Python - Editor

Jan Popko
Python Grundkurs

Business Trends Academy (BTA) GmbH
bta
BUSINESS TRENDS ACADEMY
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

Editor

Die Python-Shell ist zum Entwickeln nicht wirklich geeignet, sie ist eher zum Testen und zum Ausprobieren da.

Ein richtiges Python-Programm (Script) ist eine Datei mit der Endung .py

Wir legen einen neuen Ordner **C:/bta_py** an. Hier kommen unsere Scripte rein.

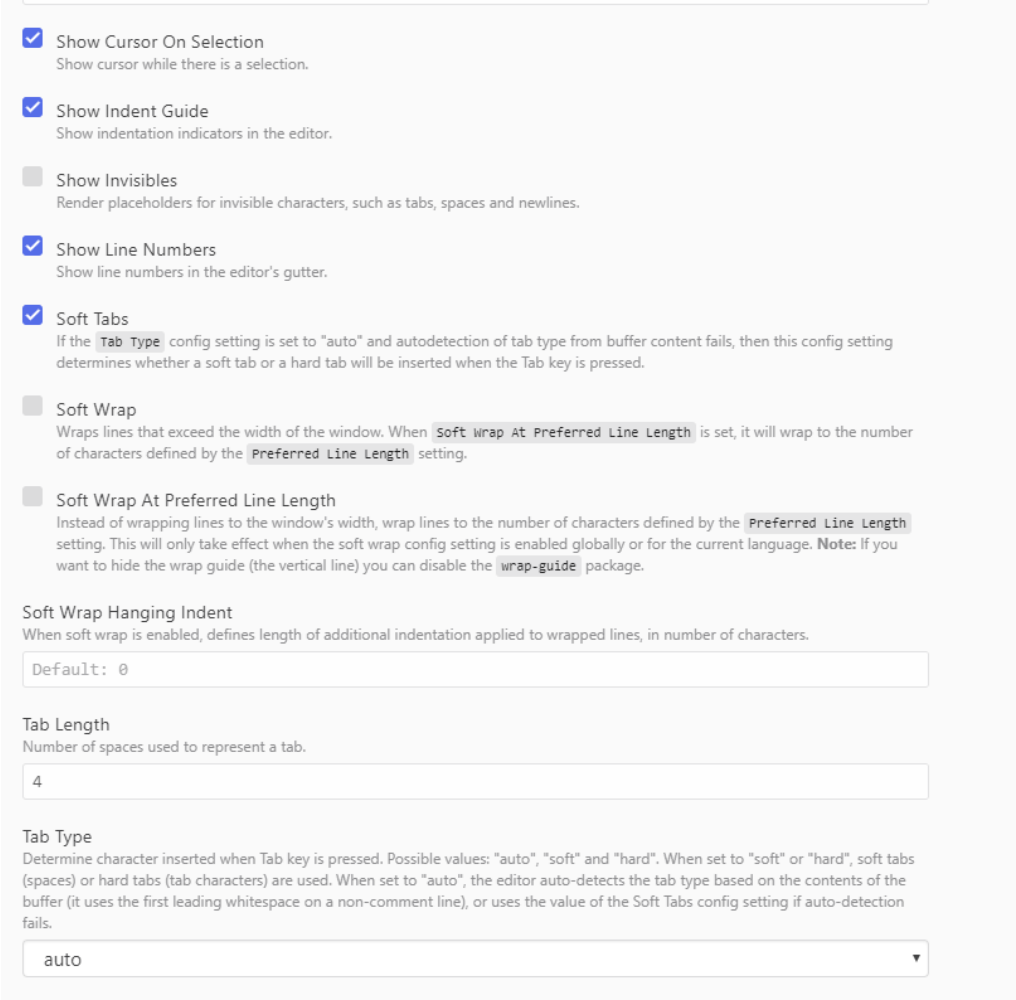
Wir nutzen den Atom-Editor zum Schreiben unseres Python-Codes.

unter **atom.io** kann der Editor runtergeladen werden.

Jetzt stellen wir Atom so ein, dass er gut mit Python funktioniert.

