



INTRODUCTION TO VISUAL AND INTERACTIVE PROGRAMMING

CT803-4-0-OIVIP

Topic: List

Topic Learning Outcomes

At the end of this topic, you should be able to:

- Identify the use of List in programming

Contents & Structure

- List
 - Add
 - Insert
 - Delete
 - Accessing the list item

Introduction

- Variables allowed to store a single piece of data that can use in programming.



List



Lists are collections of data items that can be used in visual programming.



It can be used to store multiple values at once.



It can store anything from numbers and text to entire objects or other lists.



Lists are an essential tool for managing data and can add complexity and interactivity to the projects.



It consists of numbers paired with different items and each item can be retrieved using its respective paired number.

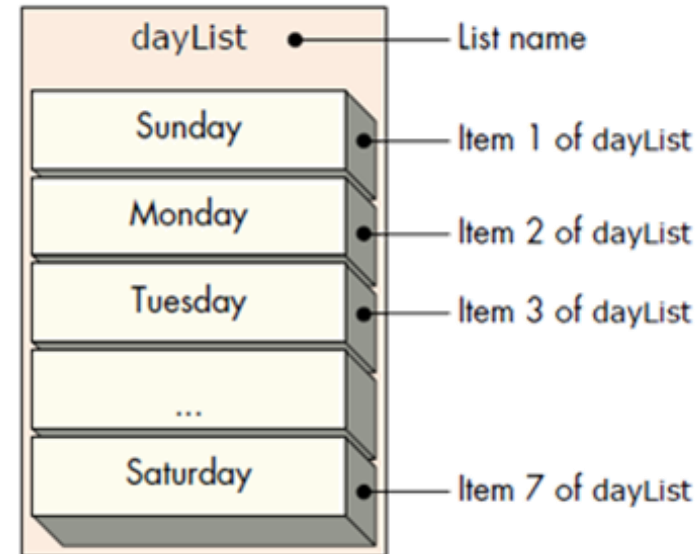


It can also be defined as a variable containing/storing multiple variables at once.

Lists

- A list is like a container where you can store and access multiple values.

Figure: A list that contains the days of the week



Variables or Lists



Variables are great for individual pieces of information like scores, lives and health.



However, sometimes we want to store a list of information.

For this, you need to use a LIST



List is a special variable type that can contain multiple other variables.



In other programming languages this is usually known as an Array.

Creating a List in Snap!

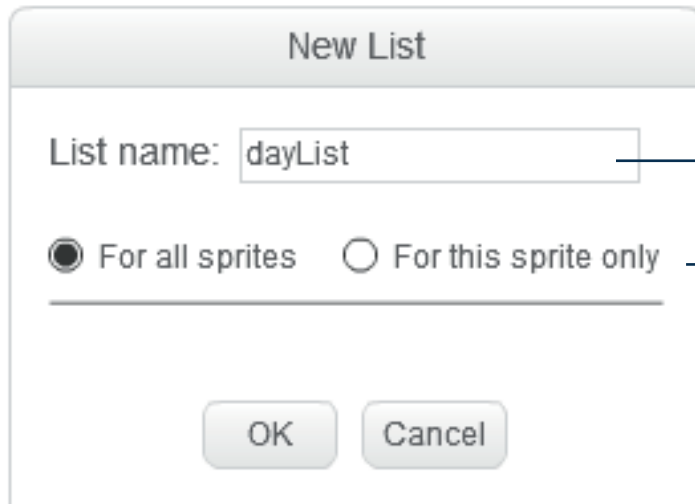
- To create a list in Snap!, just declare a variable.
- Example, your variable name is practiceList.
- At the beginning of a program, you need to make sure that the list is empty.
- Use the set to block on the Variables menu to make practiceList equal to an empty list.
- To get an empty list, click on the left black triangle to remove the blank item.



Creating List in BYOB (Offline Snap!)



Click to create and name a new list. When you create a list for the first time, the list blocks will appear. You can choose whether the list is for all sprites (global) or just for one sprite (local).



The 'New List' dialog box contains the following elements:

- Title: New List
- List name: dayList
- Scope: For all sprites, For this sprite only
- Buttons: OK, Cancel

Name

Scope



dayList

List name. Use the checkbox to show or hide the list's monitor on the Stage.

