

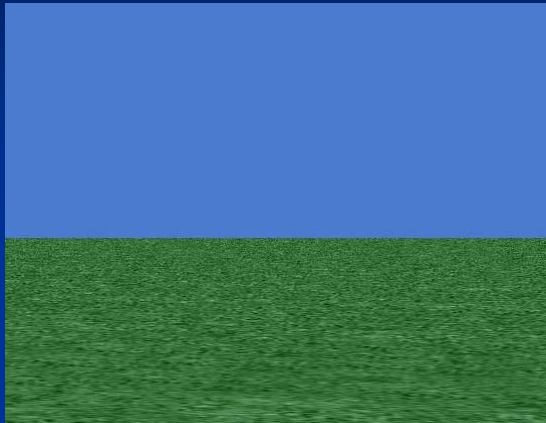
Tutorial for Arrays and Lists

By Ruthie Tucker

Description

- This presentation will cover the basics of using Arrays and Lists in an Alice world
- It uses a set of chickens on a farm
- Prerequisites
 - Everything
 - This is the last tutorial you should do in this course

The Characters



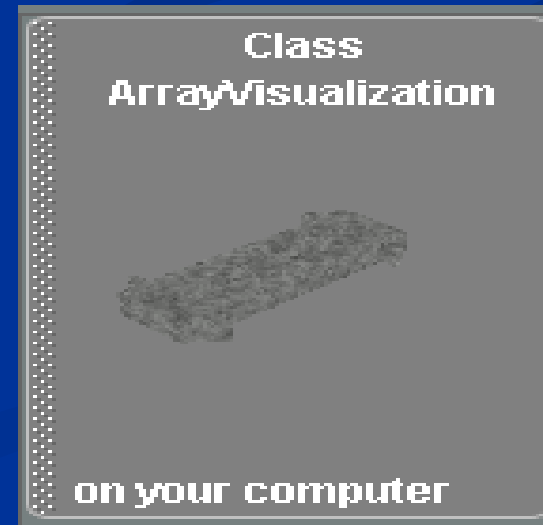
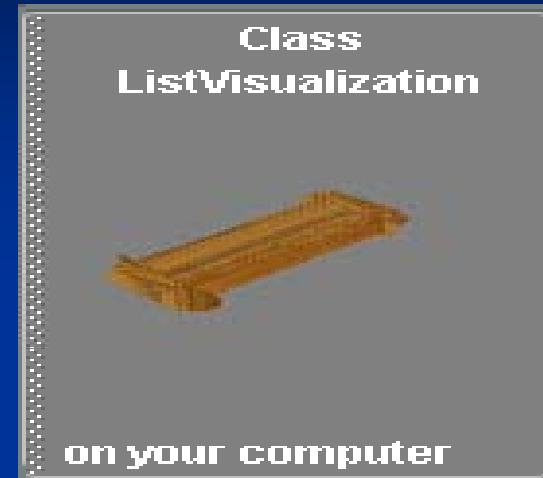
Side Note

■ Glitch in Alice

- Alice cannot compute your world if you put the same objects in a “List Visualization” as an “Array Visualization”.
- I will demonstrate how to use the “List Visualization” later, but for now we are just going to make a regular list.
- However, you can use a list without the actual ListVisualization, if you also want to use an “Array” in your world.

Getting Started

- What is a List?
 - A list is simply a way to organize information.
- What is an Array?
 - An Array is a structure for collecting and organizing objects or information of the same type into a group.
 - In Alice Arrays and Lists are also visual objects which can be dropped into your world



