



# Informatik für Schüler, Foliensatz 2

## Ausdrücke, Variablen, Datentypen, Ein- und Ausgabe

Prof. G. Kemnitz

Institut für Informatik, Technische Universität Clausthal  
26. August 2009

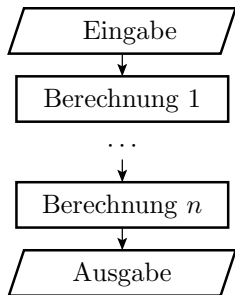
## Einfaches Programm

Ein einfaches Programm:

- liest Eingabedaten ein
- führt schrittweise Berechnungen durch und
- gibt die Ergebnisse aus.

In Python lässt sich jeder Berechnungsschritt einzeln ausprobieren. Python im interaktiven Modus starten:

- Terminal öffnen
- »python« eingeben.
- Anweisungen eingeben.





## Experiment: interaktive Berechnung

- Starten Sie Python im interaktiven Modus

```
...$ python
```

- Geben Sie Folgendes ein. Was berechnet der Computer?

```
(3 * 8) + 1
```

```
(25 / 5) - 7
```

```
2 ** 4
```

```
'Hallo' + ' ' + 'Welt'
```

```
3 > 5
```

```
8 >= 8
```

- 3, 8, ... Zahlenkonstanten
- 'Hallo', ... Zeichenkettenkonstanten



## Variablen für Zwischenergebnisse

- eine Variable
  - ist ein Speicherplatz für ein Ergebnis oder ein Zwischenergebnis
  - hat einen Namen, einen Wert und einen Typ
  - in Python wird der Typ zusammen mit dem Wert zugewiesen (in vielen anderen Programmiersprachen muss der Variablentyp vor der Übersetzung festgelegt werden)
- Vereinbarung und Wertzuweisung an eine Variable:  
`a = 3 + 5;`
- Abfrage des Wertes (Ausdruck ohne Zuweisungsziel eingeben):  
`a`

## Datentypen

- Variablen und Ausdrücke haben einen Typ, abzufragen mit:  
`type(a)`

Typ	Beschreibung
str	Literal, Zeichenkette: Folge von Buchstaben, Zifferen und Sonderzeichen (string)
int	ganze Zahl: natürliche Zahlen und negative ganze Zahlen (integer)
float	Gleitkommazahl: rationale und reelle Zahlen
bool	Wahrheitswert: wahr (True) oder falsch (False)
type	Die Typenangabe ist auch ein Typ
...	



## Experiment: Variablen und Datentypen

Welcher Wert und welcher Typ wird den folgenden Variablen zugewiesen?

```
a = (3 * 8) + 1
```

```
b = 'hallo ' * 3
```

```
c = 0.1 * 10
```

```
d = c + a
```

```
e = c > a
```

Kontrollieren Sie ihre Ergebnisse am Rechner.



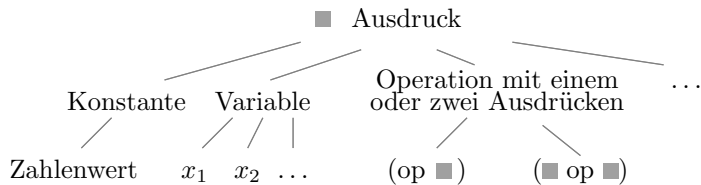
## Zuweisungen

- Allgemeiner Aufbau einer Zuweisung:

*Variable = Ausdruck*

(*kursiv*: Platzhalter für ...)

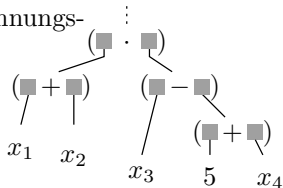
- ein Ausdruck kann sein:
  - eine Konstante, z.B. 0.1, 8, 'hallo' ...
  - eine Variable
  - eine Operation mit einem oder zwei Ausdrücken z.B. (3 \* 8), (c > a), (-0.1), ...



Beispielausdruck

$$(x_1 + x_2) \cdot (x_3 - (5 + x_4))$$

Berechnungs-  
baum



- als Folge von Anweisungen mit einer einzelnen Operationen:

```

tmp1    = x4 + 5
tmp2    = x1 + x2
tmp3    = x3 + tmp1
Ergebnis = tmp2 * tmp3
  
```

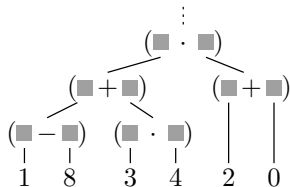




- Ein Ausdruck mit mehreren Operatoren ist bereits ein kleines Programm.
- Bei der Abarbeitung von Ausdrücken in Python gelten die üblichen Regeln für die Ausführungsreihenfolge aus der Mathematik:
  - geklammerte Teilausdrücke zuerst berechnen
  - Punktrechnung vor Strichrechnung etc.
- In Zweifelsfällen:
  - ausprobieren
  - durch eine Anweisungsfolge nachbilden oder
  - Klammern setzen.

