



Prática 14: Percorrendo Listas

O ciclo "Para cada item" e o processamento de dados
Módulo 4: Dados Compostos

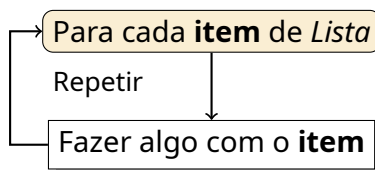
Como ler o armário todo?

Imagina que tens uma lista de 100 preços e queres saber quanto somam todos. Não vais somar $\text{item 1} + \text{item 2} \dots$ à mão! Para isso usamos os ciclos. Percorrer uma lista significa visitar uma a uma todas as suas gavetas automaticamente.

Conceitos Chave

1. O bloco "Para cada item" (For Each)

Este é o bloco mais potente para trabalhar com listas no Snap!. O programa encarrega-se de saber quantos elementos há e quando chega ao final:



- **A Variável "Item"**: É uma variável "temporária". Na primeira volta vale o conteúdo da gaveta 1, na segunda o da gaveta 2, e assim sucessivamente.

2. Esquema Visual de um Percurso

Imagina uma lista de frutas: [maçã, pera, banana].

| Volta (Iteração) | Valor de "item" | Ação do programa |
|------------------|-----------------|-------------------------|
| 1ª | maçã | Diz "maçã" por 2 seg. |
| 2ª | pera | Diz "pera" por 2 seg. |
| 3ª | banana | Diz "banana" por 2 seg. |

O DESAFIO: Chamada de Alunos

Vamos programar uma personagem que atue como um professor a fazer a chamada. Deve saudar cada aluno que esteja guardado na nossa lista.

Instruções no Snap!:


1. Cria uma lista chamada Turma e adiciona 4 ou 5 nomes (ex: Ana, Luís, Marta, Braís).
2. Usa o bloco `para cada (item) de (Turma)`.
3. Dentro do ciclo, a personagem deve `dizer` uma saudação.
4. **Truque:** Usa o bloco `junta [Olá] (item)` para que a saudação inclua o nome do aluno.

Pseudocódigo: O algoritmo de somar

```
Algoritmo SomarPreços
  Lista Preços <- [10, 5, 20]
  Variável Total <- 0

  Para Cada p em Preços:
    Total <- Total + p // Acumulamos o valor de cada gaveta

  Escrever "A conta total é: ", Total
FinAlgoritmo
```

Este documento é publicado sob licença
 Creative Commons Atribuição 4.0 Internacional (CC BY 4.0)