

- Zur Einrückung der "Blöcke" in der `if-elif-else`-Anweisung. Zwei davon bestehen nur aus einer einzigen Anweisung.
- der `if`-Block und der `else`-Block enthalten eine Parallel-Zuweisung, der andere Block enthält eine mehrfache Zuweisung.
- Prüfung auf Gleichheit geschieht mit `==`, fortlaufende Vergleiche wie `2<=len(wort)<=3` sind möglich.

5. Funktionsdefinitionen

Funktionsdefinitionen werden in Python mit dem Schlüsselwort `def` eingeleitet. Funktionen können natürlich Parameter haben und daher Argumente übernehmen. Wie probieren wieder ein ganz einfaches Beispiel mit dem Interpreter aus. Achten Sie wieder auf die gleichmäßige Einrückung des Funktionsblocks:

```
>>> def doppelt(x):
    d = 2 * x
    print "Das Doppelte von", x,"ist",d
```

```
>>> doppelt(5)
Das Doppelte von 5 ist 10
doppelt funktioniert wie eine Prozedur. Angemessener ist hier vielleicht ein "echte" Funktion mit Rückgabewert. Dazu benötigen wir die return-Anweisung:
```

```
>>> def doppelt(x):
    return 2 * x
>>> doppelt(5)
10
>>> doppelt(doppelt(5))
20
```

Der letzte Aufruf wäre natürlich mit der ersten "Prozedur"-Version nicht möglich gewesen.

Nützen wir diese Möglichkeit, um unser Script für die Lösung quadratischer Gleichungen zu ändern, indem wir eine Funktion zu definieren. (Im Edit-Fenster hilft dabei *Edit/Indent region.*)

```
from math import sqrt
def quagl(a,b,c):
    determinante = b**2 - 4*a*c
    if determinante > 0:
        d = sqrt(determinante)
        return (-b+d)/(2.0*a),(-b-d)/(2.0*a)
    elif determinante == 0:
        return -b/(2.0*a)
    else:
        return ()
```

Nun können wir quadratische Gleichungen mittels Funktionsaufrufen lösen:

```
>>> quagl(4,-8,3)
(1.5, 0.5)
>>> quagl(1,2,1)
-1.0
>>> quagl(1,0,4)
()
```

Dabei sehen wir, dass Python-Funktionen mehrere Werte (als Tupel) zurückgeben können.

Neben der `for`-Schleife kennt Python auch die `while`-Schleife, deren Syntax nach dem bisher gesagten eigentlich keiner weiteren Erläuterung bedarf. Sie lässt sich beispielsweise im Euklidischen Algorithmus für die Berechnung des größten gemeinsamen Teilers zweier Zahlen an-

wenden. Mittels Parallelzuweisung lässt sich der in Python sehr elegant formulieren.

```
>>> def ggt(a,b):
    while b>0:
        a,b = b,a%b
    return a
>>> ggt(105, 42)
21
```

6. Funktionen als Objekte

Zuletzt möchte ich noch zeigen, dass sich Funktionen (in Python sind Funktionen auch Objekte) als Argumente an andere Funktionen übergeben lassen:

```
>>> def tabelle(von, bis, fun):
    for x in range(von,bis+1):
        print x, fun(x)
>>> tabelle(13,16,doppelt)
13 26
14 28
15 30
16 32
>>> def cubus(n):
    return n**3
>>> tabelle(27,29,cubus)
27 19683
28 21952
29 24389
>>>
```

In der nächsten Folge möchte ich zeigen, was die Feststellung bedeutet: *"In Python ist alles ein Objekt"*, und etwas OOP treiben: wie man in Python selbst Klassen definieren und Objekte erzeugen kann.

7. Quellen

Wenn Sie sich etwas näher mit Python beschäftigen möchten, können Sie dafür viele Ressourcen in Anspruch nehmen.

- (1) Vielfältig und gratis ist alles, worauf auf der Python-Website "for beginners" verwiesen wird: <http://www.python.org/doc/Newbies.html>. Für das meiste davon sind Englischkenntnisse erforderlich.
- (2) An deutschsprachigen Büchern bieten sich an:
- (3) Mark Lutz und David Ascher: Einführung in Python, OReilly, 2000 (Englische Neuauflage in Vorbereitung.)
- (4) Gregor Lingl: Python für Kids, mitp, 2003. Gedacht als Einführung ins Programmieren für Anfänger, unter Verwendung von Python. Siehe dazu auch: <http://python4kids.net>
- (5) von Löwis/Fischbeck: Python 2, Addison-Wesley, 2001. Ist keine Einführung, sondern anspruchsvoll und tiefergehend, enthält aber noch nicht die Neuerungen von Python seit Version 2.1.

Für Ambitionierte dürfte derzeit am aktuellsten und ergiebigsten sein:

- (5) Alex Martelli: Python in a Nutshell, OReilly, 2003. (Englisch)



Computer & Software Systeme GmbH

Ihr Spezialist für Computer & Netzwerke



PC-Systeme

Langjährige Kunden schwören auf die Qualität unserer PC-Systeme. Stabile Hardware senkt Ihre EDV-Gesamtkosten

Netzwerkösungen

Von der Konzeption bis zum fertig installierten Netzwerk, das alle Stücke spielt, sowie für die Administration und Wartung von Netzwerken sind wir Ihre kompetenten Partner.

Service - Wartung

EDV-Probleme stören Ihren Arbeitsablauf, kosten Energie, Nerven und Geld.

Wir garantieren Ihnen:

- regelmäßige Routine-Checks
- kurze Reaktionszeit
- kompetente Störungsbehebung

Kaufmännische Software

- Auftragsbearbeitung
- Faktura
- Lager
- POS Kassa
- Finanzbuchhaltung



A - 1 0 9 0 W i e n
R ö g e r g a s s e 6 - 8
Tel: +43/1/310 99 74-0
Fax: +43/1/310 99 74-14
email:office@excon.at
www.excon.at