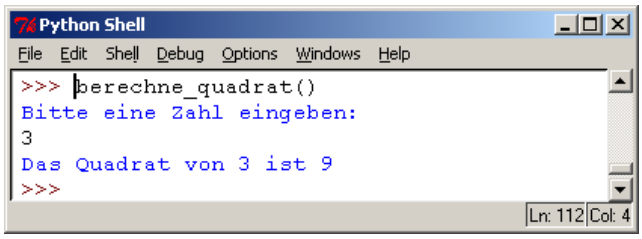
Übungen2

Teil 1: Input/Output

Tastatureingaben: `input()` und `raw_input()`

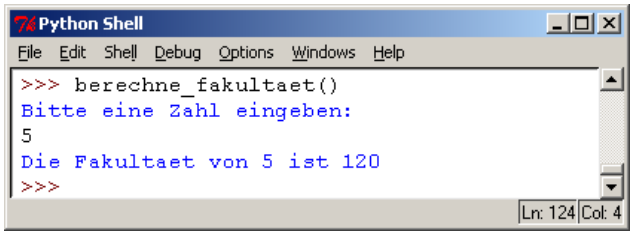
Aufgabe 1

Schreiben Sie eine Funktion **`berechne_quadrat()`**, die wie im folgenden Screenshot das Quadrat einer vom User eingegeben Zahl ermittelt:



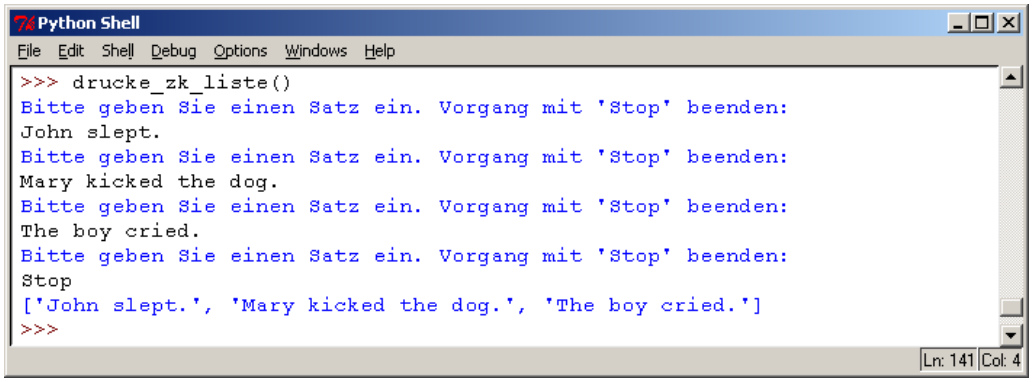
Aufgabe 2

Schreiben Sie eine Funktion **`berechne_fakultaet()`**, die wie im folgenden Screenshot die Fakultät einer vom User eingebebenen natürlichen Zahl berechnet.



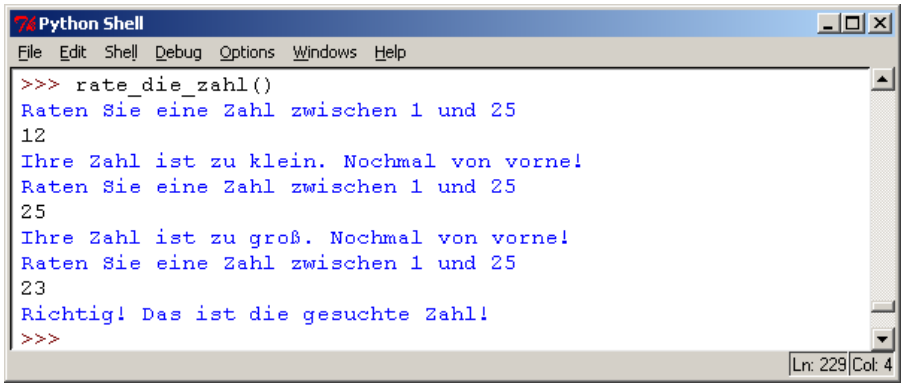
Aufgabe 3

Schreiben Sie eine Funktion **`drucke_zk_liste()`**, die wie im folgenden Screenshot den Benutzer auffordert, solange einen Satz einzugeben, bis der Vorgang durch *Stop* beendet wird. Die vom User eingegebenen Sätze sollen zu einer Liste von Zeichenkette verknüpft und anschließend gedruckt werden:



Aufgabe 4

Schreiben Sie eine Funktion **`rate_die_zahl()`**, die wie im folgenden Screenshot den Benutzer auffordert, eine Zahl einzugeben, die mit einer vorgegebenen Zahl verglichen wird solange, bis beide Zahlen übereinstimmen:



