



Informatik für Schüler, Foliensatz 8 Wiederholung, Funktionen

Prof. G. Kemnitz

Institut für Informatik, Technische Universität Clausthal
9. Dezember 2008



Syntaxfehler (-Beseitigung)

Was bedeuten folgende Fehlermeldungen?

- File "tmp.py", line 3 ... IndentationError: unexpected indent
- File "tmp.py", line 6 SyntaxError: Non-ASCII character '\xc3'
- File "tmp.py", line 6, in <module> print x, y NameError: name 'y' is not defined

Welche Syntax-Fehler enthält das folgende Programm? Was für Fehlermeldungen sind zu erwarten? Was ist zu ändern?

```
a = y +1
if a=4:
b=b+1
    for i in (1, 5, 7): #wiederhole für drei Werte
        b += i
        c = a + i*'x';
```



Zuweisung oder Ausdruck?

Welchen Wert und welchen Typ hat das Zuweisungsziel bzw. der Ausdruck?

```
3 == 1+2
x = (2,)
x += (3,)
x *= 2
x[:-2]
b = 'hallo'
b+x
f = open('test.txt', 'rb')
z = f.read()
```

Was ist Syntax-Highlight (Hervorhebemodus) und wo und wie aktiviert man ihn?

Kontrollfluss

Was steht am Ende in dem ausgedruckten String s?

```
s='a'  
for x in range(3):  
    s += 'x'  
    for y in range(2):  
        s += 'y'  
for idx in s:  
    s += '*'  
print s
```

Wieviele Zeilen werden ausgegeben und was steht in jeder Zeile?

```
for a in (0, 1, 2):  
    for b in (True, False):  
        print a, b
```



Funktion

Eine Funktion

- ist eine Verallgemeinerung eines Ausdrucks
- berechnet aus den Werten der Argumente (arg) ein Ergebnis.
- Das Ergebnis wird als Funktionswert zurückgegeben und hat einen Typ.
- Die Argumente müssen auch bestimmte Typen haben.

Syntax:

functionsname ([arg {, arg}])

Welchen Datentyp haben die folgenden bereits genutzten Funktionen und ihre Operanden?

- `math.cos(x)`
- `type(x)`
- `range([a,] e[, s])`
- `open('Text1', 'Text2')`



Definition eigener Funktionen

```
def functionsname([arg {, arg}]):  
    {Berechnungsschritt}  
    return Ausdruck
```

Beispiel Increment-Funktion: Berechnung des um »1« erhöhten Werts; anwendbar auf alle Datentypen , für »x+1« zulässig ist

```
def inc(x):  
    return x+1
```

Was wird ausgegeben?

```
print inc(5), print inc(0.7)  
print inc('Text')
```

- Eine Funktion muss vor Benutzung definiert sein
- wird durch einen eingerückten Block beschrieben.
- endet wenn die Einrücktiefe zurückgesetzt wird
- kann Ein- und Ausgaben an das Dateisystem enthalten.

Aufgabe 8.1: Ausdruck durch Funktion ersetzen

Ersetzen Sie in dem Program:

```
for idx in range(20):  
    x= (idx-10)*0.5  
    y= (x*x + x + 1 ) # den Ausdruck rechts ersetzen  
    print '%4.1f : %4.1f |' %(x, y) + int(2*y)*'.' + '*'
```

den Ausdruck für die Funktion $x^2 + x + 1$, durch eine selbst zu definierende Python-Funktion.

- Erst Vorgabeprogramm verstehen und testen, dann ändern.

Hinweis: Die Teilzeichenketten »%4.1f« sind sog. Format-Strings. Sie beschreiben, wie die Gleitkommawerte von x und y als Text darzustellen sind, im Beispiel mit insgesamt vier Stellen, davon einer Nachkommastelle.



Aufgabe 8.2: Umlaute beseitigen

Bei Eingabe der folgenden Umlaute-Zeichenkette im interaktiven Python-Modus:

```
'äöüÄÖÜß'
```

wird als Wert des Ausdrucks ein doppelt so langer String aus Hexadezimalwerten angezeigt:

```
'\xc3\xa4\xc3\xb6\xc3\xbc\xc3\x84\xc3\x96\xc3\x9c\xc3\x9f'
```

Die Umlaute werden offenbar alle durch `'\xc3'` gefolgt von einem Byte mit einem Wert größer `»\x7f«` (dezimal 127), das den Umlaut beschreibt, dargestellt. Um `»ä«` durch `»ae«`, `»ö«` durch `»oe«` etc. zu ersetzen, sind offensichtlich die Zeichen `'\xc3'` zu löschen, `'\xa4'` durch `'ae'` etc. zu ersetzen und alle normalen Zeichen zu übernehmen:



```
zin=raw_input('Eingabe:')
zout =''
for c in zin:
    if c == '\xc3':
        pass                # löschen von '\xc3'
    elif c == '\xa4':
        zout += 'ae'        # ersetzen von 'ä'
        ...                 # Ersetzen von ö etc.
    else:
        zout += c           # sonst übernehmen
print zout
```

- Testen Sie das Programm und versuchen Sie es zu verstehen.
 - Umlaute in den Kommentaren? \Rightarrow ...
 - »pass« ersetzt einen eingerückten Block und tut sonst nichts
 - »raw_input('Text')« liest die Eingabe in eine Zeichenkette (keine Weiterverarbeitung wie bei der Funktion input())



Hinweise:

- Erweitern Sie das Programm so, dass es auch die restlichen Umlaute ersetzt.
- Ersetzen Sie die gesamte Schleife für die Umlautersetzung durch eine selbst zu definierende Funktion mit der Zeichenkette mit Umlauten als Eingabe und der Zeichenkette ohne Umlaute als Rückgabewert.
- Schreiben Sie das Programm so um, dass sie es dazu benutzen können, in ihren eigenen Python-Programmen bei Bedarf die störenden Umlaute in darstellbare ASCII-Zeichen umzuwandeln.
- Heben Sie sich das Programm für die nächste Klausur auf.