LIST - Example

$L1 = [45, 56, 67, 20]$



LIST – Example of empty list

L3 = []



Adding Item to List

add thing to dayList

Add a new item to the end of the list.

insert thing at 1 of dayList

Insert an item at the specified index



Insert a new item to manually into the list

Delete Item from List



Remove the item at the specified index

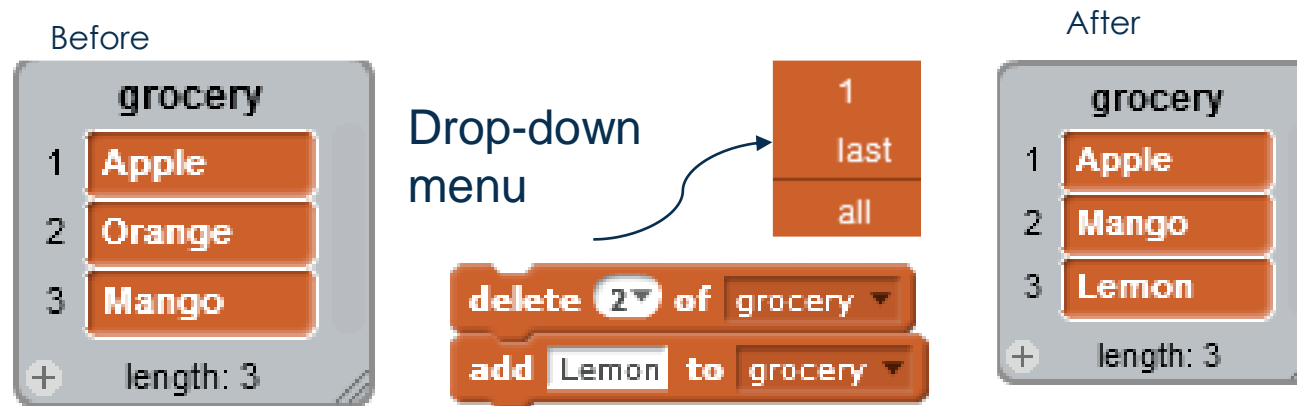


Remove the item manually from the list

List - Example

- **Add and Delete**

- The **add** command places a new item at the end of a list, while the **delete** command removes an item from a specific position.



The **add** command is straightforward, but let's examine the **delete** command more closely. You can type the index of the element you want to delete directly into the block's parameter slot, or you can click the dropdown arrow.

