Informatik für Schüler, Foliensatz 3 Vorhandene Funktionen und Wiederholschleifen

Prof. G. Kemnitz

Institut für Informatik, Technische Universität Clausthal

23. September 2009

Funktionen

- Eine Funktion berechnet aus den Werten der Operanden ein Ergebnis, dass einer Variablen zugewiesen oder in einem Ausdruck wie ein Teilausdruck verwendet wird; eine Art großer Berechnungsschritt, der in Wirklichkeit selbst ein Programm ist
- Python besitzt für viele Aufgaben fertige programmierte Funktionen, die nur noch verwendet zu werden brauchen; Beispiele:
  - Bestimmen des Datentyps:

    ```
    type(x)
    Funktionswert ist eine Zeichenkette mit der
    Typangabe
    ```

- Hilfe zu einem Datenobjekt, einem Datentyp, einer Funktion oder einem anderen Objekt, das einen Namen hat:

  `help(`*`Objektname`*`)`
  `Hilfetext mit »q« schließen`

- Betrag von x:

  `abs(x)`
  `Argument darf int oder float sein; der Funktionswert hat denselben Typ`

- Länge einer Zeichenkette s

  `len(s)`
  `Diese Funktion gibt es für alle Datentypen, die aus mehreren gleichen Elementen bestehen; der Funktionswert hat den Typ int`

Experiment: Ausprobieren einiger Funktionen

Welcher Wert und welcher Typ wird den folgenden Variablen zugewiesen?

```
v1 = 5 * '*'
v2 = len(v1)- 25
v3 = len(v1 + '25')
v4 = abs(v2)
```

- Typabfrage immer mit der Funktion type(...)
- Wertabfrage: Ausdruck, der keiner Variablen zugewiesen wird; eine einzelne Variable ist auch ein Ausdruck

Funktionen aus externen Modulen

- Die meisten fertigen Funktionen stehen in Zusatzmodulen; viele mathematische Zusatzfunktionen z.B. im Zusatzmodul »math«
- Import des Moduls »math« und Nutzung der Funktion cos():

```
import math
y = math.cos(math.pi/2)
```

  - Objekten aus Modulen Modulnamen vorangestellen
- Abfrage aller im Modul »math« vereinbarten Funktionen und Variablen

```
help(math)
```

  - zeigt die bereitgestellten Funktionen an; mit »:q« beenden

```
help(math.cos)
```

  - Hilfetext für eine einzelne Funktion; mit »:q« beenden

Experiment: Berechnung von $y = \cos(0.1 \cdot \pi)$

```
import math
x = 0.1 * math.pi
y = math.cos(x)
print 'x =', x,  '   cos(0.1*x)' , y
```

Schleifen

Erweiterung des ersten Beispielprogramms so, dass es die Funktionswerte für mehrere Eingabewerte berechnet:

- Lösung 1: mehrfaches Kopieren der Anweisungsfolge und anpassen der Werte von x:

```
x = 0.1 * math.pi
y = math.cos(x)
print 'x =', x, 'cos(x) = ', y
x = 0.2 * math.pi
y = math.cos(x)
print 'x =', x, 'cos(x) = ', y
...
```

- Lösung 2 Schleife: Anweisung an den Rechner, dass die Befehlsfolge mit unterschiedlichen Werten für x mehrfach auszuführen ist

```
for sv in sequenz:
    Anweisungsfolge
```

Wiederhole für x in *sequenz*
$y = \cos(0.1 \cdot x)$
print 'x =', x, 'cos(0.1*x) =', y

sv – Schleifenvariable;
Anweisungsfolge einrücken; Mögliche Sequenzobjekte in Python:

- Tupel:

  ```
  (0, 1, 2, 3, 4)
  ```

- Liste:

  ```
  [0, 1, 2, 3, 4]
  ```

- Funktion zur Erzeugung einer Sequenz, z.B.:

  ```
  range([a,] e [,s])
  ```

  Erzeugt eine Liste der ganzen Zahlen kleiner »e« beginnend mit »a« , Schrittweite »s«; ohne »a« und »s« der natürlichen Zahlen kleiner »e«;

  ```
  range(5)          ⇒[0, 1, 2, 3, 4]
  range(1, 5, 2)    ⇒[1, 3]
  ```

Berechnen einer Tabelle von Kosinuswerten

```
import math
for idx in range(9):
  # Kommentar: Beginn des Schleifenkoerpers
  x = idx*math.pi/16
  y = math.cos(x)
  print 'x =',x,' cos(x) = ', y
  # Kommentar: Ende des Schleifkoerpers
print 'Schleife abgearbeitet'
```

- die Anweisungen des Schleifenkörpers müssen in der Programmdatei um eine einheitliche Anzahl von Zeichen eingerückt sein
- Kommentare werden bei der Abarbeitung ignoriert. Sie beginnen mit »#« und gehen bis zum Zeilenende.

