

Lektion 1: Erste Schritte in Python

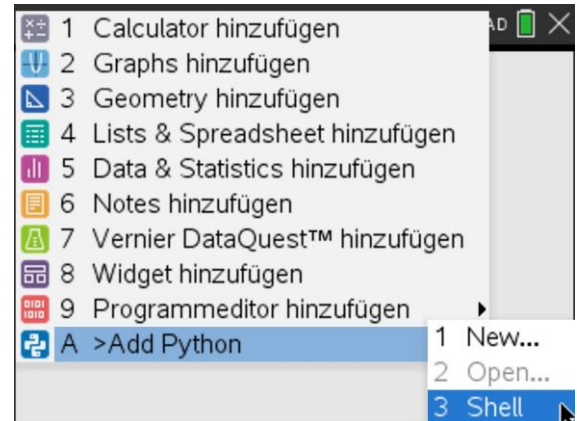
Übung 1: Arbeiten mit der Shell

In dieser ersten Übung in Lektion 1 lernen Sie die Shell der Programmiersprache TI-Python anhand mehrerer Beispiele kennen

Lernziele :

- Verwendung des TI-Python
- Entdeckung der mathematischen Funktionen in Python
- Unterscheidung zwischen dem Programmeditor und der Konsole (Shell)
- Verwendung einer Programmieranweisung in der Konsole

- Starten Sie ein neues Dokument und fügen Sie die Seite « **>Add Python** » hinzu.
- Wählen Sie aus dem Submenü « **Shell** ».

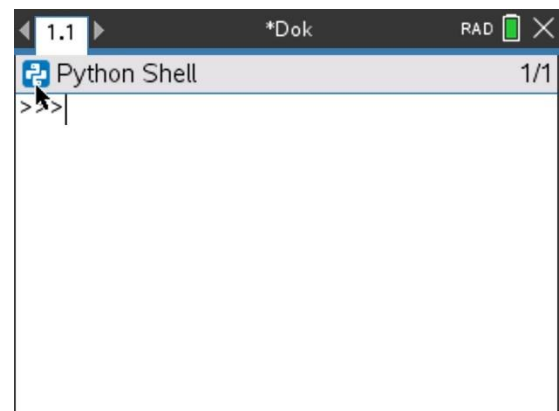


Hinweis für Lehrkräfte : Die Verwendung der **App Python** setzt ein OS der Version 5.2.0 oder höher voraus.

Die Shell ist die eigentliche Oberfläche von Python. Hier können einfache Berechnungen durchgeführt werden, aber man ruft von hier auch Funktionen oder Programme auf. Man kann auch kleine Programme schreiben.

Die Gesamtheit aller Einträge in der Shell bildet die History. Man kann mit den Pfeiltasten in der History ältere Einträge auswählen und sie an die aktuelle Stelle kopieren.

Im « **Menu** » bei « **Tools** » gibt es auch die Möglichkeit, die History der Shell zu löschen (« **Clear History** ») oder die Shell auch neu zu starten («**Reinitialize Shell** ») .





10 Minutes Coding

TI-NSPIRE CX II-T

Ein paar grundlegende Bemerkungen zu Berechnungen

- Die normale Division / liefert eine Dezimalzahl.
- Mit // erhält man den ganzzahligen Teil des Bruches.
- Mit % erhält man den Rest des Bruches.
- Die Potenz x^n wird geschrieben als $x^{**}n$. Aber man erhält sie auch über die Taste \square und ebenso gilt das auch für x^2 .
- Für den logischen Vergleich « **ist gleich** » schreibt man « == ».
- Für « **ungleich** » schreibt man « != ».
- Variablennamen fangen grundsätzlich mit einem kleinen oder großen Buchstaben an, dem eine Zeichenkette aus großen und kleinen Buchstaben und Ziffern folgen kann.
- Die Zahl π wird geschrieben als **pi**. Ihre Verwendung setzt aber den Import des Mathematik-Moduls voraus.

Für die Verwendung bestimmter Funktionen müssen die entsprechenden Module geladen werden. Man findet sie unter « **menu** »..
Als Beispiel ist das Mathematik-Modul dargestellt.

Zeichenketten (strings)

- Zeichenketten werden mit Hochkommas oder Anführungszeichen eingefasst.
- Die Länge einer Zeichenkette erhält man über **len(a)** aus dem Menü « **Built-ins ... List** »
- Der Befehl **a[k]** liefert das **k+1** ste Element der Zeichenkette, da die Zählung mit 0 beginnt.
- Zeichenketten können durch einfache Addition aneinander gehängt werden.

