs:**

In Zeile 3 erhält die Variable zufallszahl den Wert 0. Dann wird in einer while-Schleife der Variablen zufallszahl solange ein Zufallswert im Wertebereich von 1 bis 6 zugewiesen, bis zufallszahl=6 ist. Bei jedem Durchlauf wird der jeweilige Wert von zufallszahl ausgegeben.

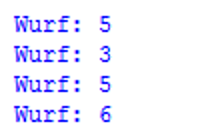
**Aufbau der while-Schleife in Python:**

Eine while-Schleife wird in Python mit dem Schlüsselwort while eingeleitet. Danach folgt die Bedingung, die erfüllt sein muss, damit die Schleife wiederholt wird, d.h. die Schleife wird wiederholt, solange die Bedingung erfüllt ist (hier im Beispiel: solange zufallszahl ungleich 6 ist). Abschließend folgt ein Doppelpunkt, und unter dem Doppelpunkt stehen **eingerückt** alle Anweisungen, die im Rahmen der Schleife ausgeführt werden sollen.

**Struktogramm:**

****

**Mögliche Ausgabe (je nach Würfelergebnis):**



**Beispiel 2:**

Andererseits lässt sich jede for-Schleife auch in eine while-Schleife übertragen. Wir vergleichen das Beispiel der for-Schleife aus der letzten Einheit mit der Umsetzung als while-Schleife.

Anwendungsbeispiel: Es soll ein Programm geschrieben werden, das die 5er-Reihe von   
5 bis 50 ausgibt:

|  |  |
| --- | --- |
| Quellcode: | |
| 1 for zahl in range(1, 11):  2 produkt = zahl \* 5  3 print(zahl, "mal 5 =", produkt) | 1 zahl = 1  2 while zahl < 11:  3 produkt = zahl \* 5  4 print(zahl, "mal 5 =", produkt)  5 zahl = zahl + 1  **Datei:** L3\_2\_2\_b\_while\_Schleife.py |
| Im Vergleich ist ein wesentlicher Unterschied festzustellen:  In der for-Schleife greifen wir auf eine festgelegte Zahlenliste zurück. Diese nutzen wir bei der while-Schleife nicht. Daher muss dort der Zähler zunächst deklariert und initialisiert werden. Ebenso muss in der while-Schleife der Zähler manuell verändert werden. | |
| Struktogramm: | |
|  | |