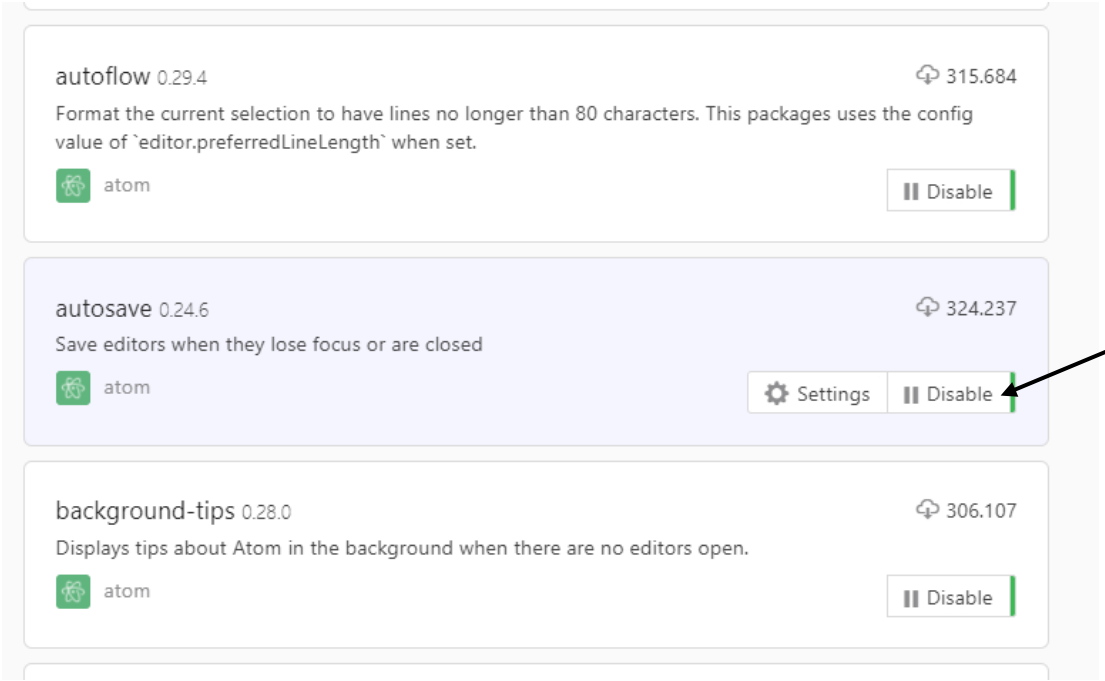
Editor

unter **File/Settings/Editor** machen wir folgende Einstellungen:



