

J1	BPE 5: Grundlagen der Programmierung Informationsmaterial	Informatik
----	---	------------

L3_2_2 Die 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.

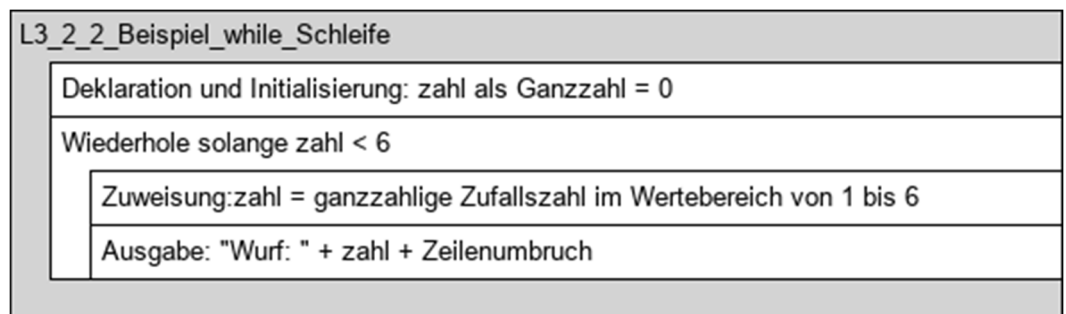
Ist die Anzahl der Schleifendurchläufe, bevor das Programm beginnt, nicht exakt bekannt, benutzt man eine **while-Schleife**.

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

Beispiel:

Mit Hilfe eines Programms soll ein Würfelvorgang simuliert werden. Es soll solange gewürfelt werden, bis eine 6 gewürfelt wurde.

Struktogramm:



Ausgabe:

Wurf: 3
Wurf: 5
Wurf: 2
Wurf: 6

Allgemeine Befehlsform while-Schleife

```

for (bedingung)
{
    Anweisungen, die solange wiederholt werden, solange die Bedingung wahr ist).
}

```

J1	BPE 5: Grundlagen der Programmierung Informationsmaterial	Informatik
----	---	------------

Pogrammcode

```

<!DOCTYPE html>
<html lang="de">
  <head>
    <title> while-Schleife</title>
    <script>
      var zahl = 0;
      while (zahl < 6) {
        zahl = Math.round((Math.random() * (6 - 1)) + 1);
        document.write("Wurf: "+zahl + "<br>");
      }
    </script>
  </head>
  <body>
  </body>
</html>

```

Datei: Beispiele_HTML/L3_2_2_while_Schleife.html

Zuerst erhält eine Variable *zahl* den Startwert 0. Dann wird dieser Variablen in einer Schleife mit der Funktion *Math.random()* solange *zahl* eine Zufallszahl im Wertebereich von 1 bis 6 zugewiesen, bis der Wert der Variablen *zahl* 6 ist. Der jeweilige Wert der Variablen *zahl* wird bei jedem Durchlauf in einem Dialogfenster ausgegeben.

Hinweis: Die Funktion *Math.random()* gibt eine Zufallszahl in Form einer Gleitkommazahl im Bereich von 0 bis 1 (0 inklusiv, nicht aber 1) zurück.

Die Funktion *Math.round()* gibt die nächstgelegene ganze Zahl einer Gleitkommazahl zurück.

z.B.: 3.49 → 3 3.5 → 4