

## Create a To Do list app

For our To Do list, we will want to be able to do the following:

- Add an item to the list
- Delete an item from the list
- Clear the entire list
- View the items in the list

In many of the previous instructions sets, we have told you everything to do. The knowledge will stick with you better if you think about what you need to do.

I like to read about my options. This is available from <http://ai2.appinventor.mit.edu/reference/components>

In the Designer block:

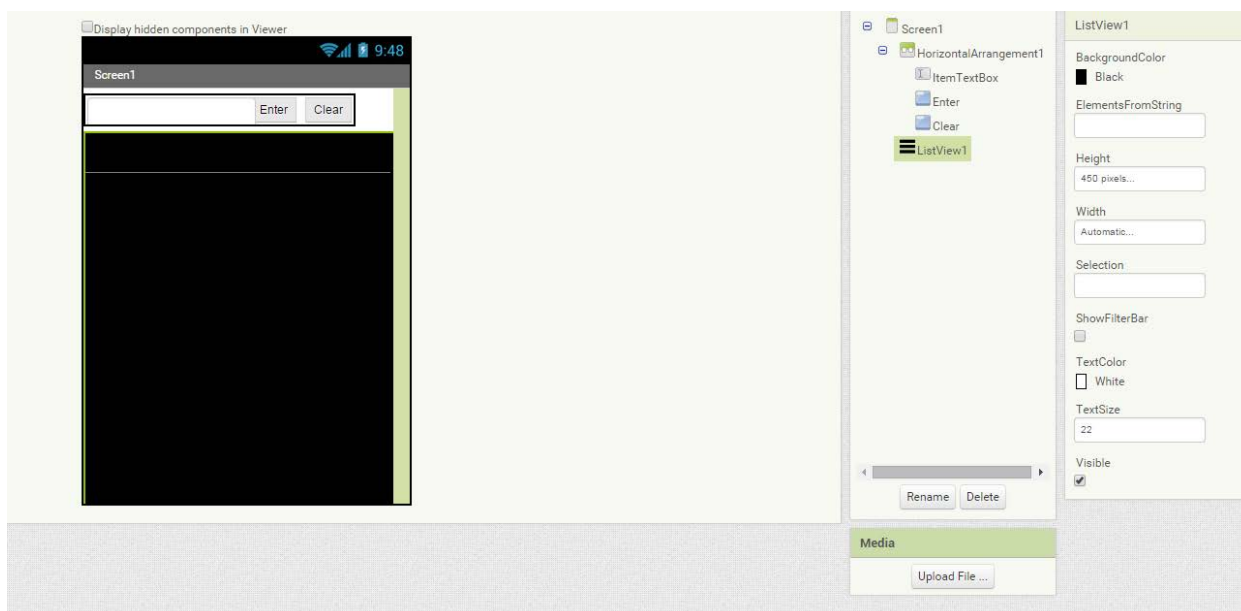
You will need:

Using Horizontal Layout:

- Textbox: to enter the item. I use the Textbox “hint” field to provide the user with a description of what he/she is to type. How does this differ from the “Text” you can also specify? Note you have to change the name of the button AND the Text for the button. Otherwise you won’t know which button you are programing in the Blocks tab.
- Button: To tell the system to add the item
- Button: to clear the entire list

A listView Component to display the contents of the list you create. You will drag this item below your buttons and size it to be 450 pixels high.

This is what I see in my Designer tab:



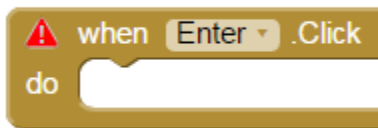
Have you figured out what kinds of things you do in the Designer Tab and what things you do in the Blocks tab? Visit with your neighbor and see if you agree on their uses.

When you move to the Blocks tab, what will you need to add?

Here are some things to consider:

- a. A global variable to store your ToDo List. Give it a name and set it to an empty list.
- b. The actions for when you click on the Enter Button. What do to if the item is already in the list?
- c. The actions for when you click on the Clear Button.

Go ahead and create your ToDo list and try adding something to it. You will need blocks such as



What do you want to do when the button is clicked?