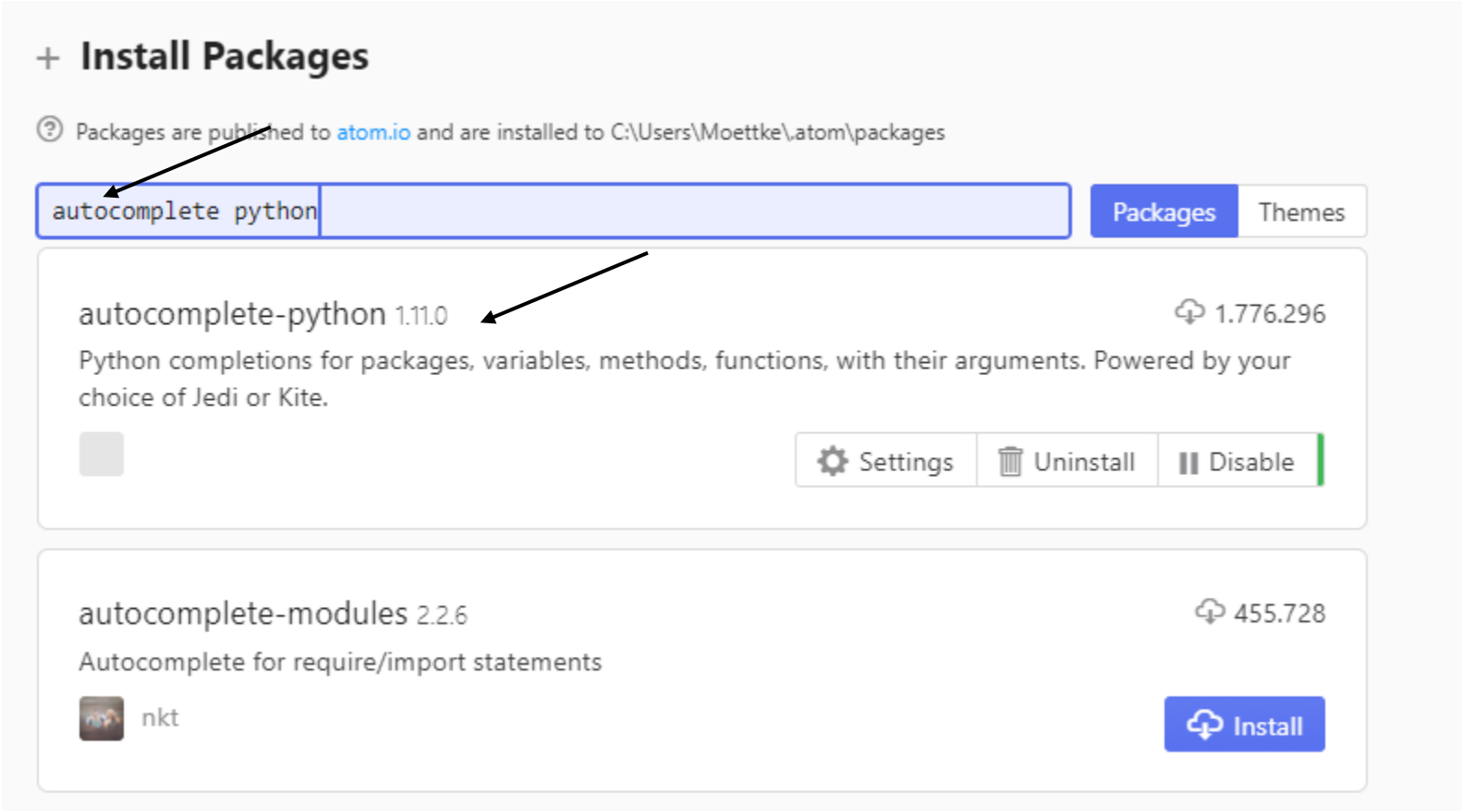
Editor

unter **File/Settings/Packages** machen wir folgende Einstellungen:



Editor

unter **File/Settings/+Install** machen wir folgende Einstellungen:



Editor und Konsole

Über **File/New File** legen wir uns eine neue Datei an und Speichern sie im Verzeichnis **C:/bta_py** unter dem Namen **"hello_world.py"** ab.

Inhalt der Datei:
print("hello world!")

Datei speichern (**STRG S**)

Die Eingabeaufforderung (cmd) öffnen. Ins richtige Verzeichnis (**C:/bta_py**) gehen.

cd.. *# verlässt ein Verzeichnis und wechselt ins Übergeordnete*

cd bta_py *# wechselt ins Verzeichnis bta_py*

Wenn wir im richtigen Verzeichnis angekommen sind, führen wir das Python-Script aus:
python hello_world.py

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

Editor und Konsole

Aufgabe:
Schreibe ein Programm, welches den User auffordert, seinen Namen einzugeben und den User dann mit "Guten Tag, USERNAME" begrüßt.

Aufgabe*:
Schreibe ein Programm, welches den User auffordert, einen Geldbetrag (int) einzugeben.
Jetzt soll berechnet werden, wie dieser Betrag in Scheine und Münzen zerlegt werden kann. (z. B. 85€: 1 x 50, 1 x 20, 1 x 10, 1 x 5)

Hilfe:

- Typumwandlung der Eingabe in int,
- ganzzahlige Division //
- und Modulo %

Sequenzen

Eine Sequenz ist eine Sammlung, in der eine Folge von Objekten zusammengefasst wird.
In Python unterscheidet man mehrere Arten von Sequenzen:

Strings, Bytestrings, Tupel und Listen

Die Elemente (Items) einer Sequenz sind von 0 beginnend durchnummeriert.

String s:

-4	-3	-2	-1
M	**O**	**N**	**D**
0	1	2	3

```
print(s[2])
N
print(s[-3])
O
```

Sequenzen

Tupel (Tuple):

In einem Tupel sind mehrere Objekte auch unterschiedlicher Datentypen zu einem Objekt zusammengefasst.
Es ist von der Idee her, eher **ein** komplexes Einzelobjekt, keine Aufzählung **vieler** Einzelobjekte. **Ein Tupel ist nicht änderbar!**

Name und Geburtsdatum einer Person **("Martin", "1978")**

Koordinaten eine Punktes **(4 , 5)**

Adresse **("Hauptstr.", 4, "10439", "Berlin")**

Tupel können auch Variablen enthalten:

```
anrede = "Hallo Martin"
text = "wie geht es Dir?"
zusammen = (anrede, text) # Die Klammern können weggelassen werden
print(zusammen)
```


