

# Zeichenketten

## Information

Zwei Zeichenketten können miteinander verbunden werden. Durch den Befehl  
`text3 = text2 + text1`  
werden die Zeichenketten `text2` und `text1` aneinander gehängt und das Ergebnis in `text3` abgespeichert. Diese Verbindung bezeichnet man als **Verkettung** (oder vornehmer: Konkatenierung; lateinisch: catena = die Kette)

**Beispiel:** Gegeben sind `text1 = "Hallo"` und `text2 = "Welt"`. Dann wird in der Variablen `text3` die Zeichenkette "HalloWelt" abgespeichert.

**Merke:** Bei Zeichenketten hat das PLUS-Zeichen eine andere Bedeutung als bei Zahlen. Beim Kompilieren schaut BASCOM deswegen auf die Deklaration. Liegen zwei Zeichenketten vor, dann wird verkettet; liegen hingegen zwei Zahlen vor, dann wird addiert.

1. Gegeben sind die beiden Zeichenketten Zk1 und Zk2 aus Abb. 1

- 1.1 Zeichne die beiden möglichen Zeichenketten, die sich aus der Verkettung ergeben können. Wie lauten die entsprechenden Befehle, die zu diesen neuen Zeichenketten Zk führen?

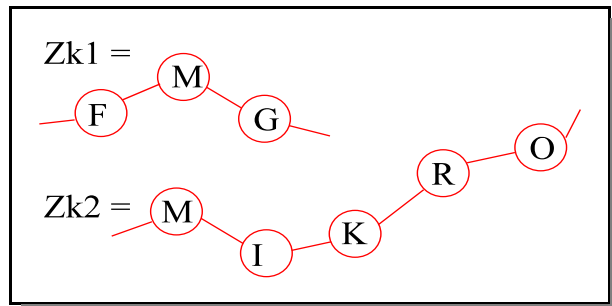


Abbildung 1

- 1.2 Was wird BASCOM wohl tun, wenn Du versuchst eine Zeichenkette mit einer Zahl durch ein PLUS zu verknüpfen? Stelle eine Vermutung auf.

# Zeichenketten

2\*. Betrachte das folgende Programm:

```

Do
  Input Text1 Noecho
  Text1 = Text1 + " ENDE"
  Print Text1;
Loop
    
```

- 2.1 Wozu dient der Parameter "Noecho"? Welchen Zweck hat das Semikolon? Wie muss die Variable Text1 deklariert werden, damit Zeichenketten mit maximal 8 Buchstaben gespeichert werden können?
- 2.2 Schreibe das Programm und teste es aus. Lasse auch einmal "Noecho" und das Semikolon aus und vergleiche!
- 2.3 Ersetze " ENDE" durch die Zahl(!) 67. Kompiliere! Vergleiche das Ergebnis mit Deiner Vermutung aus 1.2.
- 2.4 Jetzt sollen nacheinander zwei Zeichenketten eingeben und deren Verkettung ausgegeben werden (Bei der Eingabe für jede Zeichenkette CR senden!). Ändere das Programm entsprechend ab und teste es aus.
- 2.5 Das Programm aus 2.4 soll nun automatisch ein Leerzeichen zwischen die beiden Zeichenketten einfügen.  
**Beachte** : BASCOM kann in **einem** Befehl immer nur **eine einzige** Verknüpfung (Addition, Subtraktion, Verkettung...) vornehmen!

3\*. Schreibe ein Programm, welches folgenden Dialog führen kann:

Attiny meldet	Anwender gibt ein	
Wie heißt Du?		
	Elsa*	
Hallo Elsa*! Geht es Dir gut?		
	Ja	Nein
Schön!	Schade	

\*) oder ein anderer Name

# Zeichenketten

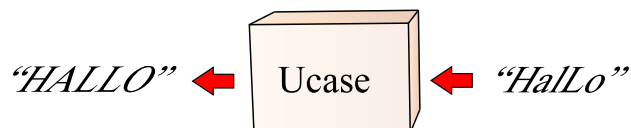
Funktionen kennst Du aus der Mathematik: Zuordnungen, z. B. Lineare Funktion, quadratische Funktion, aber auch die Zuordnung Tageszeit -> Lufttemperatur.

Auch beim Programmieren benutzt man **Funktionen**. Auch hierbei handelt es sich um **Zuordnungen**. Am Beispiel der UCASE-Funktion machen wir dies deutlich:

```
dim s1 as string*10
dim s2 as string*10
s1 = "HalLo"
s2 = Ucase(s1)
```

In diesem Fall wird durch die Funktion Ucase der Zeichenkette "HalLo" die Zeichenkette "HALLO" zugeordnet; diese Zeichenkette "HALLO" wird anschließend in der Variablen s2 gespeichert.