Cinderella Conducts

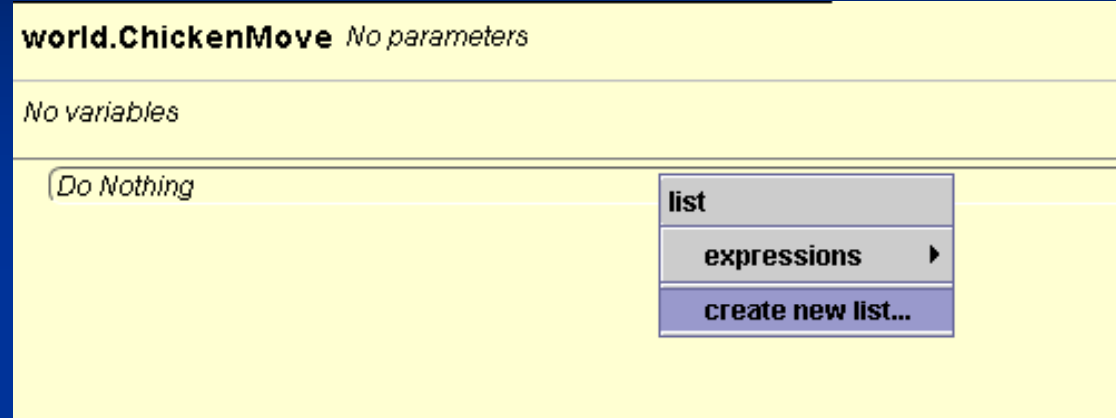
- We are going to make the world you just watched and use Lists and Arrays to make the chickens dance
- This is the code to make Cinderella conduct while the chickens dance. If you want to add this to your world you can code it now and save it. We will use it at the end.

The image shows a Scratch code editor with the following structure:

- Do in order**
 - // Farm girl prepares to direct chickens
- Do together**
 - cinderella.hips.torso.upperBody.leftUpperArm roll left 0.25 revolutions more...
 - cinderella.hips.torso.upperBody.rightUpperArm roll right 0.25 revolutions more...
- cinderella turn left 0.5 revolutions more...
- Do together**
 - cinderella.hips.torso.upperBody.leftUpperArm roll right 0.2 revolutions more...
 - cinderella.hips.torso.upperBody.rightUpperArm roll left 0.2 revolutions more...

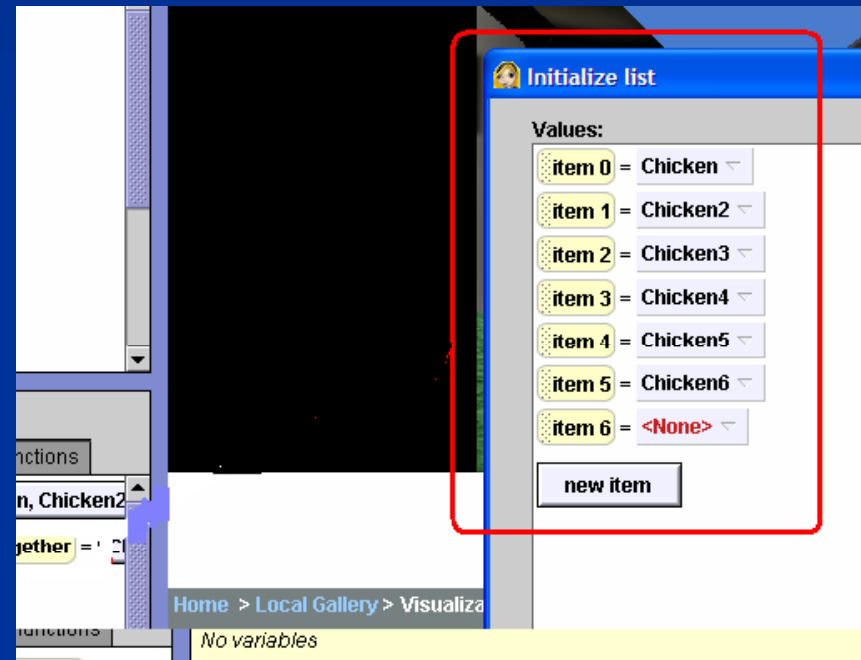
Making a List

- Create a world level method named “ChickenMove”
- Drag the “For all Together” Button onto the method editor
- Select “Create new list”



Lists

- Name it “ChickDanceTogether”, select “object” as the type.
- A white box will come up that says “initialize list”. Click new item until all the chickens are in your list. The first item starts at 0. Keep your chickens in order.