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

Sequenzen

Liste (List):

Eine Liste ist eine Aufzählung **vieler** Einzelobjekte. Meist sind die Elemente vom gleichen Typ, müssen es aber nicht sein. **Listen sind änderbar!**

Wochentage:
```
w = ["Mo", "Die", "Mi", "Do"]
```

Ein neues Element hinzufügen:
```
w.append("Fr")

print(w)
["Mo", "Die", "Mi", "Do", "Fr"]

print(w[1])
Die
```

Sequenzen

Schleifen über Listen:

```
list = ["apple", "banana", "cherry"]
  for x in list:
   print(x)
```

Ein Element aus einer Liste entfernen:

```
list.pop() # enfernt das letzte Element
list.pop(0) # enfernt das erste Element
```

Sequenzen

Sequenzen vom gleichen Typ können konkateniert werden mit **+**:

```
z = [1, 2, 3]
x = [4, 5]
print(z + x)
[1, 2, 3, 4, 5]
```

Sequenzen können vervielfältigt werden mit *:

```
print(4*"Tag ")
Tag Tag Tag Tag
```

Bestimmung der Länge einer Sequenz:

```
s = "Sonne"
print(len(s))
5
```

```
z = [1, 2, 3]
print(len(z))
3
```

Dictionaries

Entsprechen den assoziativen Arrays aus PHP. Es handelt sich um eine Liste mit Key:Value-Paaren.

```
dict = {"Marke": "Mercedes", "Model": "CLS", "Jahr": "2018" }
```

Key Value Key Value Key Value

```
print(dict["Model"])
```

CLS

Schleifen über Dictionaries:

Gibt alle Keys aus:

```
for x in dict:
  print(x)
```

Gibt alle Values aus:

```
for x in dict:
  print(dict[x])
```

Dictionaries

Gibt alle Keys und Values aus:

```
for x, y in dict.items():
  print(x, y)
```

Kontrollstrukturen

Verzweigungen (if, if-else, if-elif)

```
if Bedingung:
 Anweisung
else:
 Anweisung
```

```
if kategorie == "Loge":
 preis = 20
else:
 preis = 10
```

```
if bedingung1:
 Anweisung
elif bedingung2:
 Anweisung
elif bedingung3:
 Anweisung
else:
 Anweisung
```

Es gibt in Python kein Switch-Case!

Kontrollstrukturen

Bedingte Ausdrücke:

```
wert1 if bedingung else wert2
```

```
gruss = "Guten Tag" if alter > 18 else "Hallo"
```

Kontrollstrukturen

While-Schleife

```
while Bedingung
  Anweisung
```

```
antwort = 'j'

while antwort == 'j':
    l = input("Länge eingben: ")
    b = input("Breite eingeben: ")
    flaeche = float(l) * float(b)
    print(flaeche)
    antwort = input("nochmal j/n ?")

print("danke")
```

Kontrollstrukturen

break beendet die Ausführung einer Schleife komplett:

```
print("Zahl zwischen 1 und 10 eingeben. ")

while True:
    zahl = int(input("Zahl: "))
    if 1 <= zahl <= 10:
        break
    else:
        print("nochmal")

print("danke")
```


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

Kontrollstrukturen

for-Schleife:

```
a = [1, 2, 3, 4]
for i in a:
    print(i*i, end=" ")


for c in "Sonne":
    print(c, end="  ")
```

for-Schleife mit range()

```
for i in range(3):
  print("Hallo", end=" ")
```

Ausgabe: Hallo Hallo Hallo

```
for i in range(10, 15):
      print(i*i, end=" ")
```

Ausgabe: 100, 121, 144, 169, 196

Funktionen

Eine Funktion ist ein Objekt, das eine bestimmte Teilaufgabe eines Programms lösen kann.

```
def summe(x, y):
    s = x + y
    return s

summe = summe(4, 5)
print(summe)
```

Voreingestellte Parameter:

```
def summe(x = 4, y = 5):
    s = x + y
    return s

summe = summe()
print(summe)
```

Voreingestellte Parameter können überschrieben werden.

Funktionen

Funktionen können eine beliebige Anzahl von Parametern übergeben bekommen:

```
def summe (*zahl):
    summe = 0
    for i in zahl:
        summe += i
    return summe

summe = summe(5, 7, 6, 10)
print(summe)
```