Python - Einführung

Jan Popko
Python Grundkurs

Business Trends Academy (BTA) GmbH
Nestorstraße 36
D-10709 Berlin
Tel.: +49 (0) 30 894 087 57
Fax: +49 (0) 30 895 429 94
Geschäftsführer:
Gabriele Fleischmann-Hahn
Maxi-Marlen Fleischmann
Hauptsitz des Unternehmens:
Nestorstraße 36, D-10709 Berlin
HRB 115251 B / Amtsgericht Berlin Charlottenburg
Steuer-Nr. 27/248/31179

Die ersten Schritte

Die Entwicklungsumgebung **IDLE** (der Name stammt von einem Mitglied der Monty-Python-Gruppe, Eric Idle) beinhaltet die **Python-Shell**.

Die Python Shell ist eine interaktive Konsole, ähnlich der IRB von Ruby.

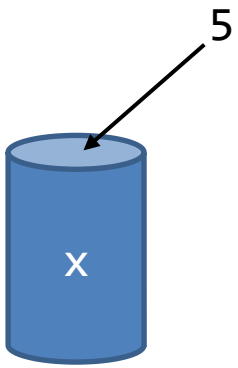
Mit dem Befehl **python** in der cmd wird die Python-Shell gestartet.

>>> (Prompt, eine Aufforderung zur Eingabe, ist die Eingabe erfolgt, wird sie durch Enter beendet und von der Python-Shell direkt ausgeführt)

```
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:57:15) [MSC v.1915 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 2+2
4
>>> _
```

Variablen

Variablen kann man sich als Behälter für Daten vorstellen.



Hier wird die Ganzzahl **5** in den Behälter mit dem Namen **x** gepackt.
Das nennt man Zuweisung. Der Zuweisungsoperator ist das **=** .

Die Anweisung in Python würde also so aussehen:
x = 5

Variablen

Auch Mehrfach-Zuweisungen sind möglich:

```
x, y = 5, 20
```

Der Behälter x enthält nun die Ganzzahl 5.

Der Behälter y enthält nun die Ganzzahl 20.

Ausgabe des Inhalts einer Variable:

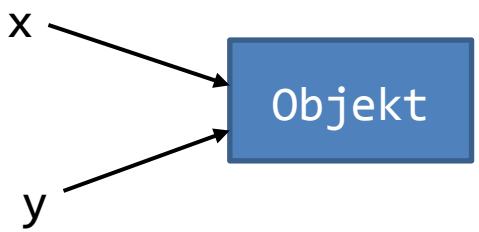
```
print(x)

print(y)
```

Variablen

Diese Vorstellung einer Variable als Behälter gilt jedoch nur für einfache Datentypen, wie Zahlen oder Zeichenketten!

Bei Objekten sieht die Sache etwas anders aus.

Bei Objekten sind die Variablennamen eher Zeiger, als Behälter.



Variablen

Python unterstützt die **dynamische Typisierung** von Variablen.

Der Typ einer Variablen ist nie an seinen Namen gekoppelt, sondern gehört allein zum Objekt.

```
x = 1
```

Das Objekt (1) ist vom Typ Integer, aber dem Name (x) ist das egal.

Dieses Prinzip nennt man "**Duck-Typing**".
Man sieht dem Objekt an, von welchem Typ es ist.

Der Begriff Duck-Typing leitet sich von einem Gedicht ab:

"When I see a bird that walks like a duck and swims like a duck and quacks like a duck, I call that bird a duck"

James Whitcomb Riley

Variablen

Mit der Methode **type()** kann man den Datentyp bestimmen:

```
type("Martin")
class str

type(5)
class int

type(5.5)
class float

# geht auch mit Variablen
x = 1
type(x)
class int
```

Die ersten Schritte

Mit den Pfeiltasten (hoch und runter) auf der Tastatur, kann man die zuletzt eingegebenen Befehle durchgehen und mit Enter erneut ausführen.

Ausgabe auf dem Bildschirm:
print("hello world!")