Lists



- We want the chickens to roll left and right together.
- First you need to right the code to have one chicken roll left and then right.
- Above is the code to have one chicken roll left and then right
- I selected Chicken 2 but any chicken will work

Lists

- Now drag the tab label “item_from_ChickensDanceTogether” over the box that was previously labeled “Chicken 2”
- Repeat this for each command to make the chickens roll

The image shows a Scratch code editor interface. At the top, there is a 'For all' loop block with the label 'world.ChickensDanceTogether'. Inside the loop, there is a 'Do in order' block. The 'Do in order' block contains three 'roll' commands. The first 'roll' command is highlighted with a red box, and a red arrow points from the 'item_from_ChickensDanceTogether' label in the 'For all' loop to the 'roll' command. The 'roll' commands are as follows:

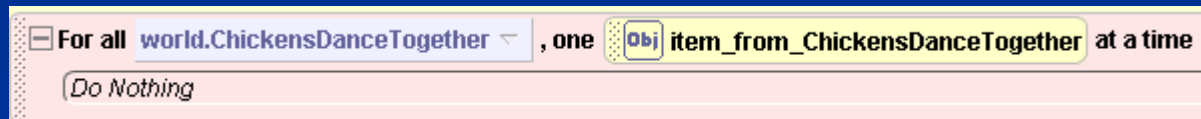
Label	Direction	Revolutions	Duration	More..
item_from_ChickensDanceTogether	roll left	0.2 revolutions	duration = 1 second	more..
item_from_ChickensDanceTogether	roll right	.4 revolutions	duration = 1 second	more..
item_from_ChickensDanceTogether	roll left	0.2 revolutions	duration = 1 second	more..

Lists

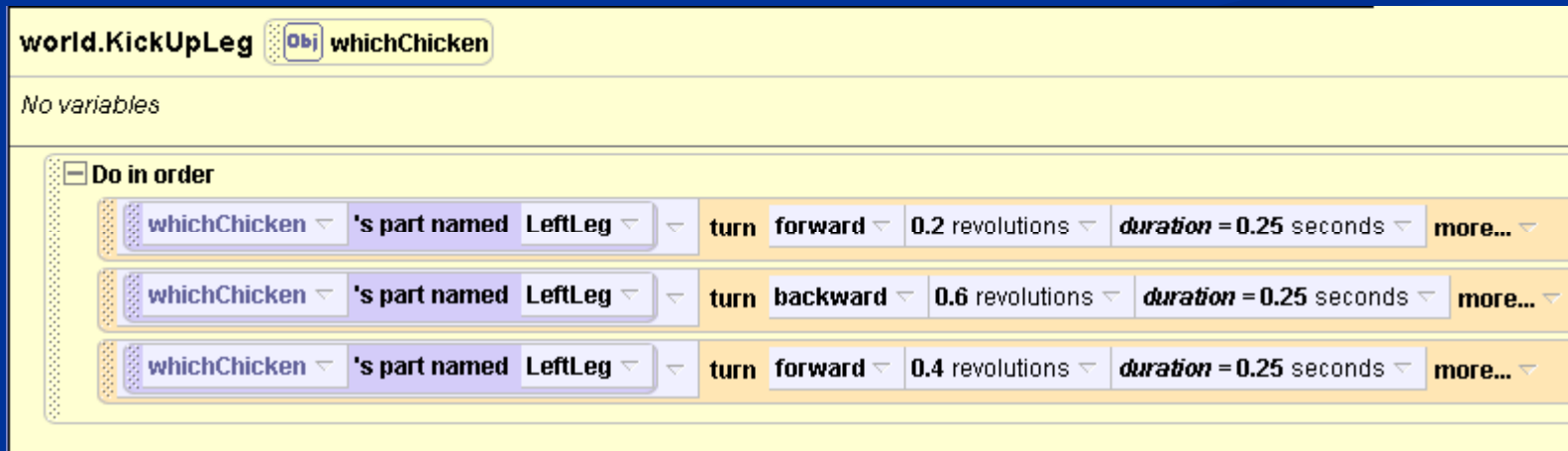
- Now we have programmed the Lists to have all the chickens roll left and then right at the same time.
- But what if we want them to do something, one at a time?
- This is where ‘For all in order’ comes in.
- We will now program the chickens to kick up their left leg, one at a time.

