



#### Lektion 7: micro:bit mit Python

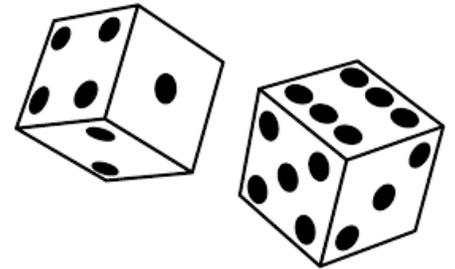
#### Anwendung: Wurf zweier Würfel

In dieser Anwendung wird ein Programm entwickelt, mit dem Daten unter Verwendung des micro:bit gesammelt werden. Während das Programm läuft, beobachtet man, wie ein Punktdiagramm auf dem geteilten Display des TI-nspire wächst.

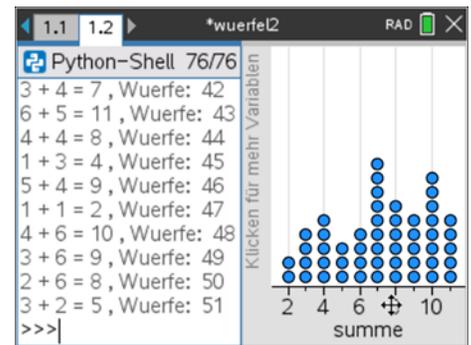
#### Lernziele:

- Schreiben eines Datenerfassungsprogrammes für den micro:bit
- Erstellen eines dynamischen Data & Statistic-Diagrammes der gesammelten Daten

1. Diese Anwendung beruht auf den letzten drei micro:bit-Übungen: Schreiben eines Programmes, das eine Geste wie "Shake" (oder einen Tastendruck) verwendet, um einige Daten zu sammeln, speichern der Liste als TI-Nspire-Variablen und...



2. ... dann die Einrichtung einer TI-nspire-Seite, so dass
  - das Python-Programm auf dem einen Teil der Seite läuft (die Python Shell) und
  - auf der anderen Seite sich ein Data- & Statistic-Plot aufbaut.



3. Beginnen Sie Ihr micro:bit-Programm mit den üblichen Importen einschließlich des Zufallsmoduls und einer leeren Liste namens **summe = [ ]**

Speichern Sie diese Liste sofort in einer TI-Nspire-Variablen mit demselben Namen), die dann zunächst also auch leer ist:

**store\_list("summe", summe)**

**print()** erteilt eine Anweisung an den Benutzer, bevor die Schleife beginnt. Es wird die "Shake"-Geste verwendet, um die Würfel zu würfeln.

```

zwei_wuerfel.py
from microbit import *
from random import *

summe=[]
store_list("summe",summe)
print("Wuerfelwurf: micro:bit schuettern")

while get_key() != "esc":
  +=|

```

4. Verwenden Sie in der **While**-Schleife diese Geste, um
  - zwei Würfel zu werfen (es werden zwei zufällige ganze Zahlen erzeugt.)
  - **addieren** Sie sie
  - **hängen** Sie die Summe an die **Summenliste** an
  - **drucken** Sie die beiden Würfelwerte, ihre Summe und die Wurfnummer (Anzahl der Würfe bisher) auf dem TI-Nspire-Bildschirm. Hinweis: **len(summe)** ist die Anzahl der Listenelemente und damit die Wurfnummer
  - **anzeige** beider Würfel-Werte auf dem micro:bit
  - **speichern** der **Liste** in einer TI-Nspire-Variablen



5. So sieht der Würfelwurf aus (alternativ kann auch ein Tastendruck verwendet werden):

```

◆ ◆ if accelerometer.was_gesture("shake"):
◆ ◆ ◆ display.clear()
◆ ◆ ◆ w1 = randint(1,6)
◆ ◆ ◆ w2 = randint(1,6)

```

*Achten Sie auf die Einrückungen.*

6. Bilden Sie die Summe **s** und fügen (**append**) Sie die Summe als neues letztes Element der Liste der Summen hinzu:

```

s = w1 + w2
summe.append(s)

```

7. Zeigen Sie die beiden Würfe auf dem micro:bit-Display an. Denken Sie daran, dass die beiden Würfe möglicherweise den gleichen Wert haben; daher muss durch eine Pause sichergestellt werden, dass beide tatsächlich angezeigt werden.:

```

display.clear()
display.show(w1)
sleep(250)
display.clear()
display.show(w2)
sleep(250)

```

*Eine andere Verzögerung der sleep()-Anweisung ist u.U. sinnvoll. Versuchen Sie jetzt, das Programm auszuführen und schütteln Sie das micro:bit. Auf dem micro:bit sollten nacheinander zwei Zahlen angezeigt werden.*

```

1.1 *wuerfel2 RAD 18/18
*zwei_wuerfel.py
[
while get_key() != "esc":
  if accelerometer.was_gesture("shake"):
    display.clear()
    w1=randint(1,6)
    w2=randint(1,6)

```

```

1.1 *wuerfel2 RAD 20/20
*zwei_wuerfel.py
while get_key() != "esc":
  if accelerometer.was_gesture("shake"):
    display.clear()
    w1=randint(1,6)
    w2=randint(1,6)
    s=w1+w2
    summe.append(s)

```

```

1.1 *wuerfel2 RAD 30/30
*zwei_wuerfel.py
display.clear()
display.show(w1)
sleep(250)
display.clear()
display.show(w2)
sleep(250)

