


## Listen in Snap!<sup>1</sup> – Didaktische Anmerkungen

### Scratch – Snap!

Für das algorithmische Problemlösen in der Sekundarstufe 1 reicht Scratch als Werkzeug aus. Selbst für fortgeschrittene Schüler:innen lassen sich vielfältige und algorithmisch anspruchsvollere Fragestellungen finden, die in Scratch umgesetzt werden können. Dabei können sogar schon eigene Blöcke (ohne Rückgabe) implementiert werden und auch die Verwendung einfacher Listen ist in Scratch problemlos möglich.

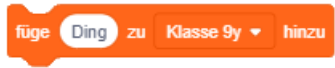

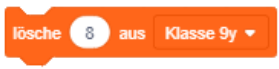

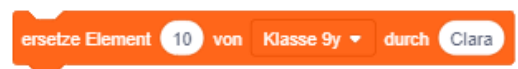

Ganz ähnlich zu Scratch, aber mit deutlich mehr Möglichkeiten, ist die grafische Programmierumgebung Snap! (zu finden unter <https://snap.berkeley.edu/snap/snap.html>). Hat man bereits Erfahrungen beim Implementieren in Scratch, fällt der Umstieg auf Snap! erfahrungsgemäß nicht schwer. Anders als in Scratch können in Snap! beispielsweise auch eigenen Blöcke mit Rückgaben implementiert werden. Zusammen mit den Snap!-Bibliotheken, die bei Bedarf zu einem Projekt dazu geladen werden können, ist Snap! an vielen Stellen deutlich mächtiger als Scratch und wird als Werkzeug sowohl in der Sekundarstufe II als auch in Universitäten eingesetzt. Wir gehen an dieser Stelle nur auf ausgewählte Aspekte zu Listen ein, die für interessierte Schüler:innen auch im Unterricht in der Sekundarstufe I relevant sein können. Die folgenden Beschreibungen richten sich daher an Lehrkräfte. Bei Bedarf können Sie im Unterricht ausgewählte Aspekte davon interessierten Schüler:innen erklären.

### Mehrere Variablen gleichzeitig verwalten

In Snap! können auch mehrere Variablen oder Werte gleichzeitig in einer Liste gespeichert bzw. verwaltet werden. Anders als in Scratch 3.0 gibt es jedoch keinen „neue Liste“ – Befehl: eine neue Liste kann entweder anonym über den Block  genutzt oder über eine entsprechende

Variable erzeugt werden, der dann eine Liste zugewiesen wird. Beispiel: 

Für Operationen auf Listen gelten im Prinzip die gleichen grundsätzlichen Überlegungen zu Listen wie bei Scratch. So gibt es in Snap! jeweils entsprechende Blöcke:

Scratch 3	Snap!
	
	
	

In die Stelle des Listensymbols  muss dann jeweils die gewünschte Variable eingesetzt werden.

<sup>1</sup> Snap! wird von der University of California, Berkeley zur Verfügung gestellt: <https://snap.berkeley.edu>

## Importieren von Listen

In Snap! können csv-Dokumente als Listen importiert werden. Die einfachste Variante hierzu ist, das gewünschte Dokument in Snap! hineinzuziehen. Dann wird automatisch eine Variable mit dem Namen der csv-Datei angelegt und die zugehörige Liste sowohl auf der Bühne als auch als Dialogfenster angezeigt. Probieren Sie es mit der Datei `Sortiment_Bsp.csv` aus.