Lists

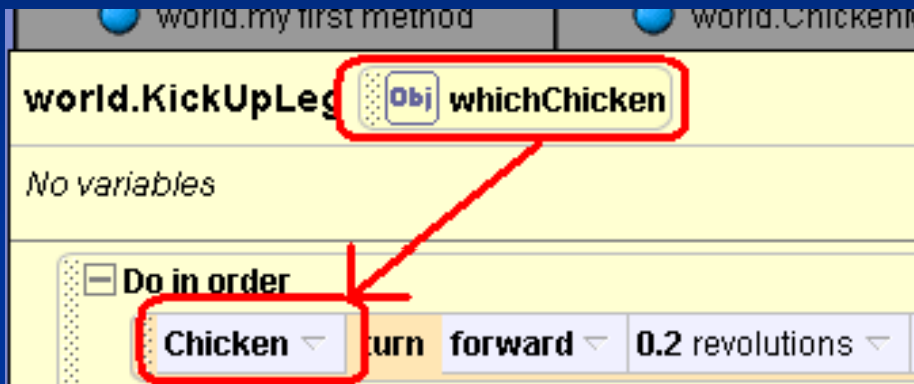
- Click on the “For all in order” tab at the bottom of your screen and drag it into your method



- Now create a new method named “KickUpRightLeg” and give it an object parameter called “WhichChicken”
- Below is how the code will look when we have finished the next few slides

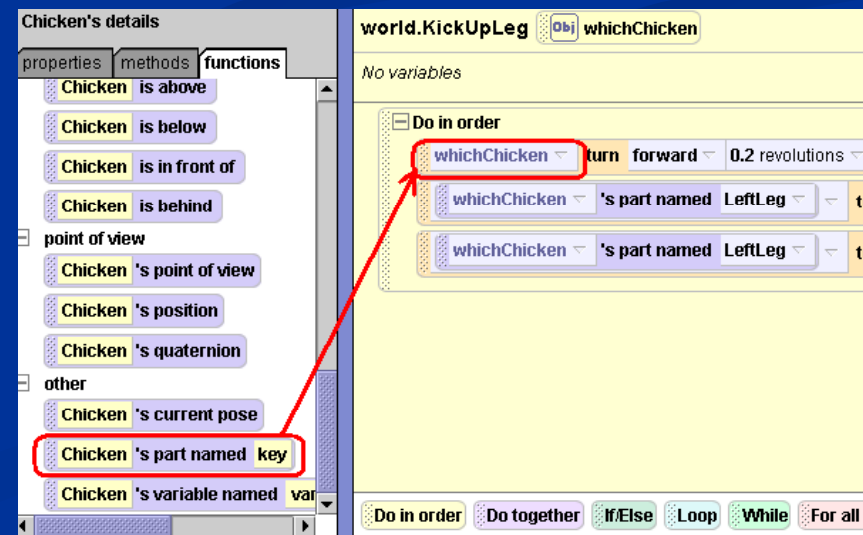


Part Named Key



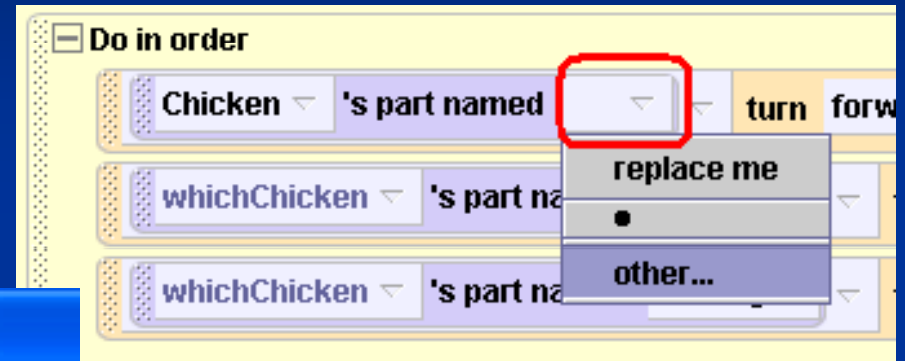
- Make a simple command using any object like “chicken” and have it do the command. Then, create a parameter and drag it into the original object.

