



Die Programmiersprache Python - eine kurze Einführung

Der Python-Prompt lautet >>> und bedeutet, dass Python bereit ist, eine Anweisung auszuführen oder eine Frage zu beantworten.

Python arbeitet interaktiv – wir können eine Anweisung schreiben und sofort die Auswertung erhalten. Python antwortet also mit der ausgewerteten Eingabe.

Grundrechenarten:

Anweisung	Ausgabe/Erklärung
>>> 5 + 2	7 (Addition)
>>> 5 - 2	3 (Subtraktion)
>>> 5 * 2	10 (Multiplikation)
>>> 5 / 2	2.5 (Division)
>>> 5 // 2	2 (Ganzzahldivision)
>>> 6.0 // 2	3.0 (Ergebnis ist float)
>>> 5 ** 2	25 (Potenzieren)

Arbeiten mit Strings (Zeichenketten)

Anweisung	Ausgabe/Erklärung
>>> einstring = "Zeichenkette"	immutable, d. h. einzelne Zeichen können nicht verändert werden
>>> einstring[0]	'Z' erstes Zeichen des Strings
>>> einstring[0] = 'W'	ERROR Fehlermeldung, da immutable
>>> 3 * einstring	'ZeichenketteZeichenketteZeichenkette'
>>> einstring[-1]	,e' erstes Zeichen von hinten
>>> einstring[1:6]	,eiche' 1. bis 6. Zeichen (jedoch ohne 6.)
>>> einstring[:-1]	,Zeichenkett' letzter Buchstabe wird gelöscht

Ausgabe (print-Befehl)

Anweisung	Ausgabe/Erklärung
>>> augen = 2 >>> nase = 1 >>> print("ich habe", augen, "Augen und", nase, "Nase!")	Trennung durch „Komma“ (gemischte Datentypen, keine Typumwandlung) <i>ich habe 2 Augen und 1 Nase!</i>
>>> print("ich habe " + str(augen) + " Augen und " + str(nase) + " Nase!")	Trennung durch „Plus“ (Typumwandlung erforderlich) <i>ich habe 2 Augen und 1 Nase!</i>



Quelltext

Kommentare beginnen mit #-Zeichen oder sind zwischen je 3 Anführungszeichen
Zusammengehörende Code-Blöcke werden eingerückt In der Regel 4 Leerzeichen Beginn immer in gleichen Spalte keine geschweiften Klammern, Strichpunkte etc.
Am Ende einer Anweisung steht ein Doppelpunkt , danach Einrückung Anweisung können durch Backslash (\) in der nächsten Zeile fortgeführt werden

Wiederholungen

Vorbemerkung:	
Zahlenbereiche	
>>> range(8)	die ersten 8 Zahlen ab 0 Ergebnis: 0,1,2,3,4,5,6,7
>>> range(3, 6)	alle ganzen Zahlen von 3 bis 6 (ohne die 6) Ergebnis: 0, 1, 2, 3, 4, 5, 6, 7
Feste Anzahl an Wiederholungen (for-Schleife) (die Laufvariable i nimmt nacheinander jeden Wert an)	
for i in range(0, 3): print(i)	Ausgabe: 0 1 2
for i in ("Auto", 2, 3.14): print(i, type(i))	Ausgabe: ('Auto', <type 'str'>) (2, <type 'int'>) (3.14, <type > 'float')
Bedingte Wiederholungen (while-Schleife)	
i = 0 while i < 3: print(i) i = i + 1	Ausgabe: 0 1 2
Weitere Anweisungen in Wiederholungen	
break	verlässt diese Schleife sofort und macht nach der Schleife weiter
continue	beendet diesen Schleifendurchlauf und macht sofort mit dem nächsten Durchlauf weiter

Bedingungen

if ... : (einseitige Bedingung) elif ... : (für Mehrfachauswahl) else: (Alternative)	Beispiel: <pre>from random import choice optionen = ['Sonne', 'Wolken', 'Regen', 'Sturm'] wetter = choice(optionen) if wetter == 'Sonne': badengehen() elif wetter == 'Wolken': radeln() else: regenschirmEinpacken()</pre>
---	---



Funktionen

Methoden beginnen mit dem Schlüsselwort <code>def</code>	Beispiel: <code>from math import sqrt def gibWurzel(zahl=0): „optionaler Kommentar, der angezeigt wird“ if zahl < 0: return False else: return sqrt(zahl)</code>
Optional: Kommentar, der bei Aufruf angezeigt wird, folgt in „“	
Rückgabewert folgt nach Schlüsselwort <code>return</code>	

Klassen

Klassen beginnen mit dem Schlüsselwort <code>class</code>	Beispiel: <code>class Quader(): def __init__(self): self.laenge = 1 self.breite = 1 self.hoehe = 1 def volumen(self): vol = self.laenge * self.breite * self.hoehe return vol</code>
Als Konstruktur dient die Methode <code>__init__(self)</code> <code>self</code> bezeichnet dabei den Selbstbezug, also die Klasse selbst.	
Attribute der Klasse werden mit Hilfe von <code>self</code> gesetzt.	
Weitere Methoden können mit Hilfe des Schlüsselwortes <code>def</code> hinzugefügt werden.	

Vererbung

Oberklassen werden bei der Klassendeklaration in die Klammer nach dem Klassennamen gesetzt.	Beispiel: (Klasse Quader erbt von Klasse box) <code>class Quader(box): def __init__(self): box.__init__(self) self.laenge = 1 self.breite = 1 self.hoehe = 1</code>
Mehrfachvererbung ist möglich, die einzelnen Oberklassen werden durch ein Komma voneinander getrennt.	
Im Konstruktor einer Klasse müssen auch die Konstruktoren der Oberklassen aufgerufen werden.	