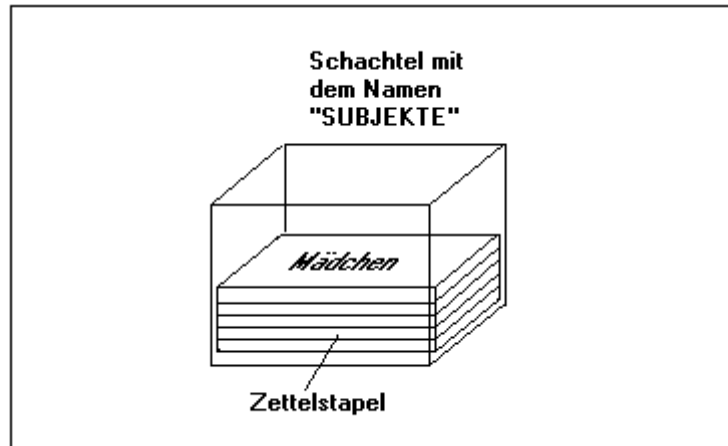


## Verwendung von Listen

Listen werden immer dann verwendet, wenn bestimmte Daten hintereinander zusammengefasst werden sollen. Mit Hilfe des Schachtelmodells lässt sich die Anweisung

SETZE "Subjekte" [ Mädchen Computer Katzen Tische ]

so darstellen:



LOGO legt eine neue Schachtel mit dem Namen "SUBJEKTE" an. In diese Schachtel wird diesmal nicht ein einziger Zettel hineingelegt, sondern ein Zettelstapel. Auf jedem der Zettel ist eines der Subjekte notiert.

In einer **Liste** können beliebig viele Elemente hintereinander gespeichert werden.

Es stellt sich als nächstes die Frage, wie LOGO an einzelne Zettel des Stapels herankommen kann. Die Anweisung

DRUCKEZEILE :Subjekte

druckt immer die vollständige Liste, also kein einzelnes Wort aus der Liste. Um einen Zettel aus dem Stapel hervorzuholen, verwendet man die ERSTES-Anweisung, die das erste Element aus der Liste zurückgibt.

Mit der Anweisung

DRUCKEZEILE **ERSTES** :Subjekte

erhält man das erste Element der Liste angezeigt. Es ist das Wort

Mädchen.

Mit der Anweisung

DRUCKEZEILE **LETZTES** :Subjekte

erhält man das letzte Element der Liste angezeigt. Es ist das Wort

Tische

Mit der Anweisung

DRUCKEZEILE **ELEMENT 3** :Subjekte

erhält man das 3. Element der Liste angezeigt. Es ist das Wort

Katzen

**Aufgabe 1:**

Lege eine Liste VERBEN mit den folgenden Verben an: tanzen rechnen schleichen stehen.  
Lasse das erste, das letzte und das zweite Element anzeigen.

**Anzeigen von Elemente aus einer Liste**

Das folgende LOGO-Programm zeigt alle Elemente einer Liste untereinander an.

```
PR Subjekte_anzeigen
  SETZE "Zähler" 1
  SOLANGE [:Zähler <= 4] [DZ ELEMENT :Zähler :Subjekte SETZE "Zähler" :Zähler+1]
  ENDE
```

Das folgende LOGO-Programm sucht das Wort „Katzen“ in der Liste Subjekte.

```
PR suche_Subjekt
  SETZE "Zähler" 1
  SOLANGE [:Zähler <= 4] [ WENN ( ELEMENT :Zähler :Subjekte ) ="Katzen" DANN
  ( DZ "gefunden" )
  SETZE "Zähler" :Zähler + 1 ]
  ENDE
```

Hinweis: Die runde Klammer vor WENN und nach „gefunden“ ist notwendig, damit LOGO dies als ein Befehl interpretiert.

**Aufgabe 2:**

- Schreibe ein Programm VERBEN\_ANZEIGEN, das alle Elemente der Liste Verben anzeigt.
- Schreibe ein Programm SUCHE\_SUBJEKT, das ein Wort aus der Liste Verben sucht und „gefunden“ anzeigt.
- Verändere das Programm SUCHE\_SUBJEKT so, dass ein beliebiges Wort gesucht wird.  
Also PR SUCHE\_SUBJEKT :Wort
- Verändere das Programm SUCHE\_SUBJEKT :Wort so, dass „nicht gefunden“ angezeigt wird, wenn das Wort nicht in der Liste ist.