- Click on any object (like chicken)
- Now go to functions and scroll down to “Part named Key” insert it everywhere you have a parameter



Part Named Key Continued

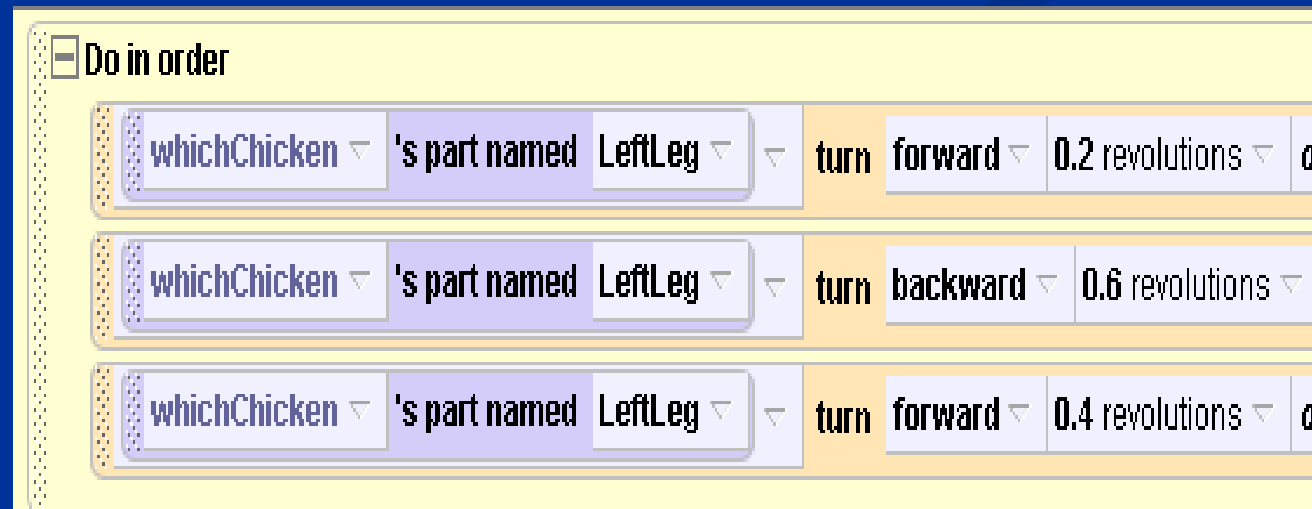
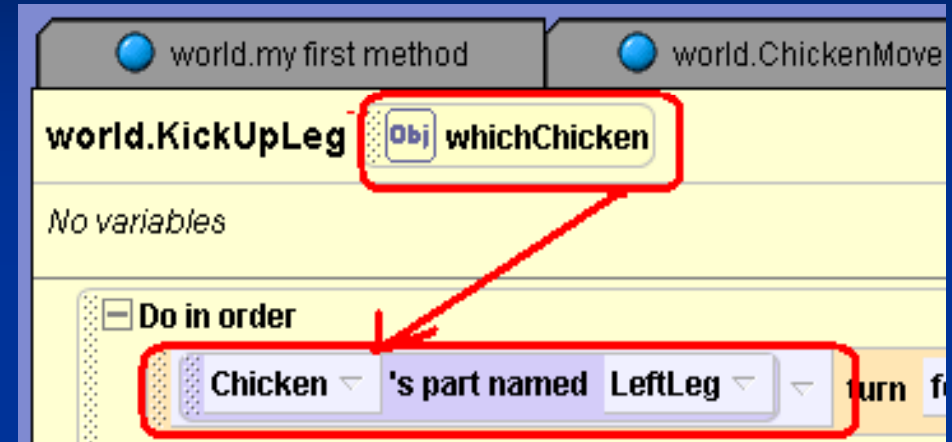
- Now, click on the white box (circled in red) and scroll down to “other”.