Die Funktion Ucase wandelt also alle Buchstaben von s1 in Großbuchstaben um. (Übrigens: Ucase ist eine Abkürzung für uppercase = Großbuchstaben.) Ganz allgemein können wir uns Funktionen als Maschinen vorstellen, die mit etwas gefüttert werden und dieses auf bestimmte Art und Weise verarbeiten.



Womit sie gefüttert werden (in unserem Fall die Zeichenkette "HalLo") bezeichnet man im Englischen als **Input**, im Deutschen als **Argument** oder **Parameter**. Was sich durch die Verarbeitung ergibt, nennt man im Englischen den **Output**, im Deutschen **Funktionswert** oder **Rückgabewert**.

**Beachte:** Durch den Befehl `s2 = Ucase(s1)` wird also

1. Der Input, d. h. s1, in eine Zeichenkette mit Großbuchstaben umgewandelt und
2. Das Ergebnis dieser Umwandlung (der Output) in der Variablen s2 gespeichert.

# Zeichenketten

Weitere Funktion für den Umgang mit Zeichen bzw. Zeichenketten:

Lcase(zk)	wandelt eine Zeichenkette zk in Kleinbuchstaben um
Chr(c)	wandelt eine Zahl c (vom Typ byte) in das zugehörige ASCII-Zeichen (vom Typ string*1) um
Mid(zk,p,a)	kopiert a Buchstaben aus einer Zeichenkette zk, beginnend mit der Position p (vgl. die Beispiele unten)
Len(zk)	Gibt die Länge der Zeichenkette zk zurück, d. h. die Anzahl ihrer Zeichen

## Beispiele:

Gegeben sind:      zk = "Hallo Welt!"      c = 65      p = 7      a = 2

Lcase(zk)	"hallo welt!"
Chr(c)	"A"
Mid(zk,p,a)	"We" (Die Zählung beginnt also mit 1, nicht mit 0; und Leerzeichen werden mitgezählt!)
Len(zk)	11

4. Gegeben sind: zk = "INFO ist toll."      c = 90      p = 4      a = 1

Bestimme die Rückgabewerte von Ucase(zk), Lcase(zk), Len(zk), Chr(c), Chr(68), Mid(zk,1,1), Mid(zk,p,1), Mid(zk,5,a), Mid(zk,1,4).

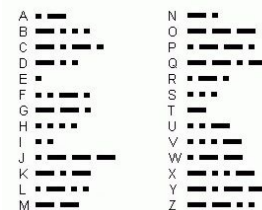
5. Das Programm aus Aufgabe 3 soll auch funktionieren, wenn der Anwender als Antwort "ja" oder "jA" oder andere Kombinationen von Groß- und Kleinbuchstaben benutzt. Verbessere es dementsprechend.

# Zeichenketten

6. Schreibe ein Programm, welches zu einer eingegebenen Zeichenkette (maximal 20 Zeichen)
- a) nur den ersten Buchstaben,
  - b) nur den letzten Buchstaben,
  - c) die ersten 3 Buchstaben,
  - d) die letzten 3 Buchstabe,
  - e\*) den ersten, den 3., den 5., usw. Buchstaben (Benutze hier eine Schleife!),
  - f) die einzelnen Zeichen der Reihe nach im Sekudentakt(!) (Benutze hier eine Schleife!),
  - g\*) die einzelnen Zeichen in umgekehrter Reihenfolge (Benutze hier eine Schleife!),
  - h) nur den 2. Bis 4. Buchstaben ausgibt.
- 7\*. Ein Text wird über das Terminal eingegeben, er soll dann in Sperrschrift ausgegeben werden; d. h. nach jedem Zeichen soll eine Leerzeichen stehen. Schreibe ein entsprechendes Programm.
- 7a. Ein Text wird über das Terminal eingegeben; bei der Ausgabe sollen alle Zeichen verdoppelt werden. Schreibe ein entsprechendes Programm.
8. Ein Text wird über das Terminal eingegeben, dieser Text soll rückwärts ausgegeben werden: Aus "Hallo Welt" soll also "tleW ollaH" werden. Schreibe ein entsprechendes Programm.

## Hausaufgabe

Schreibe ein Programm, welches ein eingegebenes Wort als Licht-Morsezeichen ausgibt.



### Hinweise:

- ☞ Du kannst Dich auf 8 verschiedene Zeichen beschränken.
- ☞ beachte, dass auch Pausen zwischen den Buchstaben eingelegt werden.