Mit dem LOGO-Befehl ANZAHL kann man sich die Anzahl der Elemente einer Liste anzeigen lassen.

```
DZ ANZAHL :Subjekte           Anzeige: 4
```

**Aufgabe 3:**

Verändere das Programm SUCHE\_SUBJEKT :Wort so, dass in einer Liste mit beliebig vielen Elementen angezeigt wird, ob das Wort in der Liste ist oder nicht. Überprüfe das Programm mit der Liste Subjekte [ Mädchen Computer Katzen Tische Tiger Stühle].

**Aufgabe 4: Einkaufslisten**

In einer Familie wird festgelegt, wer für das Wochenende welche Einkäufe erledigen soll.

Jan -> Brot Brötchen Kuchen Brezel

Jochen -> Mehl Zucker Hefe Milch

Julia -> Bierschinken Hähnchenschenkel Emmentaler

Schreibe ein Programm WER\_KAUFT\_WAS :ARTIKEL.

Hinweis: Erzeuge Listen der Form: SETZE „Jan“ [ Brot Brötchen Kuchen Brezel ]

**Aufgabe 5: Erreichbare Städte**

Für Reisende ist es wichtig zu wissen, welche anderen Städte mit der Bahn oder dem Flugzeug von einer bestimmten Stadt aus direkt zu erreichen sind.

Hamburg -> Bremen Hannover Kopenhagen

Rom -> Mailand Palermo Florenz Kairo

- Notiere jeweils 5 Städte, die von Groß-Gerau und Mainz aus direkt zu erreichen sind.
- Schreibe ein Programm VON\_WELCHER\_STADT\_ERREICHT\_MAN :STADT.

**Aufgabe 6: Bundesligapaarungen**

Rechts sind die Spielpaarungen des 13. Bundesligaspieltages 2017/18 abgebildet. Schreibe ein Programm WER\_SPIELT\_GEGEN :MANNSCHAFT

Verwende dazu Listen. Achte darauf, dass der Name der Mannschaft nur ein Wort sein darf, also HERTHA\_BSC

 Hannover	-	 Stuttgart (Fr.)
 Dortmund	-	 Schalke (Sa.)
 Leipzig	-	 Bremen
 Freiburg	-	 Mainz
 Frankfurt	-	 Leverkusen
 Augsburg	-	 Wolfsburg
 M'gladbach	-	 Bayern
 Hamburg	-	 Hoffenheim (So.)
 Köln	-	 Hertha BSC

**Listen in Listen**

*Setze "Spieltag" [ [ Hannover Stuttgart ] [ Dortmund Schalke ] ]*

In der Liste „Spieltag“ befinden sich weitere Listen für jede Paarung des Spieltags aus Aufgabe 6.

```
PR SUCHE_GEGNER :TEAM
Setze "Spieltag" [ [ Hannover Stuttgart ] [ Dortmund Schalke ] ]
SETZE "Zähler" 1
SETZE "gefunden" falsch
SOLANGE [ :zähler <= ANZAHL :spieltag ] [
  SETZE "Paarung" ELEMENT :zähler :spieltag
  WENN (ERSTES :paarung) = :team DANN ( SETZE "gefunden" WAHR SETZE "Gegner" LE :paarung )
  WENN ( LE :paarung) = :team DANN ( SETZE "gefunden" WAHR SETZE "Gegner" ERSTES :paarung )
  SETZE "Zähler" :Zähler + 1 ]
  WENN :gefunden= WAHR DANN DZ :Gegner SONST DZ "Nicht gefunden."
ENDE
```

Wichtiger Hinweis; Die runden Klammern hinter DANN dürfen auf keinen Fall weggelassen werden. !!!!!

**Aufgabe 7:**

Beschreibe das Programm SUCHE\_GEGNER Zeile für Zeile in einem WORD-Dokument.

**Aufgabe 8:**

Ergänze die fehlenden Paarungen und teste das Programm.

**Aufgabe 9: Musik**

Das untenstehende Lied hast du im letzten Schuljahr programmiert. Jetzt sollst du in eine Liste NOTEN weitere Listen einfügen, in denen jeweils die Höhe und die Länge eines Tons stehen. Schreibe ein Programm SPIELE\_LIED entsprechend dem Programm SUCHE\_GEGNER, das mit dieser Liste das Lied spielt.