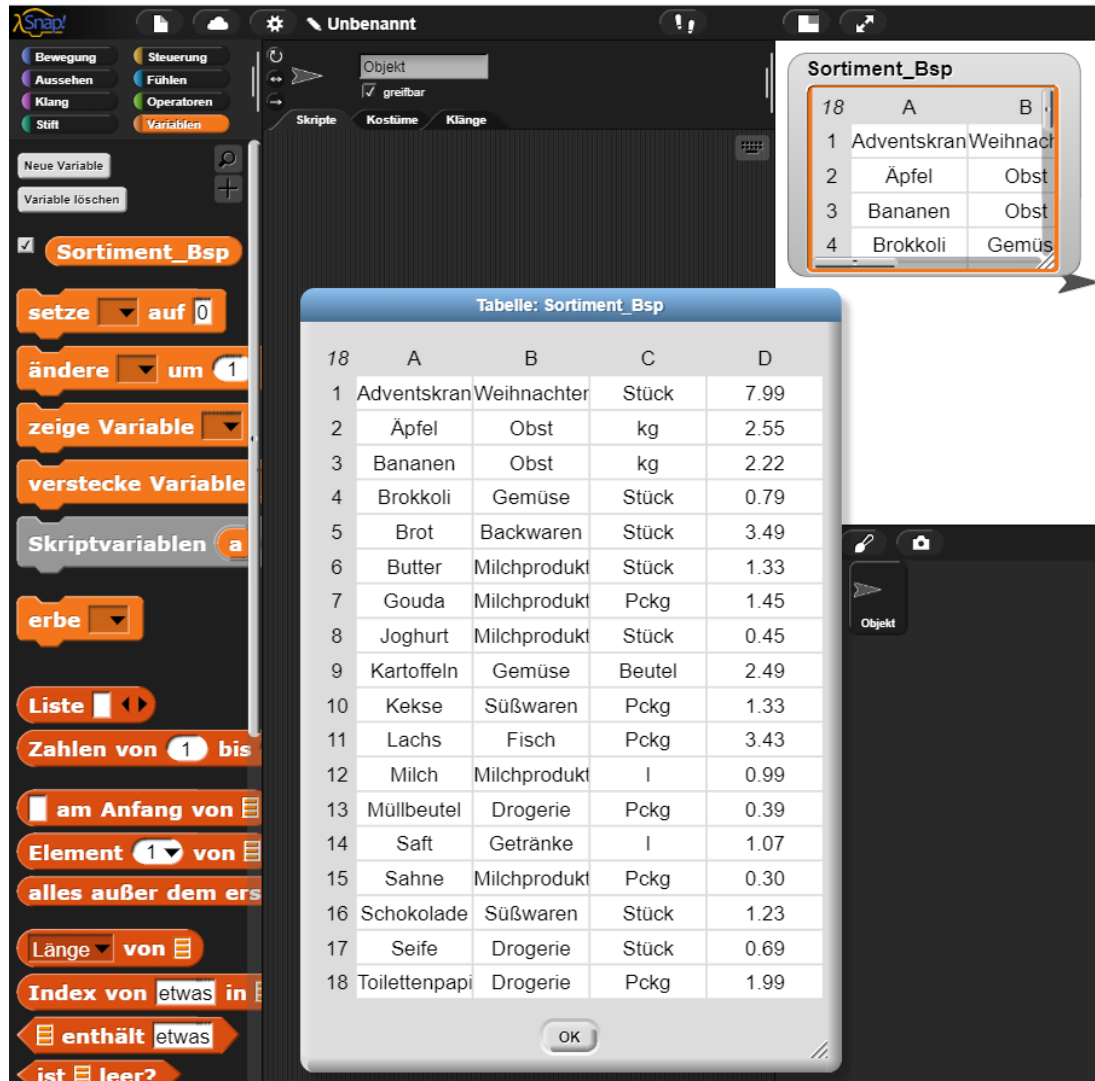


Abbildung 1: Importieren von Tabellen

## Mehrdimensionale Listen

In Abbildung 1 ist zu erkennen, dass in Snap! auch mehrdimensionale Listen verwendet werden können. Im Beispiel ist eine zweidimensionale Liste, also eine Liste von Listen zu erkennen. Noch klarer wird dies, wenn man durch Rechtsklick in die Tabelle die `Listenansicht` wählt (vgl. Abbildung 2):

