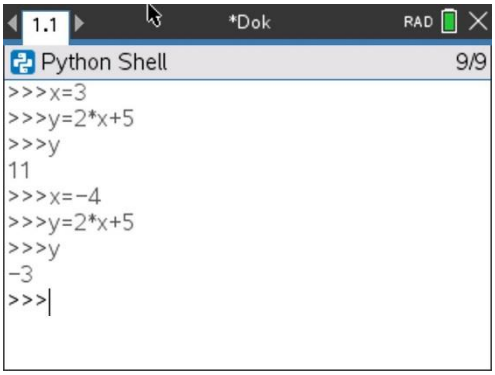
Lektion 1: Erste Schritte inPython | **Übung 3 : Funktionen in Python**

In dieser dritten Übung in Lektion 1 erstellen Sie mit dem Programmeditor Funktionen und führen sie in der Shell aus.

Lernziele :

- Notation von Funktionen in Python
- Erstellen einer Funktion

Funktionen in Python

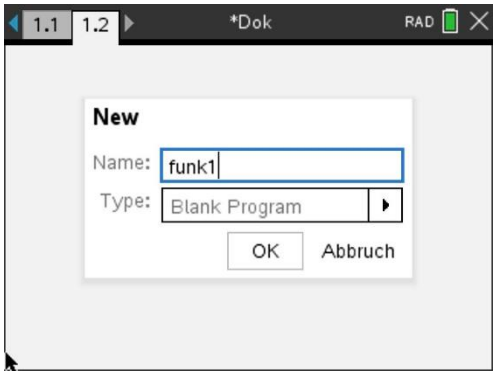


In der Shell lässt sich mit den folgenden zwei Befehlen der Wert von y schnell berechnen :

$$x = 3$$
$$y = 2x + 3$$

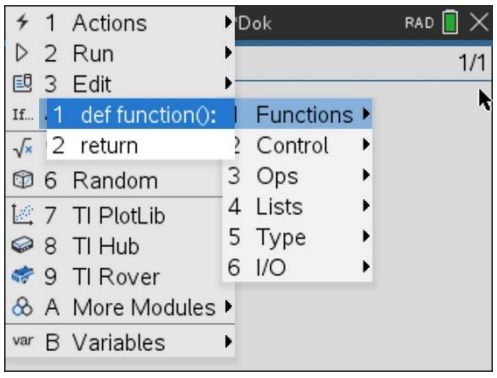
Möchte man jedoch y für andere Werte von x berechnen, so muss man diese zwei Befehle wieder neu eingeben.

Mit einer **Funktion** hingegen kann man sich diese Arbeit erheblich vereinfachen.



Eine Funktion kann als eine Folge von Anweisungen betrachtet werden, die eine bestimmte Aufgabe unter Verwendung eines oder mehrerer Argumente ausführen. Sie stellt also einen Algorithmus dar. Sie erhält stets einen Namen.

- Beim Start von Python ist nun „**New**“ anstelle von „**Shell**“ zu wählen. Man ist dann im Programmiermodus.
- Dann muss das Programm benannt werden (hier: **„funk1**“).
- Die **Programmierung einer Funktion** beginnt immer über das Menü „**Built-ins … Functions … def function(x)**“. Die Funktion erhält einen **Namen**, gefolgt von ihren **Argumenten**. Diese Zeile **endet** mit dem **Symbol:**
- Die beiden Punkte in der nächsten Zeile markieren den **Beginn des Anweisungsblocks,** der die Funktion **definiert.** Alle diese Anweisungen werden **eingerückt**, d.h. in Bezug auf die erste Zeile automatisch nach rechts um 2 Stellen verschoben und dadurch als zusammenhängender Anweisungsblock kenntlich gemacht



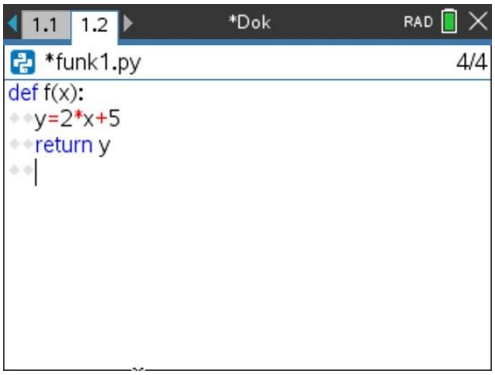
- Die Funktion gibt über den Befehl ***return*** (aus „**Built-ins … Functions**“) das **Ergebnis** (Wert einer oder mehrerer Variablen) zurück. Das Ergebnis kann auch aus einer Liste von Ergebnissen oder einer Zeichenfolge bestehen.
- Die **Einrückung** ist wichtig. Alles, was nach ***def ()*** eingerückt wird, wird als **Block** ausgeführt. **Die Einrückung darf während des Blockierens nicht variieren, die Anzahl der Leerzeichen muss unbedingt gleich bleiben!**

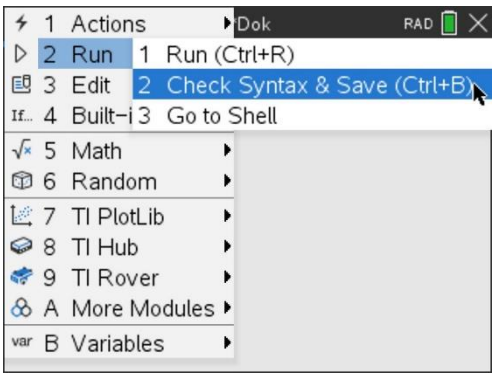
Hinweis für Lehrkräfte : Eine Funktion ermöglicht es, das untersuchte Problem in Unterprobleme zu unterteilen und so die Wiederholung von Anweisungen zu vermeiden. Einmal definiert, kann sie während der Ausführung des Programms so oft wie nötig "aufgerufen" werden.

Eine Funktion kann auch keine Argumente haben. Sie kann auch in einem anderen Programm aufgerufen werden: Es reicht aus, sie in eine Anweisung einzufügen, indem der Name und die Werte der Argumente eingegeben werden.

Zurück zum Anfangsbeispiel

- Nachdem Sie einen Namen vergeben haben, können Sie die Funktion endgültig definieren und den Anweisungsblock schreiben. Der Editor rückt dabei automatisch ein, wenn Sie durch enter eine neue Zeile beginnen. Rechenzeichen werden rot hervorgehoben.
- Mit dem Befehl return wird die Funktion abgeschlossen.
- Sie sollten sich angewöhnen, das Programm vor dem ersten Lauf auf Syntaxfehler zu überprüfen und abzuspeichern. Dazu dient der Menüpunkt « **Run … Check Syntax & Save** ») mit dem Shortcut **Ctrl+B**.
- Dann kann man das Programm durch « **Run … Run** » bzw. **Ctrl+R** starten..





- Python wechselt in die Shell und macht dazu ein **neues Fenster** auf.
- Drückt man die Taste var , so erhält man eine Liste der im Programm verwendeten Funktionen. Man kann die passende Funktion durch enter in die Shell kopieren.
- Dann muss nur noch das Argument eingefügt werden. Mit enter wird die Berechnung ausgeführt.
- Muss man am Programm etwas verändern (korrigieren oder ergänzen), so wechselt man in das entsprechende Fenster.