Hinweis: Hier werden nicht alle Zeilen aus dem Programm SUCHE\_GEGNER benötigt. So können die beiden WENN – Zeilen z.B. weggelassen werden. Der LOGO Befehl TON :HÖHE :LÄNGE spielt einen Ton

**Lied 1**

The image shows a musical score titled 'Lied 1' with five staves. The notation includes various note values (quarter, eighth, and sixteenth notes), rests, and slurs. The first staff starts with a treble clef and a key signature of one flat (B-flat). The music is written in a simple, melodic style.

**Aufgabe 10: Zeichnen**

Mit dem LOGO-Befehl AUFXY :X :Y kann man den Igel auf eine bestimmte Position der Grafikseite schicken. Der Punkt links unten hat die Koordinaten (0;0). Die Koordinaten einer Figur können wiederum als Listen in eine Liste eingefügt werden. Schreibe ein Programm ZEICHNE entsprechend dem Programm SPIELE\_LIED, das mit der Liste der Punkte ein Quadrat der Seitenlänge 100 zeichnet.

**Aufgabe 11: Eine eigene Figur zeichnen**

Erweitere die Liste der Punkte aus Aufgabe 10 und verwende das Programm aus Aufgabe 10, um eine eigene Figur zu zeichnen.

**Aufgabe 12: Daten**

Hans ist 15 Jahre alt und 170 groß. Sabine ist 15 Jahre alt und 160 groß. Diese Daten können wie folgt in einer Liste gespeichert werden:

```
SETZE „DATEN“ [ [Hans 15 170 ] [ Sabine 15 160 ] ]
```

Ergänze die Liste mit weiteren Daten von deinen Mitschülern. Schreibe die LOGO Programme ZEIGE\_DATEN\_VON :NAME, WER\_IST\_WIE\_ALT :JAHRE WER\_IST\_WIE\_GROSS :GRÖSSE. Die Programme sollen eine ähnliche Struktur haben wie das Programm SUCHE\_GEGNER :TEAM.

**Aufgabe 13: Notenspiegel (\*)**

In der Liste „Noten“ sind die Ergebnisse einer Klassenarbeit gespeichert: [ 2 3 1 6 3 4 3 4 3 4 2 1 3 2 4 5 2 4 5 1 2 4 3 5 6 3 4 2 1 4 5 3 4 2 2 4 1 2 4 5 3 3 4 2 3 ]

Erstelle ein Programm NOTENSPIEGEL, das in der Liste „Notenspiegel“ den Notenspiegel der Arbeit sowie die Durchschnittsnote speichert. Zerlege in Teilprogramme. Verwende auch Funktionen.

Beispiel für die Liste „Notenspiegel“ [ [ 1 5 ] [ 2 10 ] ..... [ 3,1234 ] ]

Verwende auch die Befehle zum Einfügen von Elementen in eine Liste:

MITERSTEM <Ausdruck> <Liste>

oder MITLETZTEM <Ausdruck> <Liste>

**Aufgabe 14: Vokabeltrainer**

In der Liste Vokabeln sind die Vokabeln und deren Übersetzung gespeichert: [ [to\_go gehen] [to\_see sehen ] ....]

Füge noch weitere Vokabeln in die Liste ein. Erstelle ein Programm VOKABELTRAINER, das zufällig eine Teilliste auswählt und das darin enthaltene englische Wort anzeigt. Der Benutzer soll zur Eingabe der deutschen Übersetzung aufgefordert werden. Wird das richtige Wort eingegeben, so erscheint richtig, sonst erscheint falsch. Es soll 3mal nachgefragt werden.

Hinweis: Dieser Befehl gibt eine Zufallszahl zwischen 1 und der Anzahl der Listenelemente zurück: 1+ zufallszahl anzahl :vokabeln

**Aufgabe 15: Sortieren (\*)**

Erstelle ein Programm SORTIEREN, das die Klassenarbeitsnoten nach der Größe sortiert. Nach der Programmausführung stehen in der Liste zuerst alle 1er, dann die 2er, usw.

Zerlege in Teilprogramme. Verwende auch Funktionen.

**Eigenschaftslisten**

Mit Eigenschaftsliste kann man Objekten bestimmte Eigenschaften zuordnen.

MITEG "Suse" "Alter" 7

Mit diesem Befehl weist man "Suse" als Eigenschaft das "Alter" 7 zu. Oder anders ausgedrückt: In die Eigenschaftsliste mit dem Namen "Suse" wird die Eigenschaft "Alter" und der zugehörige Wert 7 eingefügt.

Mit der Anweisung:

DZ GIBEG "Suse" "Alter"

erhält man das Alter von Suse ausgedrückt.

