

CHAPTER 7

Lists

starting out with >>> **APP INVENTOR**
FOR ANDROID



TONY GADDIS · REBECCA HALSEY

Topics

- Creating a list
- Iterating over a list with the `for each` loop
- Selecting an item
- Inserting and appending items
- Removing items
- Replacing items
- Searching for an item
- Other list operations

Replacing Items

- Replacing an item in a list means to change the value of one item to a new value.
- The index positions are unchanged.
- The `replace list items` block requires a list variable, an indexed item to replace, and a new value.

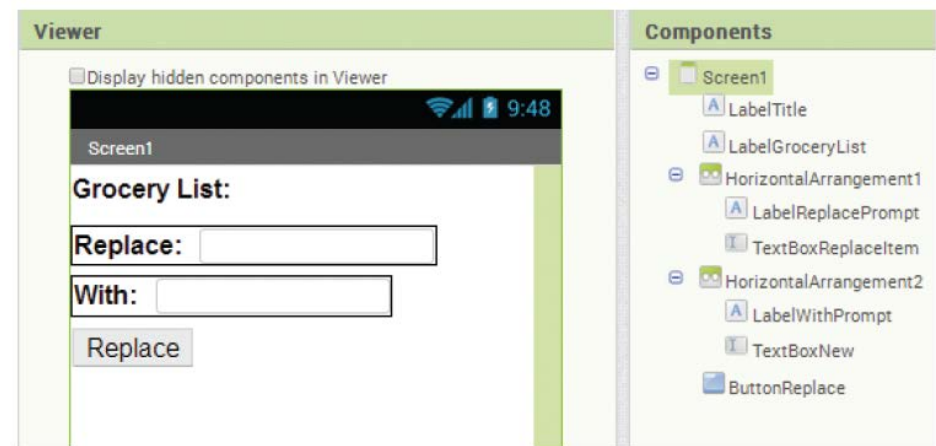
Figure 7-63 Replacing an Item (Source: MIT App Inventor 2)



Replacing Items

- In Figure 7-64 we have a place for the user to indicate the index, and a text box field for the value of the new item.
- There is also a button that causes the item to be replaced.

Figure 7-64 Grocery List App Screen Design (Source: MIT App Inventor 2)



Replacing Items

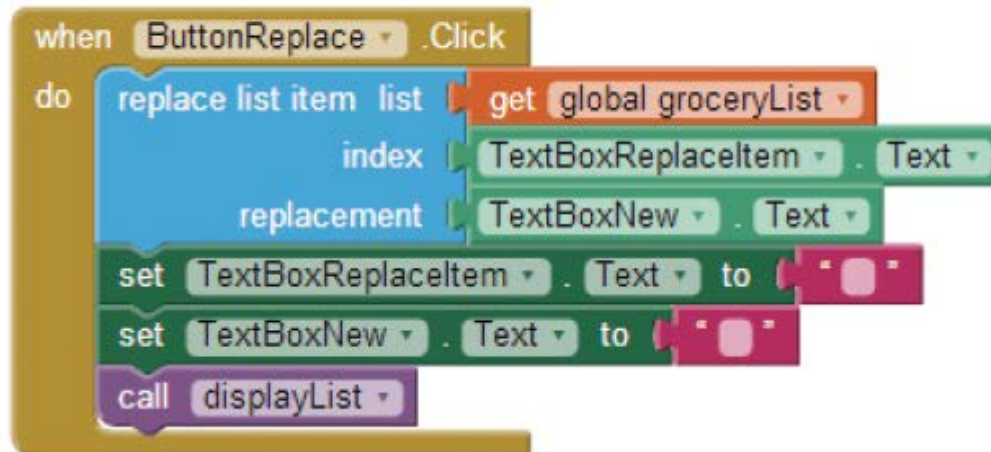
Figure 7-65 Blocks in the Grocery List App (Source: MIT App Inventor 2)

```
initialize global itemIndex to 0
initialize global groceryList to [make a list milk butter eggs apples oranges]
when Screen1.Initialize do call displayList
to displayList
do
  set global itemIndex to 0
  set LabelGroceryList.Text to ""
  for each groceryItem in list [get global groceryList]
  do
    set global itemIndex to [get global itemIndex + 1]
    set LabelGroceryList.Text to [join LabelGroceryList.Text [get global itemIndex] [get groceryItem] "\n"]]
when ButtonReplace.Click do
  replace list item list [get global groceryList]
  index [TextBoxReplaceltem.Text]
  replacement [TextBoxNew.Text]
  set TextBoxReplaceltem.Text to ""
  set TextBoxNew.Text to ""
  call displayList
```

Replacing Items

This figure contains essentially the same blocks as in section 7.5, removing items. The only difference is the when `ButtonReplaced.Click` do event handler shown in Figure 7-66.

