



Listen

Ihr habt gerade eine neue Möglichkeit kennen gelernt, wie man in der Informatik Daten speichern kann.

Diese neue Datenstruktur, in der viele gleiche Daten nacheinander gespeichert sind und über ihre Position angesprochen werden können, nennen wir (indizierte) **Liste**. (Indiziert bedeutet soviel wie durchnummeriert.)

Wir geben unserer Liste einen Namen und können dann einzelne Elemente der Liste über den Namen und deren Position ansprechen. Die Position schreibt man dazu in eckige Klammern.



Schreibweisen:

Sprungweiten = [56, 82, 71, 90, 44] beschreibt die gesamte Liste namens Sprungweiten.

Sprungweiten [1] beschreibt das Listenelement an Position 1, also im Beispiel 56.

Sprungweiten [2] liefert 82, usw.

Aufträge:

1. Schreibt in eurer Gruppe weitere Listen mit der richtigen Schreibweise auf (z.B. eure Schuhgrößen, eure Körpergrößen, eure Entfernung von zu Hause zur Schule, eure Noten in der letzten IMP-Arbeit, usw.).
2. Stellt euch gegenseitig Fragen, wie z.B. „Wie lautet das Element Schuhgrößen[3]?“.
3. Bestimmt von zwei der Listen das Maximum und den Mittelwert. Geht dabei wieder so vor, wie bei den Froschsprungweiten und sprecht über euer Vorgehen oder schreibt wieder ein Protokoll dazu.

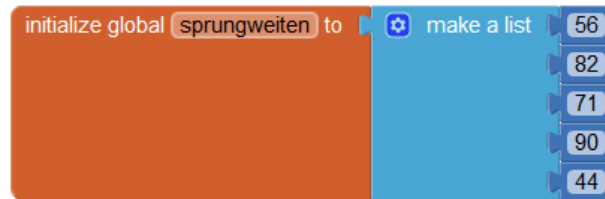
1. Individuelle Lösungen. Beispiel: Schuhgrößen = [36, 41, 39, 36, 38]
2. Individuelle Lösungen. Beispiel: Schuhgrößen[3] = 39
3. Das Vorgehen finden Sie im Protokoll zu den Froschsprungweiten. Hier geht es analog.

Bild: Eigenes



Listen im MIT-App-Inventor

Definition der Liste:

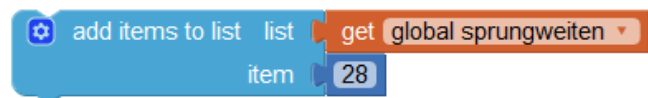


Ansprechen eines Elementes

(z.B. Sprungweiten [2]):



Hinzufügen eines Elementes am Ende der Liste (z.B. Sprungweiten [6] = 28):



Ersetzen eines Wertes

(z.B. Sprungweiten [3] = 73):



Aufträge:

4. Definiert im AI2 eine eurer Listen aus Auftrag 1.
5. Setzt auf den Screen eine ListView (z.B. IvSchuhgroessen) und einen Button. Beim Klick auf den Button sollen die Elemente der Liste der ListView übergeben werden. Programmiert das Ereignis.
6. Lasst beim Klick auf einen weiteren Button auf einem Label ein bestimmtes Element der Liste ausgeben.
7. Lasst beim Klick auf einen weiteren Button ein Element an die Liste anhängen und ein anderes ersetzen. Vergesst nicht, am Ende die veränderte Liste wieder der ListView zu übergeben.

Scratch wird von der Lifelong-Kindergarten-Group am MIT-Media-Lab entwickelt. Siehe <http://scratch.mit.edu>.
Scratch ist lizenziert unter [CC BY-SA 2.0](https://creativecommons.org/licenses/by-sa/2.0/deed.en) (<https://creativecommons.org/licenses/by-sa/2.0/deed.en>).



Mögliche Lösungen:

4. Beispiel Schuhgrößen:

```

initialize global schugroessen to
    make a list
        41
        36
        38
        37
        41
    
```

5. Screen:

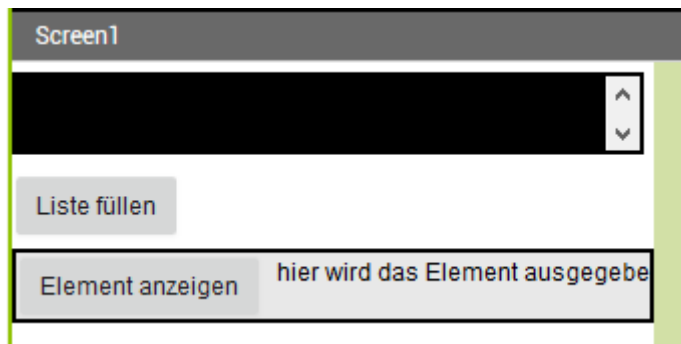


Code:

```

when btListeFuellen .Click
do set lvSchuhgroessen .Elements to get global schugroessen
    
```

6. Ergänzen des Screens um einen Button und ein Label (hier in einem horizontalen Layout):



Code:

```

when btElement .Click
do set lbElement .Text to select list item list
    index 3
    get global schugroessen
    
```



7. Ergänzen des Screens um einen weiteren Button *btAufgabe7*; Code:

```

when btAufgabe7 .Click
do
  add items to list list
  item 39
  replace list item list
  index 2
  replacement 37
  set lvSchuhgroessen . Elements to
  get global schugroessen
  
```

Der MIT App Inventor (<http://appinventor.mit.edu>) wurde ursprünglich von einem Entwicklerteam um Mark Friedman und Hal Abelson bei Google entwickelt und 2012 an das MIT übergeben.
 Der MIT App Inventor wird unter der Creative Commons Attribution-ShareAlike 3.0 Unported License veröffentlicht:
<https://creativecommons.org/licenses/by-sa/3.0>