**Aufgabe 16:**

Erstelle ein Programm, das Suse die folgenden Eigenschaften zuordnet: Alter 4, Haarfarbe braun, Schwester Lisa. Weiterhin soll Lisa die folgenden Eigenschaften bekommen: Alter 4, Haarfarbe blond, Schwester Suse.

Das Programm soll am Ende die folgenden Befehlszeilen haben:

DZ EGLISTE "Suse"

ZEIGEEGLISTEN

DZ GIBEG "Lisa" "Haarfarbe"

DZ GIBEG GIBEG "Suse" "Schwester" "Alter"

WENN ( ISTEGL? "Lisa" "Klasse" ) =FALSCH DANN DZ „Eigenschaft nicht gefunden“  
Erkläre, was jeweils auf dem Bildschirm angezeigt wird. Begründe.

### Aufgabe 17

Wähle einen Bundesligaverein aus und erfasse für die Namen der Spieler die einzelnen Positionen in Eigenschaftslisten. Z.B. MITEG "MAINZ\_05" "Zentner"."Tormann" Der Tormann von Mainz 05 heißt „Zentner“.

Informationen über die Aufstellung des letzten Spieltags erhältst du z.B. bei [www.kicker.de](http://www.kicker.de). Wähle „Stürmer1“ „Stürmer2“ usw. falls es mehrere Positionen mit der gleichen Bezeichnung gibt. Erstelle ein Programm WO SPIELT? :Name, das die Position des eingegebenen Spielers anzeigt.

### Objekte

In der Informatik besteht ein Objekt aus bestimmten Daten (Eigenschaften) und ein Objekt besitzt Methoden, das Objekt tut etwas.

Das Objekt PERSON hat z.B. einen Namen, ein Alter und eine Größe. Das Objekt Person kann z.B. „grüßen“.

WIN-LOGO ist zwar keine Objektorientierte Programmiersprache. Eigenschaften wie Vererbung oder Kapselung sind beispielsweise nicht vorhanden. Trotzdem kann man sich das Prinzip der Objekte mit WIN-LOGO verdeutlichen.

Die Methode DEFINIERE des Objekts PERSON belegt die Eigenschaften Name, Alter und Größe mit Werten. Als Trennzeichen zwischen dem Namen des Objekts und der Methode verwendet man in der Regel einen Punkt.

```
PR person.definiere :Name :Alter :Größe
  miteg "Person" "Name" :name
  miteg "Person" "Alter" :alter
  miteg "Person" "Größe" :größe
ENDE
```

Die Methode GRÜSSE des Objekts PERSON schreibt einen Gruß auf die Textseite.

```
PR person.grüsse
  dr "Hallo, ich bin " DZ GIBEG "Person" "Name"
ENDE
```

### Aufgabe 18:

Programmiere die beiden Methoden DEFINIERE und GRÜSSE. Belege die Attribute des Objekts PERSON mit Deinen Daten und lasse den Gruß auf der Textseite erscheinen.

### Aufgabe 19:

Ein Ton hat eine Höhe und eine Länge. Ein Ton wird „gespielt“. Programmiere die Methoden DEFINIERE und SPIELE des Objekts TON. „Spiele“ damit einen Ton.

**Aufgabe 20:**

Ein Quadrat hat eine X- und eine Y-Koordinate (links unten) sowie eine Seitenlänge. Ein Quadrat wird gezeichnet. Programmieren Sie die Methoden DEFINIERE und ZEICHNE des Objekts QUADRAT und zeichnen damit ein Quadrat auf der Grafikseite.

**Aufgabe 21:**

Das Quadrat aus der Aufgabe 21 soll sich auf dem Bildschirm bewegen. Dazu werden die folgenden zusätzlichen Methoden des Objekts QUADRAT benötigt: SETZE\_XY verändert die X- und Y-Koordinate des Objekts QUADRAT. LÖSCHE löscht das Quadrat an der aktuellen Position. FARBE 255 255 255 setzt die Zeichenfarbe auf weiß, FARBE 0 0 0 setzt die Zeichenfarbe auf schwarz. Die Methode BEWEGE des Objekts QUADRAT soll das Quadrat auf der Grafikseite von links unten nach rechts oben wandern lassen.

**Aufgabe 22:**

Ein roter Ball soll waagrecht auf der Grafikseite rollen. Verwenden Sie Objekte und Methoden wie bei Aufgabe 21. Verwenden Sie den LOGO-Befehl FÜLLE.