

Hilfe zu Strings

"Strings" sind Zeichenketten, eine Aneinanderreihung von Zeichen, also prinzipiell das, was man für Texte braucht. Es gibt für die ihnen zu Grunde liegenden Zeichensätze zwei wichtige Standards, nämlich einmal den traditionellen Zeichensatz ASCII [American Standard Code for Interchange of Information] und den moderneren Unicode. Die Unterscheidung soll hier nicht behandelt werden. Aber:

Keine Zeichen!

Aus einem Python-Buch: *"Man beachte, dass Python keine Zeichen im eigentlichen Sinne kennt; 'a' ist ebenfalls eine Zeichenkette."* Auf ein Zeichen in diesem Sinne kann durch einen Index zugegriffen werden. Diese Indizes beginnen bei 0. Ungewöhnlich ist aber der Wert eines solchen Zugriffs:

```
wort='Zeichenkette'  
wort[0]    → 'Z'                bzw.  
wort[3]    → 'c'                ,  
also jeweils wieder eine Zeichenkette!
```

Beispiele für das Arbeiten mit strings

Datei: <StringBeispiele.py>

Ein Beispiel für das Anwenden einer Funktionen aus dem Modul str:

```
x='Test-Wort'  
teilwort=x.rpartition('-')  
print teilwort  
liefert → ('Test', '-', 'Wort')
```

(Fast) dasselbe mit "Regular Expressions" [pattern matching], dazu muss man aber erst das Modul importieren:

```
import re  
aufteilung=re.split('-',x,1)  
print aufteilung  
liefert → ['Test', 'Wort']
```

Das ist formal etwas anders, nämlich statt eines Tupels eine Liste und das Trennzeichen fehlt.

Regular Expressions sind sehr vielseitig. *Bitte nachlesen in der Hilfe!*

Nur ein Beispiel, das ausnutzt, dass der Punkt auf alles passt:

```
aufteilung=re.split('.',x,1)  
print aufteilung  
liefert → [", 'est-Wort']
```

Es wird also das erste Zeichen abgetrennt, weil das erste Zeichen sofort auf das "Muster" passt. Die 1 am Ende des Funktionsaufrufes sagt nur aus, dass nur einmal nach dem Zeichen gesucht wird.

Will man den Punkt wirklich als Trennzeichen haben, muss man ihn als Sonderzeichen kennzeichnen. Dazu dient ganz allgemein – also auch für andere Sonderzeichen – der backslash.

```
dateiname='musterdaten.txt'  
print re.split('\.',dateiname,1)  
liefert → ['musterdaten', 'txt']
```

Ein Anwendungsbeispiel

Alle Zeichen eines strings sollen durch ein anderes ersetzt werden. Dazu definieren wir hier eine Funktion und eine Hilfsfunktion:

```
def ZeichenErsetzen(wort):
    if len(wort)==0: return ''
    else:
        kopf=wort[0]
        rest=re.split('.',wort,1)
        ergebnis=ErsetzungsFunktion(kopf)+ZeichenErsetzen(rest[1])
    return ergebnis
def ErsetzungsFunktion(zeichen):
    return zeichen.upper()
print ZeichenErsetzen("Hund und Katze")
liefert → 'HUND UND KATZE'
```

Die hier gewählte ErsetzungsFunktion ist natürlich unsinnig, sie dient nur als Musterbeispiel. Das könnte man genauso direkt mit upper() machen.

```
neu="Hund und Katze"
print neu.upper()
liefert ebenso → 'HUND UND KATZE'
```

Für die "Übersetzer"-Gruppe

Die Hilfe [zur Python Standard Library] enthält im Abschnitt [6.8. Mapping Types — dict](#)

wichtige Informationen und die richtige Datenstruktur für eure Aufgabe.

Dictionary ist euch als Begriff sicher bekannt und es handelt sich auch tatsächlich um eine Datenstruktur für Wörterbücher. Wie aber in der Informatik notwendig, hat dieses Dictionary eine fest vorgegebene Struktur, die in einer Klasse mit vielen zugehörigen Methoden ausmodelliert worden ist.

Ein Dictionary besteht prinzipiell aus einer Liste (Tupel) von Paaren. Bei den Paaren ist das erste Element das Schlüsselwort (key) und das zweite der zugehörige Wert (value).

Eine kurze Übersetzung aus der Hilfe:

Dictionaries can be created by placing a comma-separated list of key: value pairs within braces, for example: {'jack': 4098, 'sjoerd': 4127} or {4098: 'jack', 4127: 'sjoerd'}, or by the dict constructor.

Bedeutet: Man erzeugt ein Dictionary beispielsweise durch die Angabe einer Liste – Trennzeichen ist das Komma – von Paaren aus key : value – der Doppelpunkt ist hier das Trennzeichen. Anfang und Ende der Liste selbst sind mit den geschweiften Klammern gekennzeichnet.

Als Alternative kann man den Konstruktor der Klasse dict aufrufen.