LEKTION 1: ÜBUNG 1

LEHRERMATERIAL

```

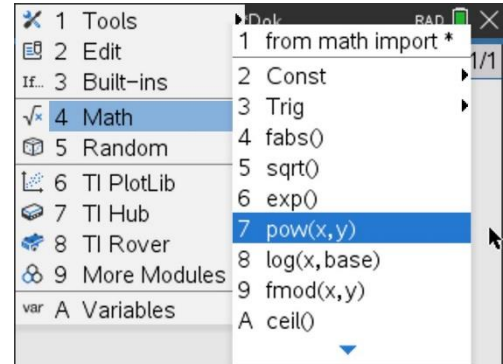
Python Shell 9/9
>>>34/13
2.615384615384615
>>>34//13
2
>>>34%13
8
>>>2**4
16
>>>3**2

```

```

Python Shell 11/11
>>>a=5
>>>a
5
>>>a==5
True
>>>a!=5
False
>>>aBc=7
>>>aBc
7
>>>

```



```

Python Shell 10/10
>>>a="TI-Python"
>>>a
'TI-Python'
>>>len(a)
9
>>>a[4]
'y'
>>>a+" Test"
'TI-Python Test'
>>>

```

Ein kleiner Tipp

Wertzuweisungen an mehrere Variablen können in einer Zeile erfolgen, wenn man die Variablen und ihre Werte durch Kommas trennt.

Verwendung von Programmieranweisungen in der Shell

Es ist in Python möglich, kleine Programme in der Shell zu schreiben und ausführen zu lassen.

Auf dem Bildschirm rechts ist als Beispiel eine **for**-Schleife dargestellt, die Sie im Menü unter „**Built-ins**“ und dann „**Ctl**“ finden.

for index in range (size)
.. block

Wir werden in einer späteren Lektion auf die Schleifen zurückkommen.

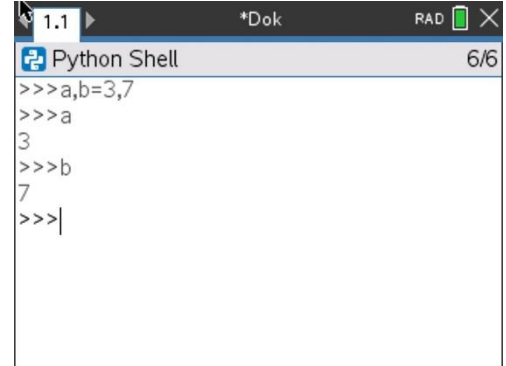
Eine kleine Anwendung

Die kinetische Energie eines Objektes berechnet sich zu $E_C = \frac{1}{2}mv^2$

m ist die Masse in kg

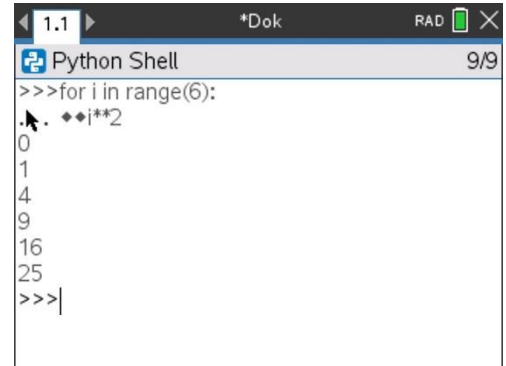
v ist seine Geschwindigkeit in m/s.

Wie groß ist die kinetische Energie e bei einer Masse von 50 kg und der Geschwindigkeit von 12 m/s ?



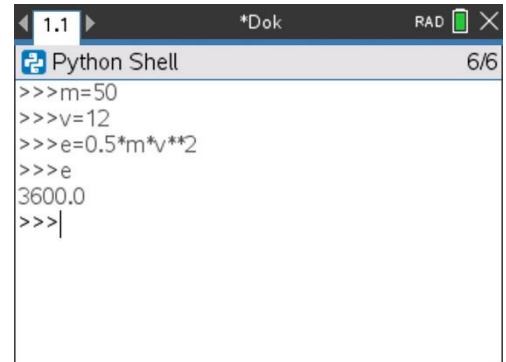
```

1.1 *Dok RAD 6/6
Python Shell
>>>a,b=3,7
>>>a
3
>>>b
7
>>>|
    
```



```

1.1 *Dok RAD 9/9
Python Shell
>>>for i in range(6):
.. .♦♦i**2
0
1
4
9
16
25
>>>|
    
```



```

1.1 *Dok RAD 6/6
Python Shell
>>>m=50
>>>v=12
>>>e=0.5*m*v**2
>>>e
3600.0
>>>|
    
```