Aufgabe 3.1: Mathematische Berechnungen

- Schreiben Sie ein Python-Programm, dass die Ableitung der Sinusfunktion an der Stelle x (x – variabler Eingabewert) nach folgendem Algorithmus[(1)] berechnet und das Ergebnis ausgibt:

$$
\begin{aligned}
h &= 0.0001 \\
y(x) &= \sin(x) \\
y(x+h) &= \sin(x+h) \\
y'(x) &= \frac{y(x+h) - y(x)}{h}
\end{aligned}
$$

- Testen Sie das Programm mit Beispielwerten und kontrollieren Sie die Ergebnisse mit dem Taschenrechner.
- Verzeichnis »Uebung3«, Dateiname AblSin.py

Hilfestellungen

- Variablennamen beginnen mit einem Buchstaben und dürfen keine Sonderzeichen wie Klammern, Operationssymbole etc. enthalten.
- Eingabeanweisung:

```
x = input('x =')
```

- Linux-Kommandos:

```
man [Befehl]
ls [-l][-a] [Pfad][Datei]
mkdir Verzeichnisname
cd Pfad
python AblSin.py
```

- zu (1): Ein Algorithmus ist eine Schritt für Schritt abarbeitbare Vorschrift.

Aufgaben 3.2: Anwendung von Schleifen

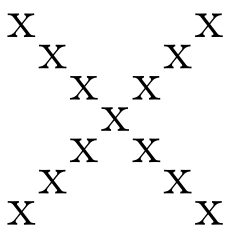
- Schreiben Sie ein Programm, dass für die Werte von 1 bis 10 eine Tabelle der Quadratzahlen erzeugt.
- Schreiben Sie ein Programm mit einer Schleife, die 16-mal durchlaufen wird. Beim ersten Durchlauf soll die Zeichenkette '*', beim zweiten Durchlauf '**', beim dritten Durchlauf '***' etc. ausgegeben werden, so dass insgesamt ein Dreieck auf dem Bildschirm ausgegeben wird.

Hilfestellung: Anweisung zur Ausgabe einer Zeichenkette aus n Sternchen:

```
print n *  '*'
```

Aufgaben 3.3: Algorithmus gesucht!

- Schreiben Sie nach demselben Schema ein Programm, dass ein Kreuz entsprechend nachfolgender Abbildung auf den Ausgabebildschirm zeichnet.



Die Aufgabe darf so abgewandelt werden, dass ein anderes regelmäßig strukturiertes Ausgabebild erzeugt wird (Rechteck, Trapez, Rombus etc.). Wer sich einen Pluspunkt verdienen möchte, zeichnet einen Weihnachtsbaum aus mehreren Dreiecken.

Aufgaben 3.4: Kreisbogen zeichnen

Schreiben Sie einen Programm, dass in folgender Weise einen Kreisbogen zeichnet:

```
-10 *
 -9 ............*
 -8 ................*
 -7 ....................*
 -6 ........................*
 -5 ..........................*
 -4 ..........................*
 -3 ............................*
 -2 ............................*
 -1 ..............................*
  0 ..............................*
  1 ..............................*
  2 ............................*
  3 ............................*
  4 ..........................*
  5 ..........................*
  6 ........................*
  7 ....................*
  8 ................*
```

Hinweise:

- Dateiname: Kreis.py, Verzeichnis Ueb3
- Für einen Kreisbogen gilt der Satz des Pythagoras:

$$x^2 + y^2 = r^2$$

  im Beispiel mit $r = 10$, nach y aufzulösen.
- benötigte Funktionen:

```
x**2            # Quadrat von x
math.sqrt(x)    # Wurzel von x
round(x)        # nächster ganzzahliger Wert
int(x)          # Umwandlung in eine ganze Zahl
'%3d' %x        # formatierte Umwandlung in eine
                # 3-Ziffern-Zeichenkette mit
                # führenden Nullen
```

Aufgaben 3.5: Kosinusfunktion zeichnen

```
-3.46 .*
-3.11 *
-2.76 ..*
-2.42 .......*
-2.07 ...............*
-1.73 ........................*
-1.38 ..................................*
-1.04 ..............................................*
-0.69 ....................................................*
-0.35 ..........................................................*
 0.00 ..............................................................*
 0.35 ..........................................................*
 0.69 ....................................................*
 1.04 ..............................................*
 1.38 ..................................*
 1.73 ........................*
 2.07 ...............*
 2.42 .......*
 2.76 ..*
 3.11 *
 3.46 .*
```

Hinweise:

- Dateiname: Kosinus.py, Verzeichnis Ueb3
- Ausdruck für die Umwandlung des Wertes von x in einen rechts ausgerichtete Zeichenkette:

```
'%5.2f' % x
```