

Einige Sprachelemente von Python

Entweder arbeiten wir im Terminal unter MacOS X oder in der Python-Shell der Entwicklungsumgebung Idle (www.python.org/)

1. Zahlen und Variable (müssen nicht deklariert werden, d.h. Speicherplatz wird dynamisch erzeugt)

Operationen	Beispiel	Bemerkung
Addition, Subtraktion, Multiplikation, Potenz	<code>2+3, 5-4, 6*7, 8**9</code>	
Division, Ganzzahl-Division, Modulo	<code>3.14/2, 3/2, 54%7</code>	
Mathematische Funktionen (Standard)	<code>round(5.67), int(5.67)</code>	runden und abrunden
Mathematische Funktionen (Import)	<code>math.sin(), math.sqrt(2), random.seed(), random.randint(1,6)</code>	<code>import math, import random</code>
Vergleiche	<code>3<4, 5<=5, 6==6, 7!=8</code>	
Logische Operatoren	<code>not, and, or</code>	
Zuweisungen	<code>a = 5 b = 'z' c = "Hallo Python" d = True</code>	Zahlen direkt zuweisen Zeichen in Hochkommas, Zeichenketten in Anführungszeichen Wahrheitswerte <code>True</code> und <code>False</code>

3. Ein- und Ausgabe

Ganze Zahlen eingeben	<code>input("Ganze Zahl : ")</code>	
Ganze Zahlen eingeben und speichern	<code>a = int(input("Ganze Zahl : "))</code>	
Resultat ausgeben	<code>print("Resultat: ",a)</code>	
Dezimalzahl eingeben und speichern	<code>b = float(input("Dezimalzahl : "))</code>	
Zeichenkette eingeben	<code>c = raw_input("Text : ")</code>	

4. Verzweigungen

Einfache Verzweigung <code>if ...</code>	<code>if n > 0: print("positiv")</code>	Einrücken mit TAB-Taste obligatorisch!
Verzweigung mit <code>if ... else</code>	<code>if n > 0: print("positiv") else: print("nicht positiv")</code>	

5. Schleifen

for-Schleife <code>for zaehler in a, b, c:</code>	<code>for i in 2, 3, 5, 7: print("Quadrat von ",i,"ist ",i*i)</code>	mit Auswahlliste
for-Schleife <code>for zaehler in range (a,b,s):</code>	<code>for i in range(1,10,1): print("Quadrat von ",i,"ist ",i*i)</code>	mit Bereich
while-Schleife while Bedingung:	<code>i = 1 while i<=10: print(i*i) i += 1</code>	

6. Funktionen definieren

<code>def funktion():</code>	<code>def wuerfel(): from random import randint return randint(1,6)</code>	Funktion <i>ohne</i> Parameter
<code>def funktion(x):</code>	<code>def kub(x): return x**3</code>	Funktion <i>mit</i> Parameter

7. Programme

Kommentar	<code># demo.py</code>	Kommentare sind sinnvoll, auch um das Programm zu strukturieren
Beispiel: Collatz-Problem:	<code># collatz.py n = int(input("Natuerliche Zahl: ")) while n != 1: if n % 2 == 0: n = n / 2 else: n = 3*n+1 print(n)</code>	Beginne mit irgend einer natürlichen Zahl n . Falls n nicht gleich 1 ist, mache ist n gerade: rechne $n : 2$ sonst: rechne $3 \cdot n + 1$ schreibe n