<#>

Replace Item from List

```
replace item 1 of dayList with thing
```

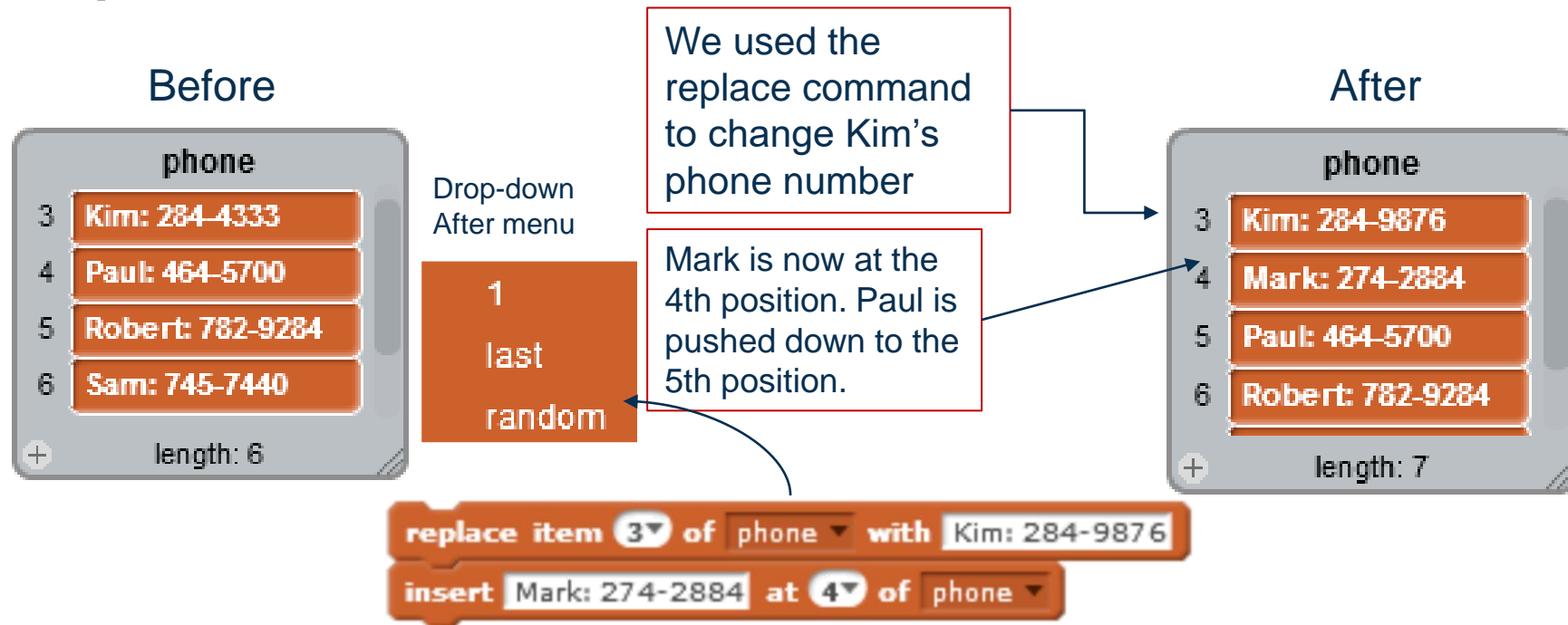
Replace the item at the specified index



Replace the item manually from the list

List - Example

- Insert and Replace



Other Lists block

item 1 of dayList

Return the item at the specified index

length of dayList

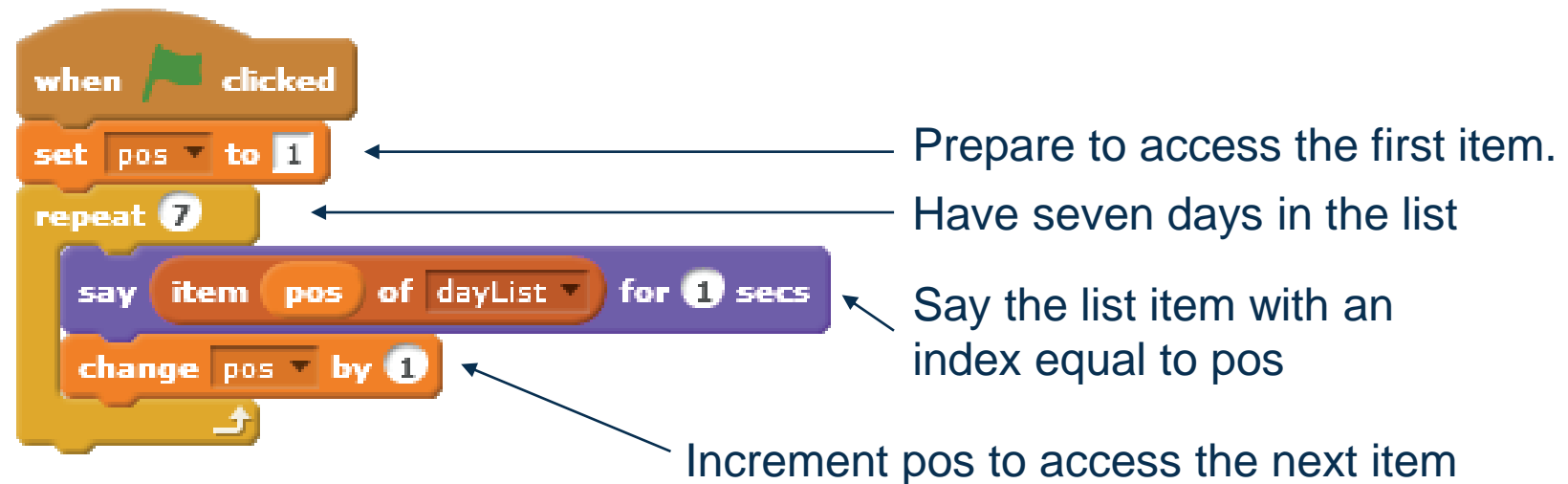
Return the number of items in the list

dayList contains thing

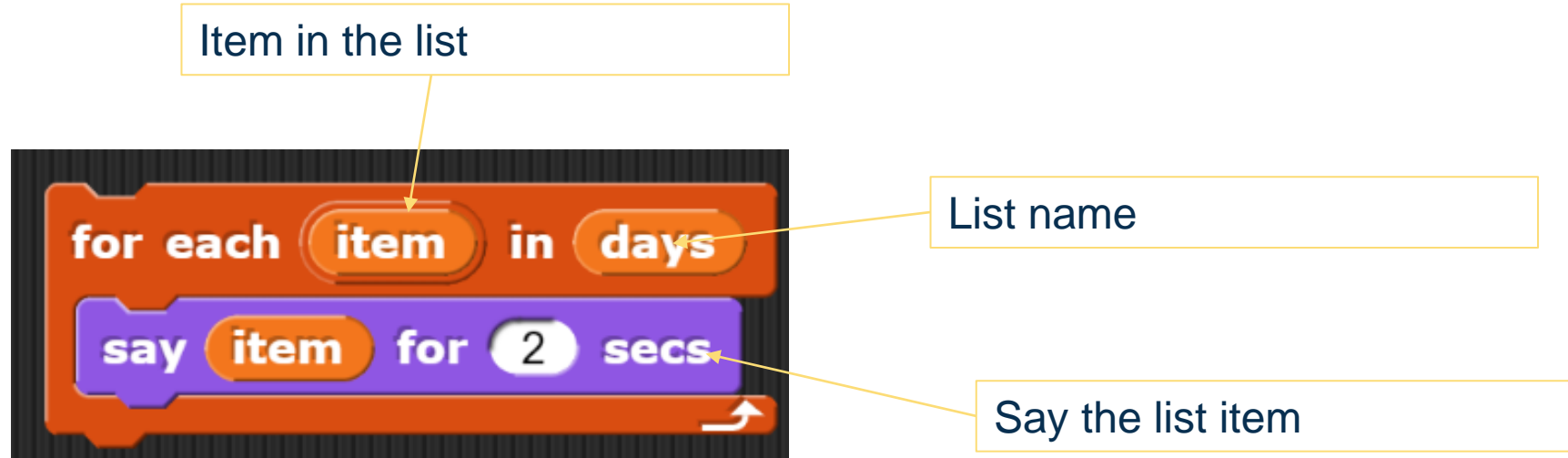
Does the list contain the specified item?

Accessing List Elements – Using Repeat Loop

- Programmer can access any element in a list using the element's index.



Accessing List Elements – Using For Loops



Dynamic List

- Lists are powerful because they can grow or shrink dynamically as a program is running.
 - Filling Lists with User Input
 - There are two common ways to fill a list with data entered by a user.
 - Asking how many entries there will be and then starts a loop to collect the user's input.
 - To have the user enter a special value (known as a sentinel) to mark the end of the list

Dynamic List

```

ask How many scores you have? and wait
set numScores to answer
repeat numScores
  ask Enter a score and wait
  add answer to scoreList
  
```

Asking the user how many scores will be entered

The contents of scoreList if the user enters 85, 100, 95, -1

Using a sentinel to control list growth

scoreList	
1	85
2	100
3	95
+ length: 3	

```

forever
  ask Enter next score [-1 when done] and wait
  if answer = -1 then
    stop this script
  add answer to scoreList
  
```

Numerical Lists

- Lists of numbers appear in many practical applications.
- We can have lists of test scores, temperature measurements, product prices, and more.
- Some procedures in numerical lists that can be done in BYOB:
 - Finding minimum and maximum value.
 - Finding the Average

Exercise Question



Based on the algorithm given, write a Snap! Program.

Repeat

 Get an item

 Add item to list

Until you can't think of any more items

Repeat

 Get any value from user

 Check if in the basket

 If item not in basket then

 Get item

 Else

 Say anything

Until end of list

Summary / Recap of Main Points

- List
 - Add
 - Insert
 - Delete
 - Accessing the list item