Can you remember how to add something to the list? Click of the List label to see all of the options. Do you remember how to grab the contents of the TextBox? Click on the TextBox object to see all of the operations you can do with a text box.

After you have added the item to the list, there are two other things you want to do:

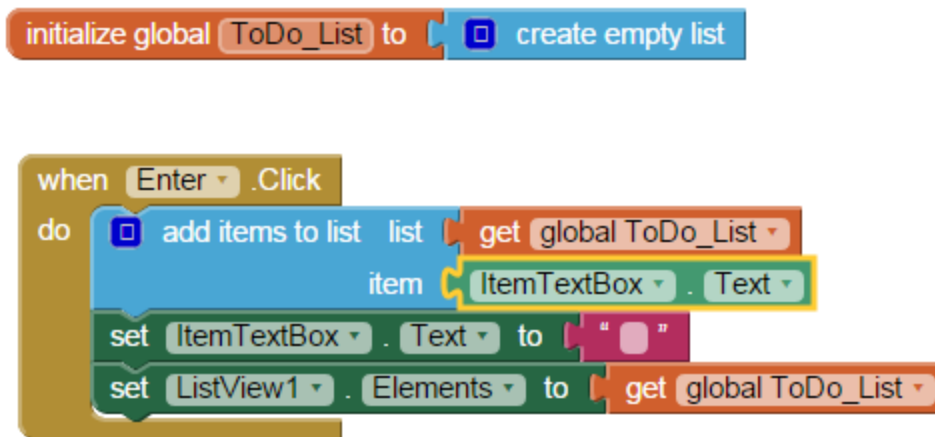
- a. Clear the text box, to make it easier to add the next thing.
- b. Allow the user to see what is currently in the list.

ListView

The listView object is designed to easily show the contents of a list. From the Blocks tab, when you click on your ListView object, it will show you what operations are available. You want the one that looks like:



This command is wanting the name of the list it should use. Grab the correct name from your Variables. Without looking at the example below, see if you can assemble all the correct pieces.



If we are adding something that is already in the list, we need to do two things:

- Determine if the item is already in the list.
- Tell the user about the error – and ignore the request.

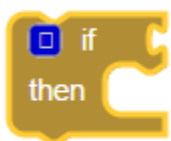
How will you check if the item is in the list? Luckily the List data type has something like that already built-in. Click on Lists to see the options. You are looking for an operation “Is in list?”. You will need to tell it what list you are asking about and what thing you are looking for in the list. Remember, these specific arguments to the request are termed parameters.

Where will you need to ask if the item it in the list? Raise your hand, and verify with your mentor that you have the right place.

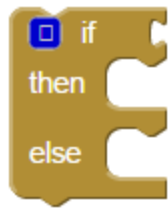
## Notifier

If the item the user tried to enter is already there, you want to tell the user. If it isn’t there go ahead and add it to the list (using your existing code).

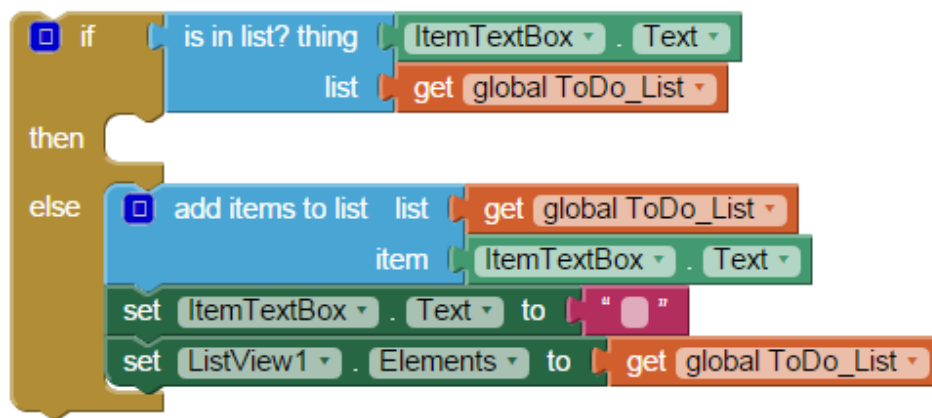
How will you do two things depending on a condition (like being in the list already)? We call that “Control”. You want an “if then block” like the one below. (It is confusing, as the other if-then-else block doesn’t connect in the right way.