Once you click on “other” and type in the appendage that you want to command. In this case the left leg. Remember the syntax must be the same as it is listed on the object

Part named key conclusion

- Finally take your object parameter and drop it back in where it says chicken
- Your code should look like the picture below when you are done



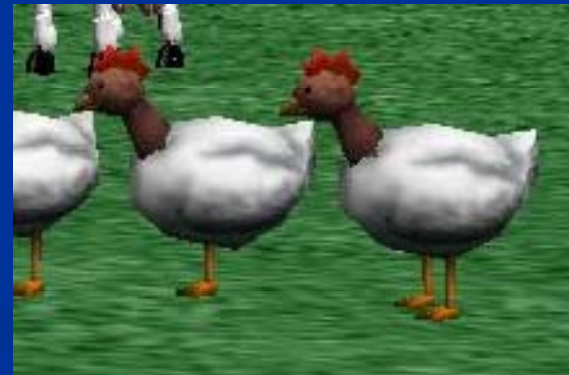
Lists

- Now go back into your first method and drag your “KickUpRightLeg” method into the “For all in Order” space, in your world level “chicken move” method.
- Select Expressions and then Item_from_ChickensDanceTogether



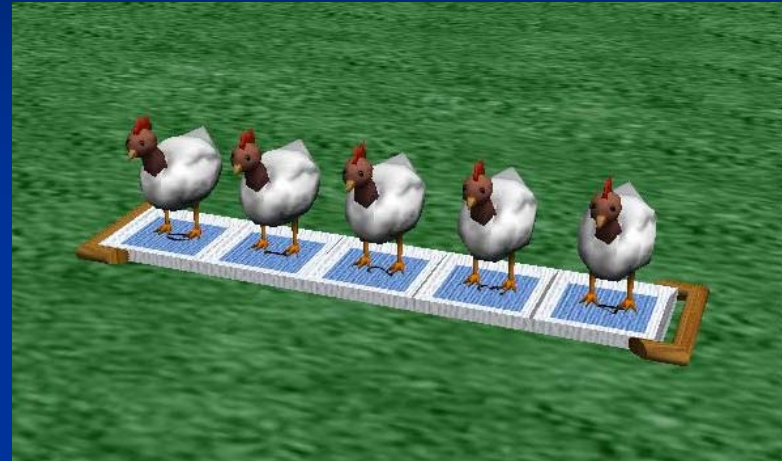
Lists

- Now that this portion is programmed all the chickens will roll right and then left at the same time for the “Together Functions” and kick up their legs one at a time for the “For all in order Functions”
- This saves lots of time and lines of code
- Imagine if we had to program each chicken to do this individually!



List Visualizations

- If you are not also using an Array and would like to visualize your list you can select the list visualization class from the Visualizations folder
- DO NOT DO THIS RIGHT NOW (BECAUSE WE ARE GOING TO CREATE AN ARRAY AS WELL) OR IT WILL MESS UP THE LIST YOU JUST CREATED
- After entering in all of the Chickens they should be aligned in a row that looks like this.
- If you “Set is Showing to False” the List Visualization will disappear, but it is still there.



