

**Lektion 5 : Verwendung des Moduls ti\_system**

**Übung 1 : Arbeiten mit Listen**

In dieser ersten Übung der Lektion 5 wird gezeigt, wie man mittels des Moduls **ti\_system** Listen in Python importieren und auch aus Python exportieren kann.

**Lernziele :**

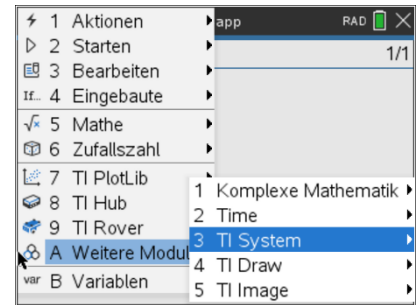
- Importieren, exportieren und erstellen von Listen.
- Wiederholung der Grafikbefehle aus Lektion 4.

Das Modul **ti\_system** enthält Befehle zur Kommunikation mit dem grafischen Taschenrechner.

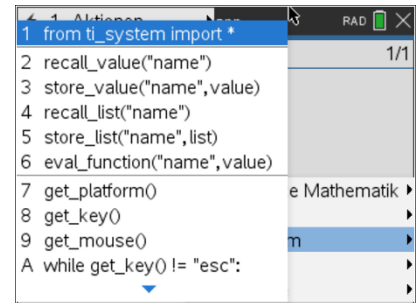
In dieser ersten Übung werden davon nur zwei verwendet :

`var=recall_list(« name »)` und `store_list(« name »,var)`

Die anderen Befehle werden in späteren Übungen verwendet.



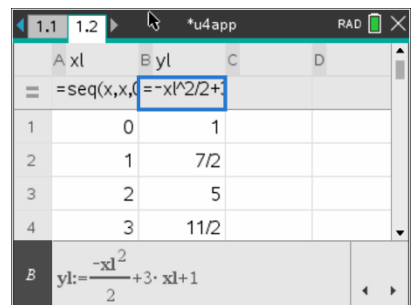
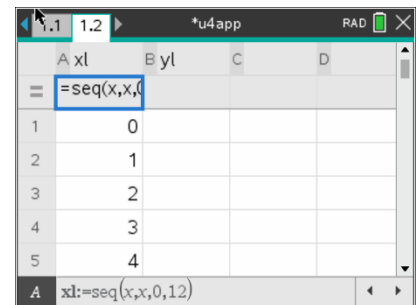
Zuerst sollen zwei Listen im Taschenrechner erzeugt und dargestellt werden, bevor sie in ein Python-Programm importiert werden.



**1 : Eingeben von Werten in den Taschenrechner**

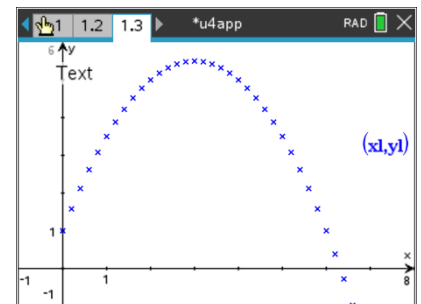
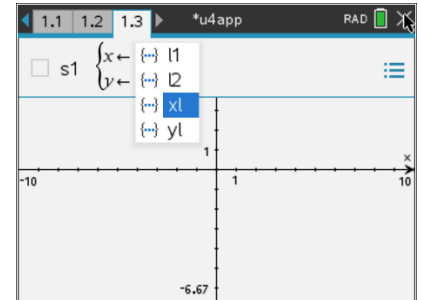
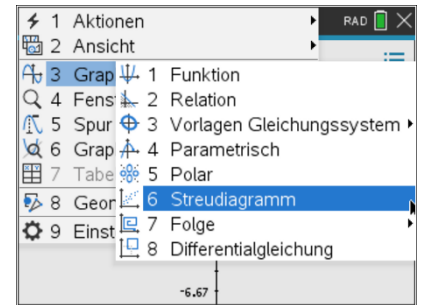
**a) Erstellen von zwei Listen x1 und y1 :**

- Dazu wird eine neue Seite aus Lists & Spreadsheet eingefügt.
- Die Spalten werden in den Spaltenköpfen benannt.
- In **x1** wird über die abgebildete Anweisung die Folge der natürlichen Zahlen von 0 bis 12 erzeugt. Die Anweisung kann auch einfach mit der Tastatur geschrieben werden.
- Die Liste **y1** enthält die Werte der Funktion  $x \mapsto -x^2/2 + 3x + 1$ . Dabei ist darauf zu achten, dass statt **x** die Bezeichnung der Listenspalte **x1** verwendet wird.



**b) Grafische Darstellung**

- Über die neue Seite **Graphs** gelangt man in ein Menü, in dem **Streudiagramm** ausgewählt werden muss.
- Für **s1** muss dann die x- und y-Liste eingetragen werden. Hier hilft ein Druck auf **var** , da man dann die Listennamen bequem auswählen kann.
- Mit den Fenstereinstellungen  $x_{min}=-1$   $x_{max}=8$   $y_{min}=-1$  und  $y_{max}=6$  erhält man das untere Bild.



