Lektion 5 : Verwendung des Moduls ti_system | **Übung 3 : Anzeigen und Zeiten**

In dieser dritten Übung in Lektion 5 wird gezeigt, wie man die Anzeige- und "Timer"-Optionen im Modul **ti_system** verwendet.

Lernziele :

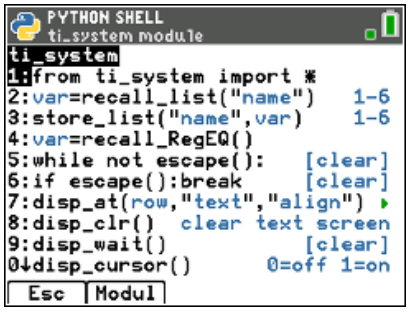
- Die Funktionsweise der **disp…**-Befehle.
- Verwendung dieser Befehle in einem Programm mit anderen Python-Befehlen.

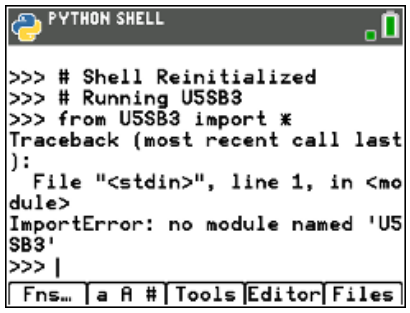
---

1 : Die Gruppe der disp – Befehle

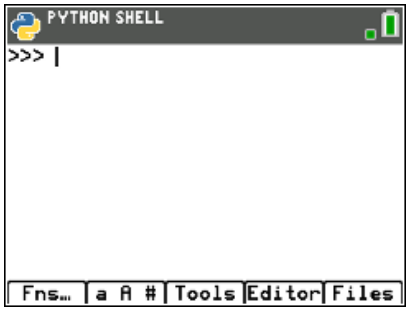
In dieser Übung geht es um Textausgaben sowie verschiedene Befehle, die den Programmablauf steuern.

- Erstellen eines neuen Programmes mit Namen U5SB3.
- Das Modul **ti_system** importieren.
- Den Befehl **disp_clr()** einfügen.
- Drücken von **f5** und wechseln in die Shell **f4**. Das Programm noch nicht ausführen !
- Beobachten der Textzeilen oder Meldungen in der Shell.



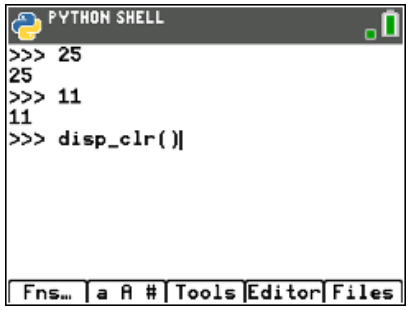


- Nun das Programm aufrufen und ausführen. Man sollte ein Bild wie nebenstehend erhalten.



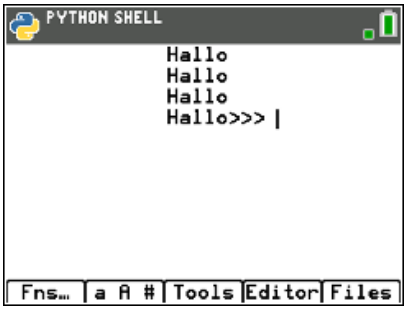
Der Befehl **disp_clr()** löscht den Inhalt des Bildschirms und setzt den Cursor oben in die erste Zeile des Bildschirms. Man erzeugt dadurch innerhalb eines Programmes einen Bildschirm, der von allen vorherigen Anzeigen frei ist.

Der Befehl lässt sich auch außerhalb eines Programmes in der Python-Shell ausführen und bewirkt ebenfalls ein « Säubern » des Bildschirmes.

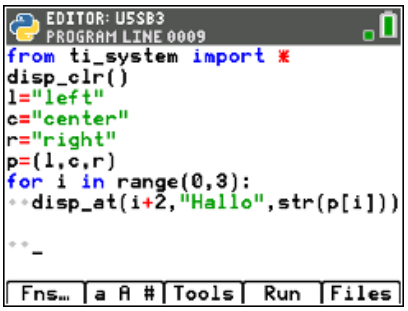


- Das Programm soll nun durch eine For-Schleife ergänzt werden, die den Befehl **7 : disp_at(row, « txt », « align »)** aus dem Modul **ti_system** enthält. Dier Befehl schreibt in der i. Zeile den angegebenen Text linksbündig, mittig oder rechtsbündig.
- **Hinweis** : Man kann nicht in die 0. Zeile schreiben, da die Zeilenzählung mit 1 (oberste Zeile) beginnt.