Figure 7-66 `ButtonReplace.Click` Event Handler (Source: MIT App Inventor 2)



Replacing Items

- Here we use the replace list item function block.
- The `TextReplaceItem.Text` for the index (this is the number that the user will type in).
- The `TextBoxNew.Text` for the replacement.
- Once we replace the item, we should clear out the `TextReplaceItem.Text` and `TextBoxNewItem.Text` and call the procedure to re-display the list.

Searching for an Item

- When we search for an item in a list, generally we are interested in two things: does the item exist in the list? If so, where or in what position is it?
- To search a list use `is in list?` and `index in list` from the *List* drawer.

Figure 7-79 Searching Blocks (Source: MIT App Inventor 2)



Searching for an Item

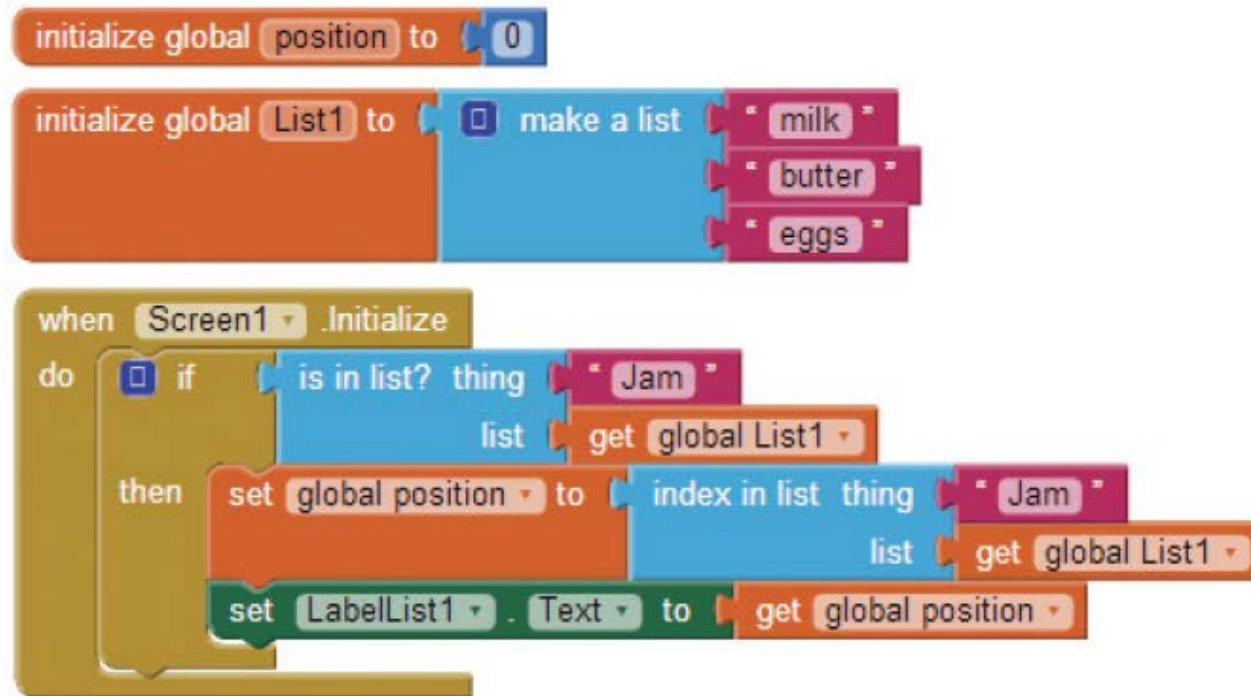
- The `is in list?` Block will return a true or false value.
- The `index in list` will return an integer, representing the index.

Searching for an Item

- In Figure 7-80, we have a variable named `position`, a list with a few items as `List1`, and the `when Screen1.Initialize do` event handler.
- `Screen1.Initialize` is used to find a position of *Jam*, which is not in the List.
- When this example runs, the result is 0.
- It does not cause an error. It may be beneficial to check whether or not the items exist first as shown in figure 7-81.

Searching for an Item

Figure 7-81 Checking if an Item Exists (Source: MIT App Inventor 2)

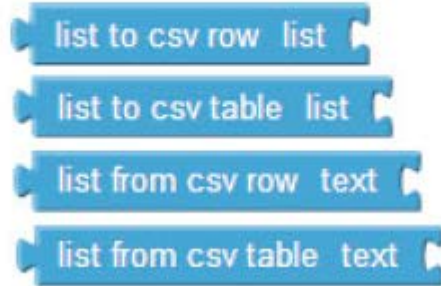


By adding the check `is in list?` before trying to find a position, we avoid any processing time if the value is return false.

Other List Functions

Other blocks can be used to create and write lists such as a comma-separated value file (CSV).

Figure 7-93 (Source: MIT App Inventor 2)



Other List Functions

- The first, `list to csv row` block, will take a list and return text that represents a single row of comma-separated values.
- The `list to csv table` assumes that each item in the list is a text block of comma-separated values and that each list item will represent an entire row.
- The `list from csv row` block will return a list made from comma-separated values.
- The `list from csv table` block will make a list that holds an entire row of the table in each list item.