## Experiment: Berechnungsbaum als Programm

Gegeben ist folgender Berechnungsbaum:



- Ermitteln Sie den zugehörigen Ausdruck und geben Sie ihn im interaktiven Modus in Python ein
- Berechnen Sie den Ausdruck als Anweisungsfolge mit je einer Operation je Anweisung.
- Ergebnisse gleich?

## Programmausgabe

Die Syntax der Ausgabeanweisung:

```
print [Ausdruck {, Ausdruck}]
```

Zeilenvorschub. Ausgabe der Werte aller aufgelisteten Ausdrücke getrennt durch Leerzeichen. Beispiele:

```
print '5+3 =', 5+3
```

```
⇒ 5+3 = 8
```

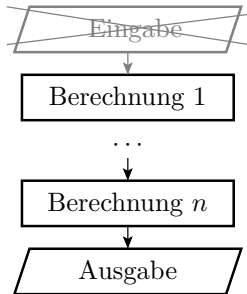
```
a=5
```

```
b=7
```

```
c=a*b
```

```
print 'a =', a, ' b =', b, ' c = ', c
```

```
⇒ a = 5 b = 7 c = 35
```





## Experiment

- Editor starten
- erstellen einer Datei test.py mit der Anweisungsfolge von der Folie zuvor
- ausführen mit

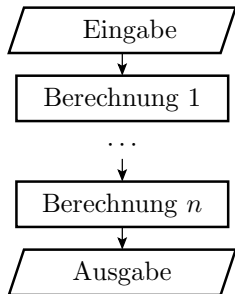
```
...$ python test.py
```

## Programmeingabe

Die Syntax einer Programmeingabe:

```
Variable = input('Prompt')
```

- »input« ist eine Funktion; eine Funktion liefert wie ein Ausdruck einen Wert, der einer Variablen zugewiesen werden kann
- Prompt: Text, der zur Eingabeaufforderung ausgegeben wird
- Der eingegebene Text muss eine Python-Konstante sein; abschließend Eingabetaste betätigen.



## Experiment

- Ein Testprogramm für die Eingabefunktion:

```
x = input('Bitte Wert eingeben:')  
print 'Wert:', x, 'Typ:', type(x)
```

- in eine Datei Test2.py schreiben
  - mit Python ausführen
- 

- Beispiele für Pythonkonstanten:

`int` ganze Zahlen, z.B. 136, -96, 0  
`float` Gleitkommazahlen, z.B. 4.0, 45e-2 ( $45 \cdot 10^{-2}$ ),  
2E2 ( $2 \cdot 10^2$ )  
`str` Zeichenkette, z.B. 'Das ist ein Text'



## Aufgabe 2.1: Programm mit Ausgabe

Legen Sie ein neues Unterverzeichnis »~/Uebung2« an  
Schreiben Sie ein Programm »a2.1.py«, dass

- zwei Variablen a und b zu Beginn konstante Zahlenwert zuweist
- von den beiden Variablen die Summe, das Produkt und die Differenz berechnet
- folgende Ausgabe erzeugt:  
...\$ Summe:       *Wert\_der\_Summe*  
...\$ Produkt:     *Wert\_des\_Produkts*  
...\$ Differenz:   *Wert\_der\_Differenz*
- Testen des Programms

### Hilfestellungen

- kursiv – Platzhalter, für den etwas einzusetzen ist
- Arbeit mit Dateien und Verzeichnissen siehe nächste Folie



## Wiederholung zur Arbeit mit Dateien und Verzeichnissen unter Linux

- Hilfe:

```
man [Befehl]
```

- Python-Interpreter beenden: »Str-D«

- Verzeichnisinhalt anzeigen (list):

```
ls [-l] [-a] [Pfad] [Datei]
```

- neues Verzeichnis erzeugen (make directory):

```
mkdir Verzeichnisname
```

- Verzeichnis ändern (change directory):

```
cd Pfad
```

- Start des Python-Programms:

```
python a2.1.py
```





## Aufgabe 2.2: Ergänzung der Eingabe

Erweitern Sie das Programm aus der vorherigen Aufgabe und eine Eingabeanweisung für die beiden Variablen a und b.