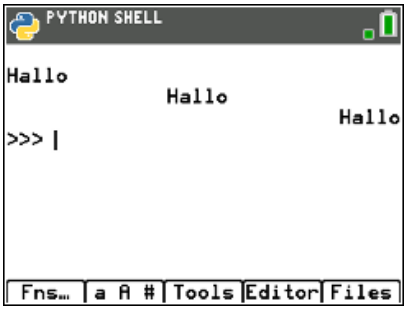
- So sieht die Ausgabe des Programmes nun aus.



Das Programm soll nun noch um die Stringvariablen **l, c, r** ergänzt werden. Sie werden in einer Liste **p** (Position) zusammen gefasst.
Bei der Ausführung des Programmes sollte das Wort in drei Zeilen verschoben dargestellt werden, denn der Schreibindex i des Wortes wird verschoben gemäß der Variablen in der **p**-Liste.

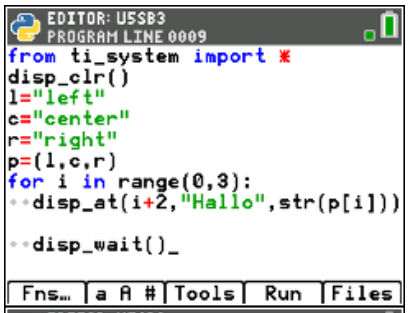


Ein Vorschlag zur Erweiterung: grafische Gedichte
.

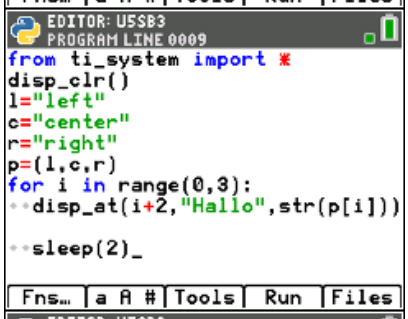


Der Befehl **disp_wait()** unterbricht die Ausführung des Programmes. Durch Drücken von clear wird die Ausführung des Programmes wieder aufgenommen.

Lehrertipp : Diese Befehl ist interessant, um den Schritt-für-Schritt-Ablauf eines Programmes zu beobachten, das eine Schleife enthält.



Löscht man die Anweisung **disp_wait()** und ersetzt sie durch die Anweisung **sleep(seconds),** so pausiert das Programm für die angegebene Zeit.
Auf diese Weise wird das vorgeschlagene Wort alle 2 Sekunden an der Position angezeigt, die dem Wert der **p[i]** Liste entspricht.



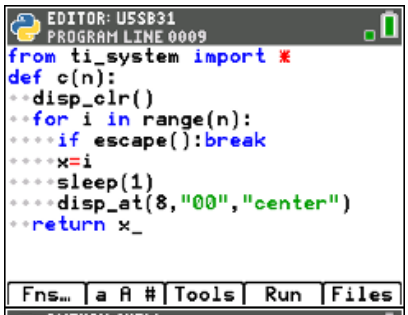
Der Befehl **disp_cursor(0)** bewirkt, dass kein Cursor in Form eines senkrechten Striches in « Hallo » erscheint – erst nach Beendigung des Programmes ist er wieder vorhanden.

disp_cursor(1) lässt den Cursor wieder erscheinen.

2 : Die escape() – Befehle

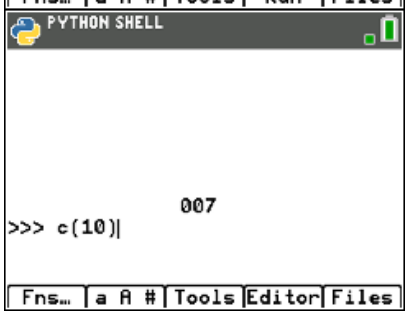
Der Befehl **if escape() : break** hält die Ausführung eines Programmes an, wenn man die Taste clear drückt.

- Ein neues Programm USB531 anlegen.
- Die Befehle wie rechts abgebildet einfügen.
- Solange clear nicht gedrückt wird, wird **i** in **x** abgespeichert.
- Während der Pause von **1 s** Länge wird der Text « **00** » angezeigt.
- Dieser Text wird bei Programmende noch durch den Wert von **x** ergänzt.



Während das Programm läuft, sieht man ein sehr leichtes Flackern des Bildschirms.

Die Anweisung **while not escape()** ermöglicht eine Schleife, deren Befehle solange ausgeführt werden, bis die Taste clear gedrückt wird.



Lehrertipp : Die Verwendung dieser Anweisungen ist sehr hilfreich bei der Verwendung von Programmen, die den TI-Innovator und den TI-Rover nutzen.