Eingabe des Users:
variable = input("Text für den User")

```
>>> x = input("bitte eine Zahl eingeben!")
bitte eine Zahl eingeben!7
>>> x
'7'
```

Kommentare
Text der nicht als Programmcode interpretiert wird
 # ich bin ein Kommentar

Business Trends Academy (BTA) GmbH
Nestorstraße 36
D-10709 Berlin
Tel.: +49 (0) 30 894 087 57
Fax: +49 (0) 30 895 429 94
Geschäftsführer:
Gabriele Fleischmann-Hahn
Maxi-Marlen Fleischmann
Hauptsitz des Unternehmens:
Nestorstraße 36, D-10709 Berlin
HRB 115251 B / Amtsgericht Berlin Charlottenburg
Steuer-Nr. 27/248/31179

Datentypen

Integer (Ganzzahlen)
Zahlen ohne Komma, sind vom Typ int

1, 38563, 54 …

Float (Gleitkommazahlen)
Zahlen mit Komma (**in Python PUNKT**), sind vom Typ float

3.5 78.6. 3.14 …

Mit Zahlen können die normalen mathematischen Operationen (plus, minus, mal, geteilt, modulo etc.) ausgeführt werden.

1 + 2 = 3

2 + 2.2 = 4.2

Datentypen

Der Modulo-Operator %
Gibt den ganzzahligen Rest einer Division zurück. Wird häufig benutzt, wenn man ermitteln will, ob ein Zahl gerade oder ungerade ist.

```
11 % 2
1      # denn die 2 steckt nur fünf mal ganz in der
       # 11 und der Rest ist 1
```

Die ganzzahlige Division //
Liefert den nach unten gerundeten ganzzahligen Quotienten

```
3 // 2
1

10 // 3
3
```

Datentypen

Strings (Zeichenketten)
eine Aneinanderreihung von Zeichen, **werden immer in "" oder '' gepackt!**
"Weihnachten", "sdfdf32340813$§%+*/", 'python' …

Bei Strings hat das **+** eine andere Bedeutung als bei Zahlen.

```
1 + 2 = 3     # Addition

"Weih" + "nachten"   # Konkatenation (Strings werden
        aneinandergehängt)
```

Geht nicht:
```
1 + "zeichenkette"
```
FEHLER!

Business Trends Academy (BTA) GmbH
Nestorstraße 36
D-10709 Berlin
Tel.: +49 (0) 30 894 087 57
Fax: +49 (0) 30 895 429 94
Geschäftsführer:
Gabriele Fleischmann-Hahn
Maxi-Marlen Fleischmann
Hauptsitz des Unternehmens:
Nestorstraße 36, D-10709 Berlin
HRB 115251 B / Amtsgericht Berlin Charlottenburg
Steuer-Nr. 27/248/31179

Datentypen

Strings (Zeichenketten)
lange Zeichenketten (über mehrere Zeilen, werden in drei Anführungszeichen, oder drei Hochkommata eingeschlossen:

```
"""Das ist die 1. Zeile
und das die 2. Zeile"""
Das ist die 1. Zeile\nund das die 2. Zeile
```

Python fügt den Zeilenumbruch **(\n)** hinzu.

Datentypen

Boolean (Wahrheitswerte)
sind vom Datentyp bool
True und False

12 > 11
True

"heute" == "morgen"
False

Booleans werden häufig durch Vergleiche ermittelt:

<	**Kleiner als**
<=	**Kleiner gleich**
>	**Größer als**
>=	**Größer gleich**
==	**Gleich**
!=	**Ungleich**

Datentypen

NoneType

None

absolut rein gar nichts, auf keinen Fall ist das ein leerer String, die Zahl Null, die leere Menge …

vergleichbar mit Null in PHP und Nil in Ruby

```
type(None)
class NoneType

x = None
type(x)
class NoneType
```

Datentypen

Typumwandlungen

```
int()  #erzeugt eine ganze Zahl, sofern das möglich ist
int("2")  # aus dem String "2" wird die int-Zahl 2


float()       # konvertiert eine Zahl oder einen String in eine
          Gleitkommazahl
float("2")    # aus dem String "2" wird die float-Zahl 2.0


str()     # konvertiert eine Zahl in einen String
str(2)     # aus der int Zahl 2 wird der String "2"
```