To turn it into an “if then else” block, you must click on the blue square and manually move the else into place. It should now look like:

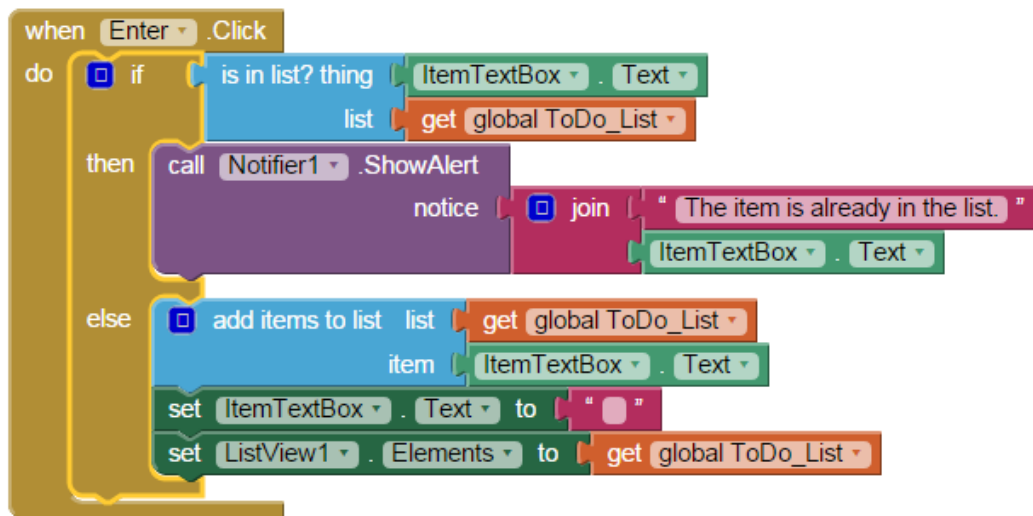


You will need to add the condition to the “if”. You need to move the code to add the items to your “else” block. So far it looks like:



How do you tell the user there is a problem? A Notifier is an object which notifies others about something that has happened. Of course, you could use a text box to place a message, but this is easier as the message disappears about the user had seen it so you don't waste screen space (sometimes called real estate) for a message that you only need part of the time. Go to your Designer tab and select a Notifier (in the User Interface drawer). It is a non-visible component, so as you drag it, it will appear below the screen.

From the Blocks tab, when you click on Notifier1, it will show you all the operations you can do with a Notifier. We just want to call **Notifier ShowAlert** to give it the message about the duplicate entry. You can be as clever as you want in telling the user what the problem is.

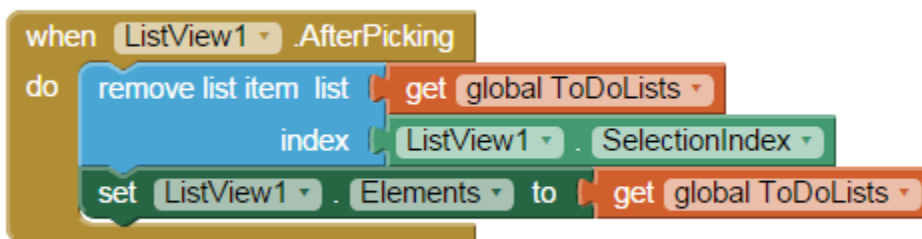


Check that out! Show your mentor when you have that working.

How would you clear the to do list?

The next step is to delete a single item in the list. Our listView has a method designed specifically for selecting an item. It is called "When ListView After Picking". When we click on an item of the listView, this method is called. It will do anything we want with the item that is selected.

We want to remove an item from the list. Go to List to find how to remove an item from a list. [Remove List Item](#) needs two parameters – which list and which item in the list. We know the name of the list, right? Luckily, our ListView will tell us which item is selected. We actually want the index (not the name of the item). Remember, the index is the number (1,2,...) which tells you whether it is the first, second, or so on element. The function which gives this is called "SelectionIndex". My code looks like:



Extension

Insert the items in order.

Use tinyDB to save the items in your list, even when you cell phone is off.