

## Anweisungen zur Stringbearbeitung im Modul Strings

Funktionen	Beschreibung, Beispiel
<p>P := Strings.Pos(zk1,zk,stelle)  zk1, zk: StringT  P, stelle: LONGINT</p>	<p>Stellt fest, ab welcher Position die Zeichenkette zk1 in zk vorkommt. Gesucht wird ab stelle (<math>\geq 1</math>).  P := Strings.Pos(„f“, „Erfurt“, 1);  P bekommt den Wert 3.  Falls die Suche nicht erfolgreich ist, hat P den Wert 0.</p>
<p>L := Strings.Length(zk)  zk: StringT  L: LONGINT</p>	<p>Ermittelt die Länge einer Zeichenkette zk.  L := Strings.Length(„Erfurt“);  L erhält den Wert 6.</p>
<p>Z := Strings.Val(zk)  zk: StringT  Z: LONGINT</p>	<p>Die Zeichenkette zk wird in eine Zahl Z umgewandelt.  Z := Strings.Val(„0815“)  Z hat danach den Wert 815.  Führende Nullen werden entsprechend bearbeitet.</p>
<p>E := Strings.ValResult(zk)  zk: StringT  E: LONGINT</p>	<p>Diese Funktion prüft, ob die Umwandlung der Zeichenkette zk in eine Zahl erfolgreich ist. Dabei wird im Ergebnis E eine Zahl festgelegt, die mit den Konstanten  Strings. ISSHORTINT  Strings. ISINTEGER  Strings. ISLONGINT  Strings. ISOUTOFRANGE  Strings. STRINGEMPTY  Strings. STRINGILLEGAL  verglichen werden kann.  E := Strings.ValResult(„0815“)  E erhält den Wert ISINTEGER (2).</p>

Prozedur	Beschreibung, Beispiel
Strings.Copy(quelle, ziel, pos, anz) quelle, ziel: StringT pos, anz: LONGINT	Kopiert aus der Zeichenkette quelle ab der Position pos die folgenden anz Zeichen in die Zeichenkette ziel. Strings.Copy(„Erfurt“,zk,1,2); zk speichert „Er“.
Strings.Delete(quelle, pos, anz) quelle: StringT pos, anz: LONGINT	Löscht in der Zeichenkette quelle ab der Position pos eine Anzahl anz von Zeichen. zk:=„Erfurt“; Strings.Delete(zk,1,2); zk speichert danach „furt“.
Strings.Append(zk, zk1) zk, zk1: StringT	Fügt an die Zeichenkette zk die Zeichenkette zk1 an. zk wird also um zk1 erweitert. zk:=„Er“; zk1:=„furt“; Strings.Append(zk, zk1); zk ist jetzt „Erfurt“.
Strings.Insert(zk1,zk,pos) zk, zk1: StringT pos: LONGINT	Fügt in die Zeichenkette zk die Zeichenkette zk1 an der Position pos ein. zk:=„Ert“; zk1:=„fur“; Strings.Insert(zk1, zk, 2); zk ist jetzt „Erfurt“.
Strings.InsertChar(z,zk,pos) z: CHAR zk: StringT pos: LONGINT	Analog Strings.Insert, bloß für ein einzelnes Zeichen z.
Strings.Str(zahl, zk) zahl: LONGINT zk: StringT	zahl wird in eine Zeichenkette zk umgewandelt. Ist die Zeichenkette zk nicht lang genug, besteht sie nur aus den Zeichen \$.
UpCase(zk) zk: StringT	Wandelt alle Kleinbuchstaben der Zeichenkette zk in Großbuchstaben um. Alle anderen Zeichen bleiben unverändert.

```

MODULE Text;
IMPORT In, Out, Strings;
VAR ort, vorort, gesamort: ARRAY 50 OF CHAR;
    plz: ARRAY 10 OF CHAR;
    code :INTEGER;
    stelle,leitzahl,laenge : LONGINT;
PROCEDURE ProgMain*;
BEGIN
    In.Prompt("Ort");In.String(ort);
    In.Prompt("PLZ");In.String(plz);
    laenge:= Strings.Length(ort); Out.Int(laenge,10); Out.Ln;
    stelle:= Strings.Pos('-',ort,1); Out.Int(stelle,10); Out.Ln;
    Strings.Copy(ort,vorort,stelle+1,laenge+1-stelle); Out.String(vorort);Out.Ln;
    Strings.Delete(ort,stelle,laenge+1-stelle); Out.String(ort); Out.Ln;
    leitzahl:=Strings.Val(plz); code:=Strings.ValResult(plz); Out.Int(code,5);
    Out.Ln;
    IF ((code > Strings.ISLONGINT) OR (Strings.Length(plz) # 5))
    THEN Out.String("Fehler in der Postleitzahl") END;
    Strings.Append(gesamort,plz); Out.String(gesamort); Out.Ln;
    Strings.Append(gesamort,ort); Out.String(gesamort); Out.Ln;
    Strings.Append(gesamort,vorort); Out.String(gesamort); Out.Ln;
    Strings.Insert(' ',gesamort,6); Out.String(gesamort); Out.Ln;
    Strings.InsertChar('-',gesamort,7+stelle); Out.String(gesamort); Out.Ln;
END ProgMain;
END Text.

```