```

8. Mit einer einzigen **print()** - Anweisung können die einzelnen Würfelwerte, ihre Summe und die Wurfnummer auf dem TI-Schirm ausgegeben werden:

```
print (w1, "+", w2,"=",s," ", "Wuerfe: ", len(summe))
```

Man erhält die nebenstehende Abbildung.

```
>>>#Running zwei_wuerfel.py
>>>from zwei_wuerfel import *
Wuerfelwurf: micro:bit schuettern
6 + 6 = 12 , Wuerfe: 1
3 + 1 = 4 , Wuerfe: 2
6 + 5 = 11 , Wuerfe: 3
6 + 4 = 10 , Wuerfe: 4
1 + 1 = 2 , Wuerfe: 5
3 + 3 = 6 , Wuerfe: 6
4 + 1 = 5 , Wuerfe: 7
>>>
```

9. Nun muss noch die Python-Liste **summe** in einer TI-nspire-Liste gleichen Namens abgespeichert werden:

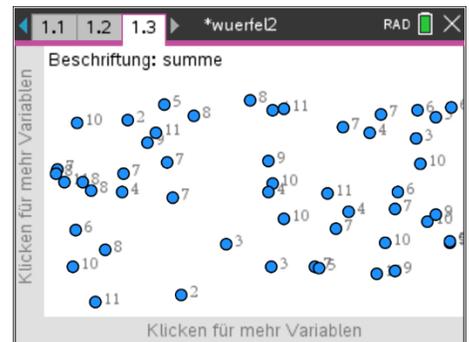
```
◆◆◆◆store_list("summe", summe)
```

Jedes Mal, wenn gewürfelt wird, werden diese Listen aktualisiert, da die Anweisung am Ende des **while**-Blockes steht.

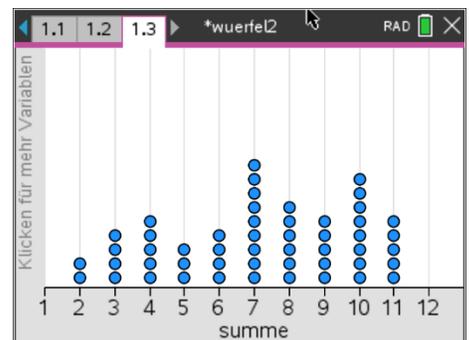
```
◆◆◆◆
◆◆◆◆display.clear()
◆◆◆◆display.show(w1)
◆◆◆◆sleep(250)
◆◆◆◆display.clear()
◆◆◆◆display.show(w2)
◆◆◆◆sleep(250)
◆◆◆◆
◆◆◆◆
◆◆◆◆print(w1,"+",w2,"=",s," ", "Wuerfe: ", len(summe))
◆◆◆◆store_list("summe",summe)
```

10. Wenn Sie überzeugt sind, dass Ihr Programm ordnungsgemäß funktioniert, können Sie Ihr Python-Programm mit den TI-Nspire-Plotfunktionen verbinden. Führen Sie Ihr Programm aus und generieren Sie etwa 50 Würfe. Drücken Sie **[esc]**, um das Programm zu beenden.

In der Python Shell (also beim Prompt **>>>**) kann man durch Drücken von **[ctrl] [doc]** oder **[ctrl] [I]** eine Seite einfügen, in diesem Fall **Data & Statistics**. Man sollte ein Bild wie nebenstehend sehen mit über den Schirm verteilten Datenpunkten..



11. Klicken Sie unten auf dem Bildschirm auf die Meldung 'Klicken Sie, um eine Variable hinzuzufügen' und wählen Sie die Listenvariable **summe** aus. Ihre verstreuten Datenpunkte sind nun entlang der x-Achse entsprechend ihrem Wert organisiert und das Fenster ist für die Daten entsprechend zugeschnitten. Dies ist ein **Punktdiagramm**.





12. Mit **[ctrl] [nach links]** kommt man wieder zur Python Shell. Drückt man nun **[ctrl] [4]**, so erhält man einen geteilten Schirm mit der Python Shell links und dem Data&Statistics – Bild rechts.

13. Die Shell wurde "neu initialisiert", sodass das Programm durch Drücken von **[ctrl] [R]** nicht erneut ausgeführt wird. Gehen Sie zurück zum Python-Editor und drücken Sie **[ctrl] [R]**, um das Programm auszuführen. Es wird in der Shell mit halbem Bildschirm ausgeführt, wie hier gezeigt. Sie sehen zunächst "Keine numerischen Daten" auf der rechten Seite, da das Programm mit einer leeren Liste startet.

Wenn Sie nun Daten sammeln (schütteln Sie das micro:bit, um die Würfel zu werfen), werden Ihre **Summenwerte** als Punkte in der Data & Statistics-App auf der rechten Seite angezeigt.

Durch Drücken von **[esc]** wird das Programm beendet und Sie können viele andere 1-Variablen-Datenanalysen in der TI-Nspire-Umgebung durchführen.

Wenn Sie **[Strg] [R]** erneut (jetzt in der Python-Shell) drücken, wird das Programm auch erneut ausgeführt.

*Tipp: Um die Shell zu Beginn jedes Laufs zu löschen, fügen Sie zu Beginn Ihres Programms die Anweisung hinzu:*

**clear\_history()**

gefunden auf **[menu] > Weitere Module > BBC micro:bit > Commands**

Viel Spaß und denken Sie daran, Ihr Dokument zu speichern!

