Lektion 4 : Verwendung des Moduls ti_plotlib

Übung 1 : Eine erste Grafik

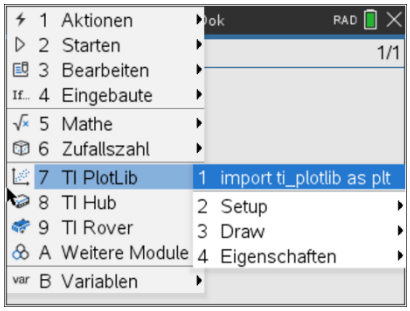
In dieser ersten Übung in Lektion 4 wird gezeigt, wie man Anweisungen zum Erstellen von Grafiken in Python schreibt und verwendet. Außerdem wird gezeigt, wie man eine Grafik zeichnet und die Anzeige konfiguriert.

Lernziele :

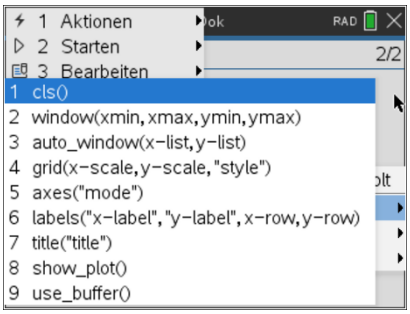
- Entdeckung des Moduls **ti_plotlib**.
- Darstellung eines Punktes und einer Strecke.
- Einstellungen einer Grafik.

1 : Das Modul ti_plotlib

Um eine Grafik darstellen zu können, muss das Programm in der Lage sein, grafische Anweisungen zu verstehen. Diese Grafikanweisungen finden sich im **Modul ti_plotlib**. Es muss deshalb wie alle Module am Anfang des Programmes eingefügt werden.

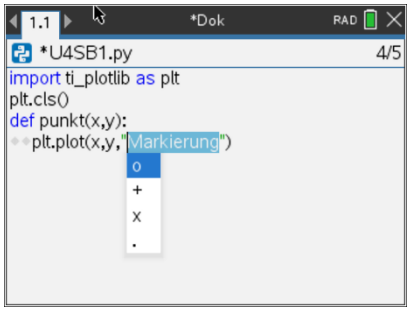


Ein neues Programm mit dem Namen **U4SB1** wird angelegt, dem das Modul **ti-plotlib** hinzugefügt wird. Dazu muss man im **Menü 7 ti_plotlib…** wählen und dann **1 import ti_plotlib as plt** importieren.

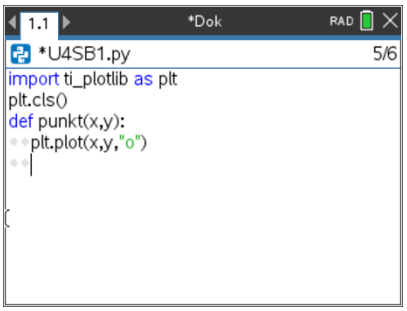


In diesem ersten Teil soll ein Programm geschrieben werden, das einen Punkt anzeigt, dessen Koordinaten bekannt sind. Anschließend wird das Programm ergänzt, so dass der Punkt eine andere Farbe bekommt und in einem Achsenkreuz dargestellt wird. Die Achsen werden beschriftet und eine Überschrift wird eingefügt.

Zuerst wird eine Funktion mit den Koordinaten eines Punktes als Argument definiert und dieser Punkt wird angezeigt.

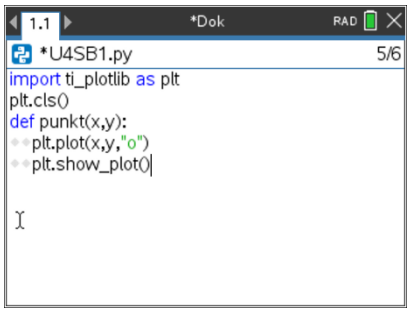


- Mit der Anweisung **plt.cls ()**, die man im Modul **ti_plotlib** im Menü **SetUp** findet (**1 cls ()**), wird der Bildschirm „gereinigt“.
- Um den Punkt zu zeichnen, wählen man die Anweisung **plot(x,y,“mark“)** im Menü **Draw** des **ti-plotlib**-Moduls.
- Jetzt kann die gewünschte Markierung ausgewählt werden (hier ein kleiner Kreis).



Lehrertipp: Die Darstellung eines Punktes in Form eines Pixels sollte gewählt werden, wenn eine große Anzahl von Punkten dargestellt werden soll.

Durch den abschließenden Befehl **plt.show_plot()** aus dem **SetUp** von **ti_plotlib** wird der Punkt gezeichnet :

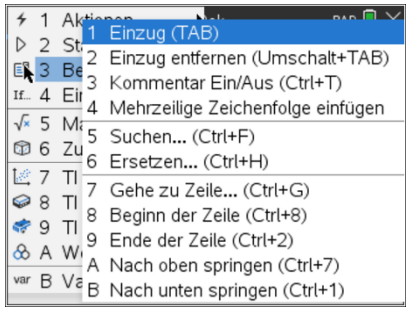


- Mit dem **Menü Starten** sollte zunächst das Programm auf Fehler überprüft und gespeichert werden, bevor man das Programm startet. Dann erhält man durch Drücken von [var] die Funktion **punkt()**, die man mit [enter] bestätigen muss.
- Anschließend kann man die Koordinaten eines Punktes eingeben. Dabei erfolgt ein Wechsel von der Shell in den Grafikmodus. Der dargestellte Punkt wurde durch **punkt(2,3)** erzeugt.
- Um einen neuen Punkt zu zeichnen, muss man den Grafik-Bildschirm durch Drücken von [enter] verlassen. Mit [var] oder Steuerkreuz [▲] kann man dann einen neuen Punkt wie oben beschrieben eingeben.
- Der Punkt **punkt(10,10)** ist nicht mehr sichtbar, da er außerhalb des Darstellungsbereiches liegt.