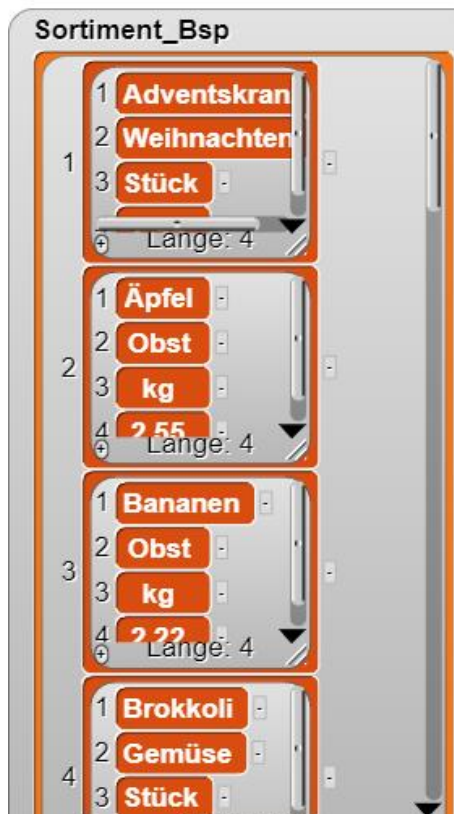


Abbildung 2: Listenansicht der gleichen Liste wie in Abbildung 1

Zur besseren Übersicht und auch zur schnelleren Analyse der angezeigten Daten empfiehlt sich bei zweidimensionalen Listen die tabellarische Darstellung, die beim Importieren auch standardmäßig wie in Abbildung 1 angezeigt wird.

### Zugriff auf Zeilen, Spalten und Zellen einer zweidimensionalen Liste

Beispiel: Zeile 7 der Tabelle:



Eine Tabelle ist eine Liste von Listen. Jede Zeile ist also ein Listeneintrag, d.h. die 7. Zeile ist der 7. Listeneintrag der Tabelle Sortiment\_Bsp.

### Beispiel: Spalte 2 der Tabelle:

Etwas ungewöhnlicher erhält man eine Spalte einer Tabelle mit dem Befehl



Abbildung 3: Spalte 2 der Tabelle Sortiment\_Bsp


Mit dem Block **wende an auf** können Listen auf vielfältige Weise manipuliert werden. Dazu wird der grau umringte Block auf jedes Element der in  eingefügten Liste angewandt und das jeweilige Ergebnis in eine neue Liste geschrieben. Dabei können die einzelnen Elemente der Eingabeliste einfache Werte wie Abbildung 4, oder auch wieder Listen wie in Abbildung 3 oder Abbildung 5 sein. Beispiele:



Abbildung 4: wende-an-Block angewendet auf 1-dimensionale Liste



Abbildung 5: wende-an-Block angewendet auf 2-dimensionale Liste

## Unterrichtsideen

Wie eingangs erwähnt, werden Listen in Snap! in der Sekundarstufe 1 eher für interessiertere Schüler:innen thematisiert und dann wahrscheinlich für Fragestellungen genutzt, die von den Lernenden selbst formuliert wurden. In der Einführungsphase oder als herausforderndere Aufgaben in der Sekundarstufe 1 sind zudem folgende Unterrichtsideen denkbar:

- *Ein Musikstück abspielen:* Ein Lied ist letztendlich nichts anderes als eine Liste von Tönen. Diese werden in Snap! durch Zahlen dargestellt, wie man am folgenden Block erkennt:



Eine mögliche Programmieraufgabe wäre also, ein als Liste gegebenes Musikstück abspielen zu lassen. Eine mögliche Vorlage hierzu finden Sie in der Datei

Vorlage\_Song\_eindimensional.xml. Als Erweiterung wäre denkbar, in einer zweidimensionalen Liste nicht nur die einzelnen Töne, sondern auch ihre jeweilige Länge (als Anzahl der Schläge) zu speichern und das zugehörige Lied abspielen zu lassen. Ein Beispiellied finden Sie als zweidimensionale Liste in der csv-Datei Song2.csv.

