



#### Lektion 1: Erste Schritte in Python

#### Übung 2 : Datentypen in Python

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

#### Lernziele :

- Kennenlernen der unterschiedlichen Datentypen in Python
- Formatieren der Daten

### Daten und Variablen

Man kann einer Variablen explizit einen Datentyp zuweisen. Dazu benötigt man zuerst das Modul **math** (*f1*... dann **Modul** , dann **1: math**... und schließlich **1: from math import \*** und **enter**).

Auf demselben Wege findet man das Datenmenü **Type**.

```

PYTHON SHELL
Func Ctl Ops List Type I/O Modul
1:int()
2:float()
3:round()
4:str()
5:complex(real, imag)
6:type()
Esc

```

- **Int()** schneidet den ganzzahligen Teil einer Zahl ab.
- **Round()** rundet auf den ganzzahligen Teil.
- **Int()** erlaubt auch kleine Berechnungen innerhalb der Klammern ; vom Ergebnis wird wieder nur der ganzzahlige Teil angezeigt.
- **Float()** wandelt eine Zahl in eine Fließkommazahl um.
- **Str()** wandelt eine Zahl in einen String um.
- Int(), float() und round() machen aus einer als String vorliegenden Zahl wieder eine « richtige » Zahl.

```

PYTHON SHELL
>>> int(17.8)
17
>>> round(17.8)
18
>>> int(4-6.3)
-2
>>> float(8)
8.0
>>> str(91)
'91'
>>> |
Fns... a A # Tools Editor Files

```

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

```

PYTHON SHELL
>>> from math import *
>>> a=17
>>> type(a)
<class 'int'>
>>> b=-3.12
>>> type(b)
<class 'float'>
>>> c="abc"
>>> type(c)
<class 'str'>
>>> |
Fns... a A # Tools Editor Files

```



#### Formatierungen

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

Beispielzahl :  $x = 1234.567890$

- a. `{:g}.format(x)` Umwandlung in das häufige Format mit 2 Nachkommastellen
- 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

```
PYTHON SHELL
3
>>> x=1234.567890
>>> '{:g}'.format(x)
'1234.57'
>>> '{:.3f}'.format(x)
'1234.567'
>>> '{:.2e}'.format(x)
'1.23e+03'
>>> '{0:>20.2f}'.format(x)
'
      1234.56'
>>> |
Fns... | a A # | Tools | Editor | Files
```