Practice 12: What is a list?

Introduction to data structures and the concept of index
Module 4: Composite Data

From a single drawer to a full cabinet
So far, a **variable** was like a box where we could only store one piece of data. But what if we want to store the names of 30 students? We would need 30 variables! To solve this, we use **Lists**: an ordered set of data stored under a single name.

Key Concepts

1. The Cabinet Analogy

Imagine that a list is a **storage cabinet**.

- **List Name:** The name of the cabinet (e.g., `Students`).
- **Index:** The drawer number (1, 2, 3...). In Snap!, counting always starts at 1.
- **Item:** What we store inside each drawer (a name, a grade...).

2. Basic Operations

Unlike a normal variable, in lists we use specific blocks from the red category:

- **Add:** Puts a new piece of data at the end of the list.
- **Item:** Retrieves the content of a specific drawer without deleting it.

THE CHALLENGE: My First List

Let's create a small manager where you can add students and check who is the first in class.
Instructions in Snap!:

1. Create a variable named `Students`.
2. When the green flag is clicked, use the `set [Students] to (empty list)` block.
3. Program the **A** key:
   - It should `ask` for a name.
   - It should `add (answer) to (Students)`.
4. Program the **Space** key:

- The sprite should say: “The first student in the list is...” followed by `item (1) of (Students)`.

Solution Pseudocode

```
Algorithm ListExample
   Define Students As List
   Students <- NewEmptyList()

   // On 'A' key press
   Print "Student name?"
   Read name
   Add name TO Students

   // On 'Space' key press
   Print "The first one is: ", Students[1]
EndAlgorithm
```

This document is published under license
Creative Commons Attribution 4.0 International (CC BY 4.0)