Mögliche Lösungen finden Sie unter

Song\_eindimensional.xml bzw.

Song\_zweidimensional.xml.

- *Geschenkpapier entwerfen:* In Snap! können in einer Liste sogar eigene Blöcke gespeichert werden. Ein Beispiel wird in Abbildung 6 dargestellt: in der Variable Malen ist eine Liste von möglichen Skripten zur Erzeugung eines zufälligen Musters auf der Bühne gespeichert. Auf diese Art können Algorithmen implementiert werden, die zufällig Skripte aus der Liste Malen auswählen und damit ein Muster auf der Bühne erzeugen.

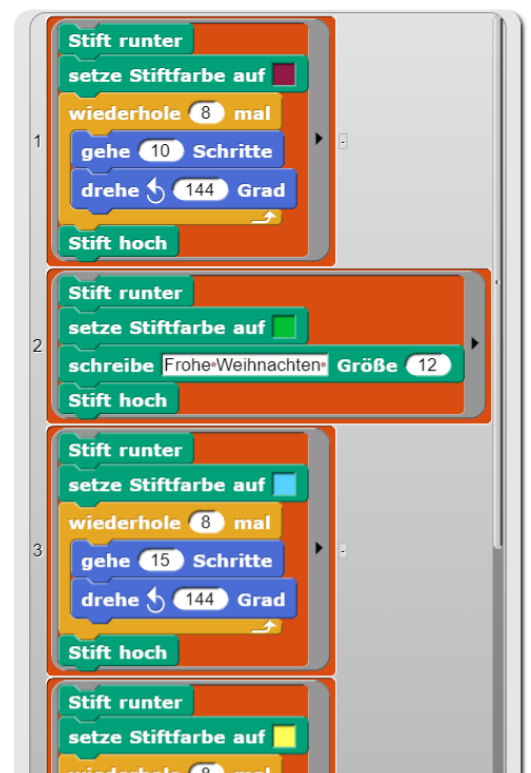


Abbildung 6: Liste von Skripten zur Erzeugung eines zufälligen Musters

- *Daten analysieren:* Es gibt im Internet viele zuverlässige Daten in Form von CSV-Dokumenten, die sich zur Analyse in Snap! eignen. So findet man beispielsweise unter [https://www.niedersachsen.de/Coronavirus/aktuelle\\_lage\\_in\\_niedersachsen/](https://www.niedersachsen.de/Coronavirus/aktuelle_lage_in_niedersachsen/) (Link vom 01.03.2021) die tagesaktuellen Zahlen zu COVID-19 in Niedersachsen. Mögliche Fragen, die man in diesem Zusammenhang algorithmisch unter Auswertung der Tabelle beantworten kann sind etwa die nach der höchsten 7-Tagesinzidenz pro 100.000 Einwohner oder die einer grafischen Darstellung der 7-Tagesinzidenzen über den gesamten Messzeitraum. Mögliche Lösungen hierzu finden Sie im Dokument `Analyse_Corona_01-03-21.xml`.

Eine Vielzahl weiterer Tabellen findet man beispielsweise in der Datenbank des statistischen Bundesamtes: <https://www-genesis.destatis.de/genesis/online> (Link vom 02.03.2021). Hier können verschiedene Parameter ausgewählt und entsprechende csv-Dokumente erzeugt und heruntergeladen werden.

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht kommerziell - Keine Bearbeitungen 4.0 International Lizenz](#). Sie erlaubt Download und Weiterverteilung des vollständigen Werkes unter Nennung meines Namens, jedoch keinerlei Bearbeitung oder kommerzielle Nutzung.

Alle Abbildungen von Scratch-Bausteinen und -Objekten sind lizenziert unter einer [Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](#).

Scratch wurde entwickelt von der Lifelong Kindergarten Group, MIT Media Lab,  
<http://scratch.mit.edu>