Dateien lesen und schreiben: `file(Dateiname,'r')` und `file(Dateiname,'w')`

Wenn Sie es noch nicht getan haben: lesen Sie die entsprechenden Seiten aus dem Text *Projekt 1* Abschnitt *Tokenübergreifendes Taggen: Chunk-Parsing* durch und legen Sie eine Datei namens **`datei.py`** an, die die im Text beschriebenen Funktionen **`schreibe_datei()`** und **`lese_datei()`** enthält. Sie werden dabei wahrscheinlich so vorgegangen sein:

```
def schreibe_datei(Dateiname,Objekt):
    Datei = file(Dateiname,'w')
    Datei.write(str(Objekt))
    Datei.close()
```

```
def hole_datei(Dateiname):
    Datei = file(Dateiname,'r')
    Objekt = eval(Datei.read())
    Datei.close()
    return Objekt
```

Testen Sie anschließend die folgenden Befehle aus und achten Sie genau auf die Fehlermeldungen:

1. `schreibe_datei('temp.txt','a b c d')`
   `hole_datei('temp.txt')`
2. `hole_datei('unbekannt.txt')`

Aufgabe 5

Wenn Sie **`schreibe_datei()`** so definiert haben wie oben angegeben, bekommen Sie bei **`hole_Datei()`** Probleme mit dem Einsatz von **`str()`**, wenn das zweite Argument, also der in die Datei zu schreibende Text, bereits eine Zeichenkette ist. Ändern Sie per **`repr()`** die Funktion so, dass dieses Problem nicht mehr auftritt. Nutzen Sie in Python die **`help()`**-Funktion, um mehr über **`repr()`** herauszufinden.

Aufgabe 6

Wenn Sie **`schreibe_datei()`** so definiert haben wie oben angegeben, bekommen Sie bei **`hole_Datei()`** Probleme, wenn (a) das Format des Objektes – wie gerade beim ersten Problem – nicht korrekt ist oder (b) wenn die Datei gar nicht existiert.

In Python nun gibt es eine Möglichkeit, in Kontrollstrukturen Ausnahmen zu formulieren, die unerwünschten Fehlermeldungen vorbeugen. Dafür verwenden wir die Anweisungen **`try:`** und **`except.`** Beispiel:

```
>>> def summe(N,N1):
     return N + N1
>>> summe(3,4)
7
>>> summe(3,'4')
Traceback (most recent call last):
[...]
TypeError: unsupported operand type(s) for +: 'int' and 'str'
>>> def summe(N,N1):
     try:
          return N + N1
     except:
          print "Sie müssen Zahlen eingeben"
>>> summe(3,4)
7
>>> summe(3,'4')
Sie müssen Zahlen eingeben
```

Modifizieren Sie die Funktion **`hole_datei()`** so, dass bei Problemen wie den oben beschriebenen eine entsprechende Meldung zurückgegeben wird.

Anmerkung: diese Aufgaben waren mies! Sehen Sie sich dazu die Kommentare in den Erläuterungen dieser Übungen an.

Teil 2: Arbeiten mit Dictionaries

Legen Sie für die nachstehenden Aufgaben die folgenden Dictionaries an:

```
uebung1 = {'a':'1', 'b':'2', 'c':'3', 'd':'4'}
uebung2 = {'erstes':'12', 'zweites':'34', 'drittes':56', 'letztes':'78'}
uebung3 = {'the':'DET', 'boy':'NOUN', 'kicked':'VERB', 'John':'NAME'}
```

Lassen Sie sich über die Methoden **.keys()** und **.values()** Schlüssel und Werte von **uebung1**, **uebung2** und **uebung3** anzeigen. Was fällt Ihnen auf?

Aufgabe 7

Schreiben Sie eine Funktion **drucke_keys()**, die die Schlüssel von **uebung1** als Liste in alphabetischer Reihenfolge ausgibt.

Aufgabe 8

Schreiben Sie eine Funktion **drucke_values1()**, die die Werte von **uebung1** in der richtigen Reihenfolge ausgibt (1-2-3-4).

Aufgabe 9

Schreiben Sie eine Funktion **drucke_values2()**, die die Werte von **uebung2** in der Reihenfolge ausgibt, die Sie haben wollen (z.B. erstes-zweites-drittes-letztes)

Aufgabe 10

Schreiben Sie eine Funktion **drucke_values3()**, die die Werte von **uebung2** so ausgibt, dass sie (a) in der Reihenfolge ausgibt, die Sie haben wollen (z.B. 12-34-56-78) und (b) als Zeichenkette aufgeführt werden, wobei sie jeweils durch ein Sternchen voneinander getrennt werden.

Aufgabe 11

Schreiben Sie eine Funktion **drucke_tagkette ()**, die die Werte von **uebung3** in der Reihenfolge, die Sie haben wollen, als Zeichenkette ausgibt und zwar so, dass die einzelnen Werte nun in Kleinbuchstaben aufgeführt werden und jeweils durch ein Sternchen voneinander getrennt sind.

Aufgaben 7- 11: