

Grundlegende Sprachelemente

- Ein Python-Programm ist eine Reihe von Python-Anweisungen, die nacheinander abgearbeitet werden.
- Die Reihenfolge der Abarbeitung wird gesteuert durch Kontrollanweisungen (Selektion und Iteration)
- Teile von Anweisungen können als Prozeduren oder Funktionen definiert werden -und damit „ausgelagert“ werden
- Grundlage aller dieser Möglichkeiten sind Python-Schlüsselworte

Syntax und Semantik

Schlüsselworte

- Schlüsselwörter sind die reservierten Wörter in Python.
- ein Schlüsselwort kann nicht als Variablenname, Funktionsname oder einen anderen Bezeichner verwenden; sie werden verwendet, um die Syntax und Struktur der Sprache Python zu definieren.
- in Python wird zwischen Groß- und Kleinschreibung unterschieden.
- es gibt etwas mehr als 30 Schlüsselworte
- Liste der Schlüsselworte

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Bezeichner

- Namen von Variablen unterliegen Konventionen
 - erstes Zeichen muss ein Buchstabe sein (auch der Unterstrich ist erlaubt)
 - anschließend folgen Ziffern und Buchstaben, kein Leerzeichen erlaubt
 - es darf kein Schlüsselwort sein
- Beispiele
 - `zahl1` gültiger Bezeichner
 - `_zahl1` gültiger Bezeichner
 - `2mal17` kein gültiger Bezeichner (erstes Zeichen ist eine Ziffer)
 - `break` kein gültiger Bezeichner (Schlüsselwort in Python)
 - `zahl 1` kein gültiger Bezeichner (Leerzeichen nach `zahl`)

Grundlegende Elemente einer Sprache

- in Python gibt es keine Deklaration der Variablen
- in Python werden Werte im Speicher angelegt und mit einem Namen verbunden
- die Verbindung eines Wertes mit dem Namen geschieht durch eine Zuweisung
- Beispiele

Anweisung	Bedeutung
<code>x = 1</code>	dem Namen x wird der ganzzahlige Wert 1 zugewiesen damit verweist x auf einen Wert mit dem Datentyp Ganzzahl
<code>anfangs_kapital = 1575.75</code>	dem Namen anfangs_kapital wird die Gleitkommazahl 1575.75 zugewiesen als Trennzeichen für die Dezimalstellen wird der Punkt verwendet damit verweist anfangs_kapital auf einen Wert mit dem Datentyp Gleitkommazahl
<code>nach_name = "Maier"</code>	dem Namen nach_name wird die Zeichenkette "Maier" zugewiesen damit verweist nach_name auf einen Wert mit dem Datentyp String

Standarddatentypen (elementare Datentypen)

Ganzzahl (Integer)

- Ganzzahlige Werte werden in Python durch den Datentyp **int** repräsentiert
- sie haben keine Nachkommastellen
- Integer-Datentypen werden genau dargestellt
- es treten keine Rundungsfehler auf
- in Python kann man theoretisch beliebig große ganzzahlige Werte speichern
- neben den Werten im Dezimalsystem sind auch Darstellungen im Oktal-, Hexadezimal und Dualsystem möglich

- Beispiel:

```
# Dezimalsystem, Basis 10)
ganzzahl = 12345

# Dualsystem, Basis 2)
dualzahl = 0b10101

# Oktalsystem, Basis 8)
oktalzahl = 0o11753

# Hexadezimalsystem, Basis 16)
hexazahl = 0xAB11F

print (ganzzahl, dualzahl, oktalzahl, hexazahl)
```

Ausgabe

```
12345 21 5099 700703
```

Gleitkommazahl (Float)

- Gleitkommazahlen oder Fließkommazahlen werden in Python durch den Datentyp **float** repräsentiert
- die Nachkommastellen werden durch einen Punkt vom ganzzahligen Anteil getrennt
- Float-Datentypen können ggf. nicht genau dargestellt werden
- es kann zu Rundungsfehlern kommen
- es ist möglich sehr kleine oder sehr große Werte anzugeben

- Beispiel 1:

```
float_zahl1 = 12345.0
float_zahl2 = 0.00000000000000012345
float_zahl3 = 12345678901234567890.1

print (float_zahl1, float_zahl2, float_zahl3)
```

Ausgabe

```
12345.0 1.2345e-15 1.2345678901234567e+19
```

- Beispiel 2:

```
a = 7.07
b = 5.05
print (a+b)
```

Ausgabe

```
12.1200000000000001
```

Zeichenketten (String)

- Zeichenketten werden in Python durch den Datentyp **str** repräsentiert
- Werte vom Zeichenketten-Typ enthalten beliebig viele Zeichen
- Zeichenketten oder Strings sind eine Folge von Zeichen, die wahlweise in einfachen oder doppelten Anführungszeichen geschrieben werden

- Beispiel:

```
str1 = "Hallo"  
str2 = 'Welt'  
str3 = "2%&/(@'+##`*'"
```

```
print (str1,str2,str3)
```

Ausgabe

```
Hallo Welt 2%&/(@'+##`*'"
```

Einfache Operationen

- mit den arithmetischen Operatoren (+, -, /, *) kann in Python addiert, subtrahiert, dividiert und multipliziert werden
- mit Hilfe des Zuweisungsoperators = werden die Werte bzw. Ergebnisse einer Berechnung einem Namen zugewiesen
- der Operator + verkettet Strings (Zeichenketten)

- Beispiel:

```
# Ganzzahl (Integerwerte)
x = 10
y = 20
summe = x + y
produkt = x * y

# Gleitkommazahl (Floatwerte)
x = 2.4
y = 1.2
quotient = x / y

# Zeichenkette (String)
text1 = "Hallo"
text2 = " wie geht es?"
text = text1 + text2

print (summe, produkt, quotient, text)
```

Ausgabe

```
30 200 2.0 Hallo wie geht es?
```