Python3 Tutorial

Aktuelle Version: 3.5.2

/dev/tal/usr/Culain

03-11-2016

Inhalt

- Basics
- Operanten
- Datentypen
- Ablaufkontrolle
- Funktionen

Basics

- Module importieren: „import sys“
- Output auf die Konsole: „print(„Hello World!“)“
- Variablen deklarieren: „i = 42“ oder „name = „Python““
- Singleline Kommentare: „# scheint zu funktionieren“
- Multiline Kommentare: „““ Powerpoint mag dreifach Anführungsstriche wohl nicht „““

print()

- print(„Hello World“)
- print(„i don‘t like automatic newlines“, end=„“)
- print(„%s %s“ %(„Hello“, „World!“))
- print(„{0} {1}!“.format(„Hello“, „World“)
- print(„Hello “ + „World!“) # ohne Leerzeichen
- print(„Hello“, „World!“) # zusätzl. Leerzeichen
- print(„\n“ * 5)

Operanten

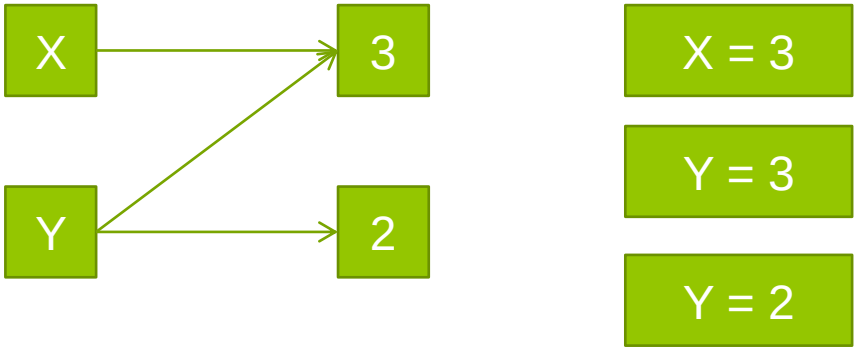
- + Add
- - Substract
- * Multiplicate
- / Divide
- % Modul
- // Floordivision
- ** Exponential
- Order of Operation beachten!

Datentypen

- Numbers 42; 3.1415
- Strings „Hello World!“
- Lists [1, 2, 3]
- Tuples (1, 2, 3)
- Sets {1, 2, 3}
- Dictionaries {1:Eins, 2:Zwei, 3:Drei}

Pointer

- …Pointer?
- Gibbet nich!
- Werden aber Verwendet
- Beispiel:



Merke: Pointer werden automatisch verwendet

Numbers

- …Zahlen halt
- Ganzzahlen
- Kommazahlen
- Kann beliebig große Zahlen speichern
- Wechselt bei Bedarf von INT zu LONG

Strings

- Quasi Liste von Buchstaben
- Mit Funktionen
- name = „Monty“
- print(name[2])
  - >>> n
- print(name[1:3])
  - >>> on
- print(name[::2])
  - >>> Mny

Strings²

- print(name[::-1])
  - >>> ytnoM
- print(name[-1])
  - >>> y
- print(name * 3)
  - >>> MontyMontyMonty
- Escapezeichen z.B. \n, \t, \\, \‘, \“ …
- …können aber auch escaped werden
  - r“\n bewirkt einen Zeilenumbruch“

Listen

- Liste verschiedener Variablen
- Inhalte können unterschiedliche Typen haben
  - Liste = [„Monty“, 42, Liste2]
- Listen können manipuliert werden
- Liste.append(1337)
- Liste.insert(2, „Leet“)
- Liste.remove (1337) | .pop(1337)
- Liste.index(„Monty“)
- Liste.reverse()
- Liste.sort()

Tuples

- Ähnlich wie Listen
- Nicht veränderbar !
- Tuple = „Monty“, 42 || Tuple = („Monty“, 42)
- Tuple[0]
  - >>> „Monty“

Sets

- Das gleiche wie Liste
- Inhalte dürfen nicht doppelt vorkommen

Dictionaries / Maps

- Ähnlich wie Listen
- Inhalte bestehen aus „Key : Value“ Paaren
- Index wird ersetzt durch Keys
- Keys müssen unveränderlich Typ haben

Ablaufkontrolle

- if, elif, else
- for
- while

if, elif, else

- Verzweigung im Ablauf
- Überprüft Wert auf „Wahrheit“
- Wert ist Wahr wenn er !=0 ist
- x = 42
- if (x > 41):
  - print(„x ist größer als 41“)
- elif (x < 41):
  - print(„x ist kleiner als 41“)
- else: # Wenn nichts anderes true ist
  - print(„x ist weder kleiner noch größer als 41“)

for - loop

- Simple schleife
- Iteriert durch irgendwas durch
- for (i in range(3)): # oder [0, 1, 2]
  - print(i)
- Schleife kann gesteuert werden durch:
  - break # beendet die Schleife
  - continue # überspringt rest der iteration

while - loop

- Unbestimmte Schleife
- Läuft so lange wie die Abfrage „true“ ist
- x = 1
- while(x < 10):
  - print(„{} ist eine schöne Zahl“.format(x))
  - x += 1 # x = x + 1

Funktionen

- Funktionieren halt
- Für mehrfachen aufruf von Kot
- Kann einen Wert zurückgeben
- ...musse aber nicht
- def HelloWorld(name):
  - print(„Hello {}“.format(name))
  - return 0