**2) Importieren** der Werte in ein Python-Programm :

- Anlage eines neuen Programmes mit dem Namen **U5SB1**.
- Die Module **ti\_system** und **ti\_plotlib** importieren (Reihenfolge egal).
- Zwei leere Listenvariablen **xx** und **yy** erzeugen.
- Die leeren Listen **xx** und **yy** füllen mit den Inhalten von **x1** und **y1**. Dazu wird der Befehl **var=recall\_list(« name »)** aus dem Modul **ti\_system** verwendet. Der « name » ist der **Listenname**.
- Lässt man nun das Programm laufen, so kann man sich durch Drücken von **var** oder direkter Eingabe des Listennamens den Inhalt der Listen **xx** und **yy** ansehen.

```

1.1 1.2 1.3 *u4app RAD 6/10
U5SB1.py
import ti_plotlib as plt
from ti_system import *
xx=[]
yy=[]
xx=recall_list("x1")
yy=recall_list("y1")

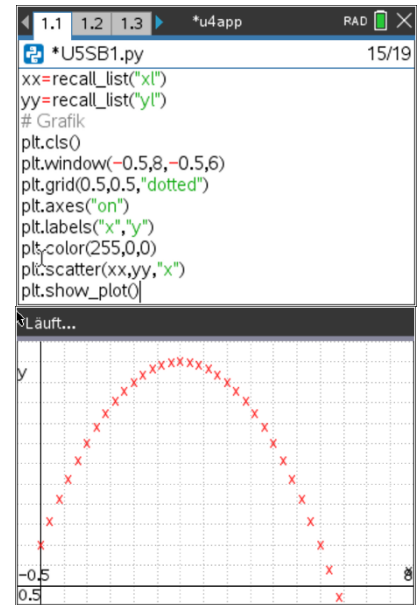
```

**Lehertipp** : Die Erstellung von leeren Listen `xx=[]` und `yy=[]` ist nicht unbedingt erforderlich, da sie beim Abruf der Listen automatisch erstellt werden. Es ist jedoch sinnvoll, Gewohnheiten beizubehalten, die notwendig sind, wenn das **ti\_system**-Modul nicht verwendet wird.



### 3) Grafische Darstellung

- Zunächst einmal muss die grafische Darstellung eingerichtet werden wie abgebildet.



- Dann kann man das Programm laufen lassen.

### 4) Exportieren von Werten aus einem Python-Programm

- Ein neues Programm mit dem Namen **U5SB11** anlegen.
- tiplotlib**, **ti\_system** und **math** importieren (Reihenfolge beliebig).
- Die Funktion **data(a,b,h)** programmieren.
- Es werden zwei leere Listen **x** und **y** angelegt, die später Werte aus dem Intervall **[a ; b]** enthalten, die mit der Schrittweite **h** erstellt werden.

```

*U5SB11.py
from ti_system import *
from math import *
def data(a,b,h):
  x=[]
  y=[]
  for i in range(a,b,h):
    x.append(i)
    y.append(sqrt(x[i]))
  store_list("xa",x)
  store_list("ya",y)
  
```

**Lehertipp** : Durch die Erstellung von zwei leeren Listen wird vermieden, dass bei der Ausführung des Programmes ein Fehler gemeldet wird.

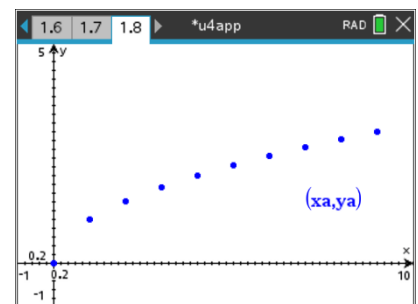
- Liste **y** enthält die Quadratwurzel der Werte aus Liste **x**.
- Die Listen werden nun mit einer **For**-Schleife gefüllt. Die **append**-Befehle bewirken, dass neue Elemente an die Liste angehängt werden. Die Listen haben also eine dynamische Länge.
- Die Listen werden mit dem Befehl **store\_list(« name »,var)** in den Taschenrechner übertragen. **var** ist der Listenname im Programm, **« name »** der frei wählbare Listenname im Taschenrechner.

```

Python-Shell
>>>#Running U5SB11.py
>>>from U5SB11 import *
>>>data(0,10,1)
>>>|
  
```

**Hinweis: Vorsicht** mit der Einrückung, denn die **store\_list**-Anweisungen müssen nicht in der Schleife stehen. Entweder mit zweimal **[del]** oder den Einzuanweisungen aus dem Menü **Bearbeiten** lässt sich der Einzug verändern.

- Nun kann man das Programm laufen lassen. Scheinbar tut sich nichts.
- Erst wenn man Python verlässt, kann man sich das Ergebnis bei **Graphs** ansehen.



**Lehertipp** : Die Ausgabe in Listen wird interessant im Zusammenhang mit den Sensoren des Microcontrollers **TI-Innovator & TI-Rover**.