

Lektion 1: Erste Schritte in Python

Übung 2 : Datentypen

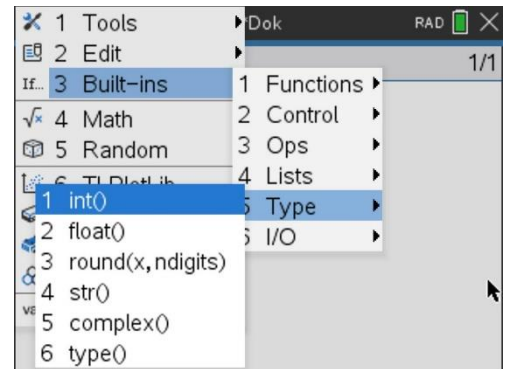
Diese zweite Übung vermittelt die Verwendung der unterschiedlichen Datentypen in Python.

Lernziele :

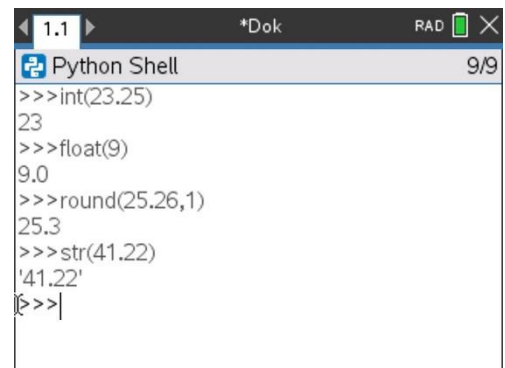
- Kennenlernen der unterschiedlichen Datentypen in Python
- Formatieren der Daten

Daten und Variablen

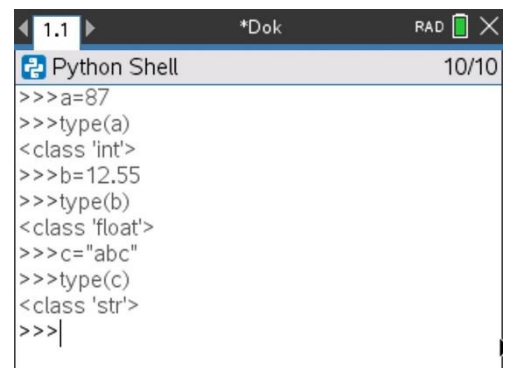
Über « **Menu ... Built-ins ... Type** » erhält man die Anweisungen für die verschiedenen Datentypen.



- **Int()** schneidet den ganzzahligen Teil einer Zahl ab. Auch Terme sind innerhalb der Klammern erlaubt.
- **Float()** wandelt eine Zahl in eine Fließkommazahl um.
- **Round()** rundet auf den ganzzahligen Teil.
- **Round(a,n)** rundet auf die n. Stelle hinter dem Punkt.
- **Str()** wandelt eine Zahl in einen String um.
- **Int()**, **float()** und **round()** machen aus einer als String vorliegenden Zahl des passenden Typs eine « richtige » Zahl.



Mit der Anweisung « **type()** » lässt sich der Typ einer Variablen bestimmen.





Formatierungen

Es gibt noch ein paar präzise Formatierungsanweisungen für die Darstellung von Strings, die der Anweisung **format()** aus dem Menü **I/O** vorangestellt werden müssen :

Beispielzahl : $x = 1234.567890$

- a. `{ :g}'.format(x)` Umwandlung in das Grundformat mit 2 Stellen hinter dem Komma
- b. `{ :.3f}'.format(x)` Darstellung mit 3 Nachkommastellen
- c. `{ :.3e}'.format(x)` wissenschaftliche Darstellung
- d. `{0 :20.3f}'.format(x)` oder
- e. `{0 :>20.3f}'.format(x)` rechtsbündig bei einer Stringlänge 20
- f. `{0 :<20.3f}'.format(x)` linksbündig
- g. `{0 :^20.3f}'.format(x)` zentriert

```
1.1 *Dok RAD 10/10
Python Shell
>>>a=1234.567890
>>>'{:g}'.format(a)
'1234.57'
>>>'{:.3f}'.format(a)
'1234.568'
>>>'{:.2e}'.format(a)
'1.23e+03'
>>>'{0:>20.2f}'.format(a)
'          1234.57'
>>>|
```