



# Informatik für Schüler, Foliensatz 6 Indizierung von Sequenzobjekte

Prof. G. Kemnitz

Institut für Informatik, Technische Universität Clausthal  
13. November 2008

## Sequenzen (Typen aus mehreren Elementen)

- String: von Zeichen, z.B.:

```
s = 'Hallo'  
text = "Zeichenkette mit dem Zeichen: ' "
```

- Tupel: von beliebigen Datenobjekten (runde Klammern)

```
t = (1, 2, 'Hallo') # Tupel: 2×»int«, 1× »str«  
lt= () # leeres Tupel  
type(lt) # Welchen Wert hat die Funktion?  
t1= (1,) # Komma zur Unterscheidung von einem  
# geklammerten Ausdruck
```

- Liste: wie ein Tupel, aber mit zusätzlichen Operationen zum Verändern, Einfügen etc. einzelner Elemente (eckige Klammern)

```
l = [1, 'a', s, t] # 1×»int«, 2×»str«, 1×»tuple«
```



## Sequenzoperationen

```
t = (1, 2, 'Hallo')
```

- Verwendung als Schleifeniterador

```
for idx in t:  
    print idx
```

- Abfrage der Länge (Anzahl der Elemente):

```
len(t)
```

- Auswahl eines Elements (Indizierung)

- `t[0]` selektiert das erste Element
- `t[1]` selektiert das zweite Element
- `t[-1]` selektiert das letzte Element
- `t[-2]` selektiert das vorletzte Element



```
t = (1, 2, 'Hallo')
```

- Indizierung eines Elements einer Sequenz, das eine Sequenz ist

```
t1 = t[-1] # ergibt 'Hallo'
```

```
t2 = t[1] # ergibt 'a'
```

```
t3 = t[-1][1] # ergibt auch 'a'
```

- Verkettung: Zusammenfassen von Sequenzen desselben Typs

```
a = 'Hallo '+' Welt' # => 'Hallo Welt'
```

```
b = t + ('Welt',) # => (1, 2, 'Hallo', 'Welt')
```

- Teilbereichsbildung (slicing)

```
t[i:j] # Auswahl der Element i bis j
```

```
t[i:] # Auswahl Element i bis Ende
```

```
t[:j] # Auswahl Anfang bis Element j
```

```
t[:] # Auswahl aller Elemente
```

$$0 \leq i, j < \text{len}(t)$$



## Beispiele

```
t = 'Asterix'
t[0]           # => 'A'
t[2]           # => 't'
t[2:6]         # => 'teri'

x = (1, ' ', 'hallo', 7)
x[2]           # => 'hallo'
x[2:4]         # => ('hallo', 7)
x[2][2:4]      # => 'll'
x[2] + x[1] + t # => 'hallo Asterix'
```

## Aufgabe 6.1: Tupel2String

Gegeben ist ein Tupel aus Zeichenketten, wobei jede Zeichenkette ein Wort eines Satzes ist, z.B.:

```
x = ('Das', 'ist', 'ein', 'Text')
```

Entwickeln Sie ein Programm mit dem Namen »Tupel2Str.py«, das aus einem solchen Tupel eine Zeichenkette des Satzes erzeugt. Zwischen den Worten sind Leerzeichen einzufügen und der Satz soll mit einem Punkt enden. Für das Beispiel lautet das auszugebende Ergebnis:

```
y = 'Das ist ein Text.'
```

**Hausaufgabe:** Zeichnen des Struktogramms

## Lösungsfindung

```
x = ('Das', 'ist', 'ein', 'Text')  
⇒ y = 'Das ist ein Text.'
```

- Es wird eine Schleife benötigt, die für alle Elemente bis auf das letzte das Element gefolgt von einem Leerzeichen an die Ausgabezeichenkette anhängt:

```
for Element in x[:-2]:  
    y = y + Element + ' '
```

- Vor dem ersten Schleifendurchlauf muss die Zeichenkette leer sein:

```
y = ''
```

- Das letzte Element wird gefolgt von einem Punkt angehängt:

```
y = y + x[-1] + '.'
```



- Das Programm soll mit der Eingabe des Tupels beginnen und mit einer Ausgabe des Ergebnistextes enden:

```
x = input('Tupel x=')  
...  
print ('Ergebnis y =', y)
```

- Linux-Kommandos:

```
man [Befehl]  
ls [-l][-a] [Pfad][Datei]  
mkdir Verzeichnisname  
cd Pfad  
python Tuple2Str.py
```





## Aufgabe 6.2: Umwandlung in Spiegelschrift

Schreiben Sie ein Programm »Spiegel.py«, das eine Textzeile einliest, die Reihenfolge der Zeichen umkehrt und den Text in Spiegelschrift ausgibt.

Lösungshinweise:

- Erzeugen einer leeren Zeichenkette

```
y = ''
```

- Schleife, die für alle Zeichen beginnend mit dem ersten an das Zeichen die restliche Zeichenkette anhängt

```
for z in x:  
    y = z + y
```

**Hausaufgabe:** Zeichnen des Struktogramms des gesamten Programms mit Ein- und Ausgabe



## Aufgabe 6.3: Zählen der ganzen Zahl in einem Tupel

Die Eingabe soll ein Tupel sein, dessen Elemente ganze Zahlen, Gleitkommazahlen und Zeichenketten sein dürfen, z.B.:

```
x = (1, 0.2, 5, 'Text')
```

Schreiben Sie ein Programm »CountInt.py«, dass die Anzahl der ganzen Zahlen innerhalb des Tupels – im Beispiel wären das zwei – bestimmt.

**Lösungshinweis:** Benötigt werden:

- Eingabe
- Wiederholschleife für alle Elemente des Tupels
- Fallunterscheidung mit einem Typenvergleich:

```
if type(Element)==type(1): ...
```

- Ausgabe



## Aufgabe 6.4: Zählen der Zeichen in einem Tupel

Die Eingabe sein ein Tupel, dessen Elemente Zeichenketten und  
Tupel aus Zeichenketten sein dürfen, z.B.:

```
x = ('Hallo', 'Paul', (',', 'was', 'tust'),  
     'du', ('gerade', '?'))
```

Schreiben Sie ein Programm »CountChar.py«, dass die Anzahl  
der Zeichen innerhalb des Tupels bestimmt, im Beispiel 26.

**Lösungshinweis:** Das Programm benötigt mindestens zwei  
ineinander verschachtelte Schleifen

- wiederhole für alle Tuppelemente
  - wenn das Element ein Tupel ist
    - wiederhole für alle Elemente des Elements
      - tue etwas