Lehrertipp: Wenn man ein Programm mit den Grafikfunktionen schreibt, sollte man die Parameter des Grafikfensters angeben und möglicherweise ein Koordinatensystem, ein Raster, einen Achsennamen usw. anzeigen lassen.

2 : Einstellungen

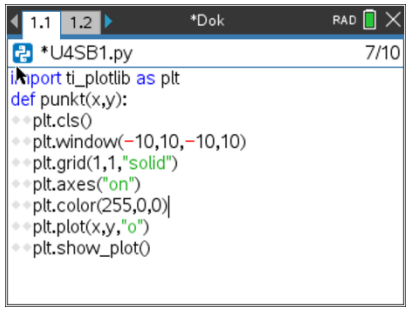
Es ist sinnvoll, dass man den Befehl **plt.cls()** in die Definition der Funktion übernimmt, um einen « sauberen » Grafikbildschirm zu bekommen.



Lehrertipp: Um eine Zeile auszuschneiden, zu kopieren oder einzufügen, verwendet man Anweisungen aus dem Menü **3 Bearbeiten.**

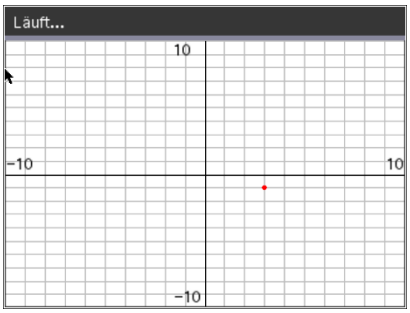
In das Programm sollten aus dem **SetUp** die folgenden Befehle übernommen werden :

- Die Fenstergrenzen : X_{min} = -10 ; X_{max} = 10 ; Y_{min} = -10 et Y_{max} = 10 (**window**).
- Die Art des Rasters (**grid**) kann man selbst bestimmen.
- Die Koordinatenachsen (**axes**).
- Die Farbe des Punktes (Menü **Draw,** dann **color(r,g,b)**).

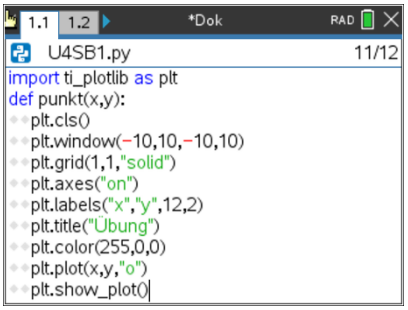


Lehrertipp: Die Farbe eines Punktes oder einer Linie muss im Code r, g, b (rot, grün, blau) angegeben werden, wobei jeder Parameter einen Wert aus dem Intervall [0; 255] annehmen kann. Die Farben sind auf 8 Bits oder 2^8 = 256 Möglichkeiten codiert, wobei die 0 mitgezählt wird, was einem Fehlen der Komponente r oder g oder b entspricht.
Es ist auch möglich, das Raster in Farbe zu zeichnen, indem man die Rasteranweisung **plt.grid(xscl, yscl, "style", (r, g, b))** verwendet.

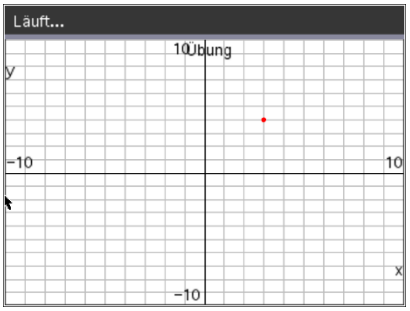
Mit P(3/-1) sollte sich das nebenstehende Bild ergeben.



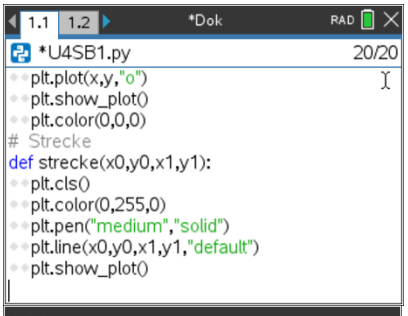
Nun können die Achsen noch benannt werden. Dazu muss der Befehl **plt.labels** eingefügt werden. Die Achsenbeschriftungen erfolgen für die y-Achse linksbündig und für die x-Achse rechtsbündig in der jeweils angegebenen Zeile.
Das sieht nicht immer glücklich aus, da sich z.B. auch Überschneidungen ergeben können. Hier muss man ev. etwas experimentieren.



Dann kann man mit **plt.title** der Grafik noch einen Titel geben, der standardmäßig zentriert dargestellt wird.



Erweiterung: Man kann das Programm ergänzen, indem man eine Funktion schreibt, mit der eine Strecke grafisch dargestellt werden kann. Die wesentlichen Anweisungen sind im Bild rechts dargestellt. Natürlich können noch Achsen, Raster, usw. hinzu gefügt werden.
Mit dem Befehln **plt.pen**, auf den über das Menü **Draw** zugegriffen werden kann, kann man das Aussehen der Strecke bestimmen.



Um die Strecke zu zeichnen, muss man bei [var] **strecke()** anwählen und dann die Werte eingeben – im Beispiel (-4,3,3,-4).

Lehrertipp: Im Editor kann man auch einen Kommentar angeben. Dieser wird grau geschrieben und mit einem vorangestellten #, was bedeutet, dass diese Anweisung nicht ausgeführt wird.
Das Zeichen # kann über [ctrl] eingefügt werden.