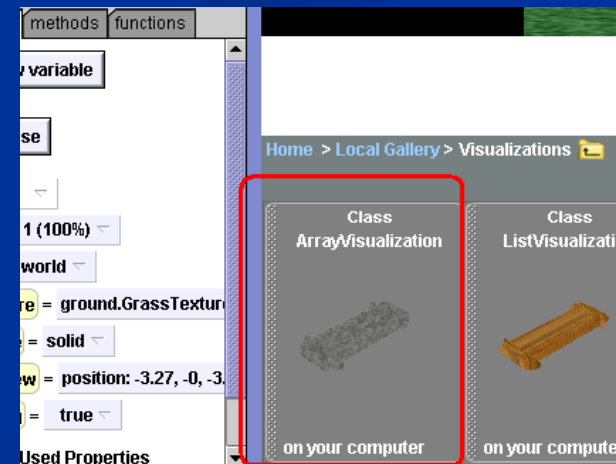
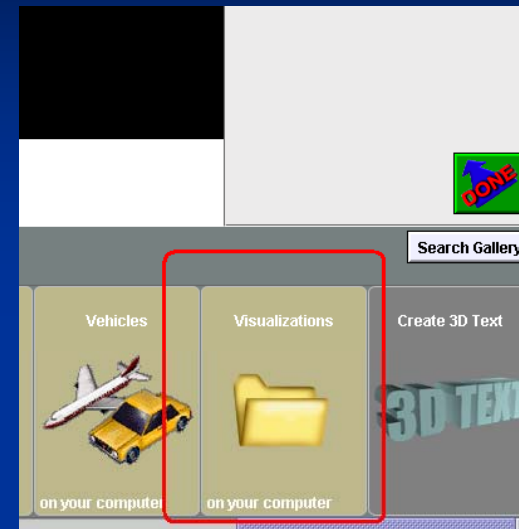
- If we were not going to create an Array you could add the List Visualization without problems.

Arrays

- Lists are great for coding a group of objects to all do the same thing (either together or in order).
- But what if we want to code only some of the objects in our group to do something.
- This is where Arrays come in.
- The following slide will show the place that you need to go to find the Array. It is called “Array Visualization”.
- Drop it into your world.

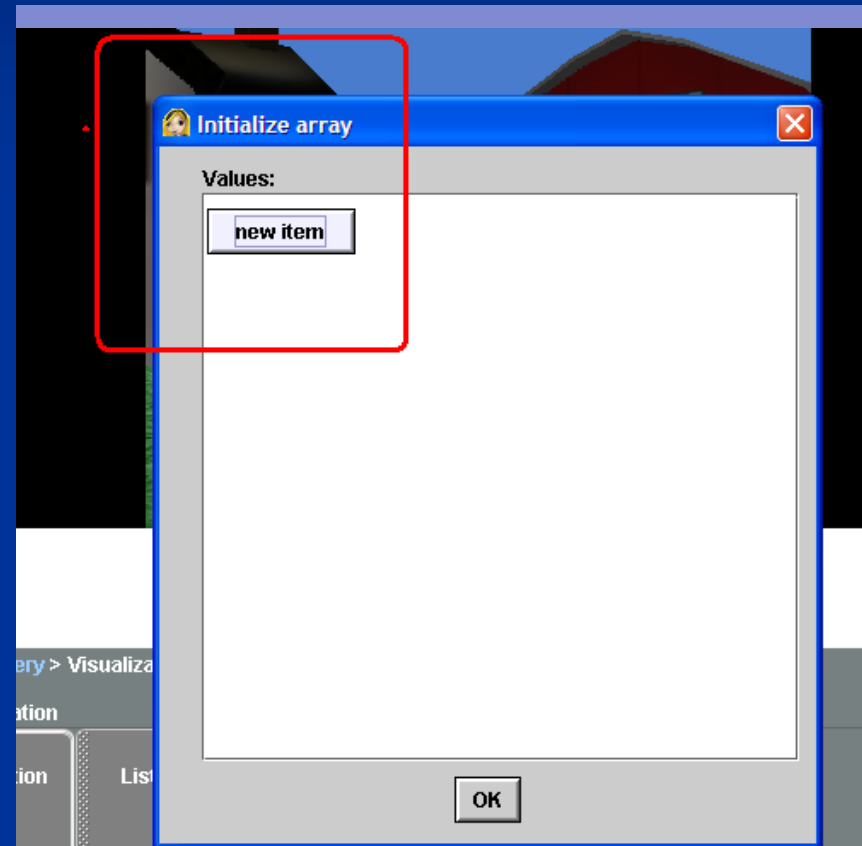
Starting the Array

- Go to add objects
- Select the folder at the end of the list entitled “Visualizations”.
- The first object on the left should be a class called “Array Visualizations”.
- It should look like a strip of concrete.
- Select it



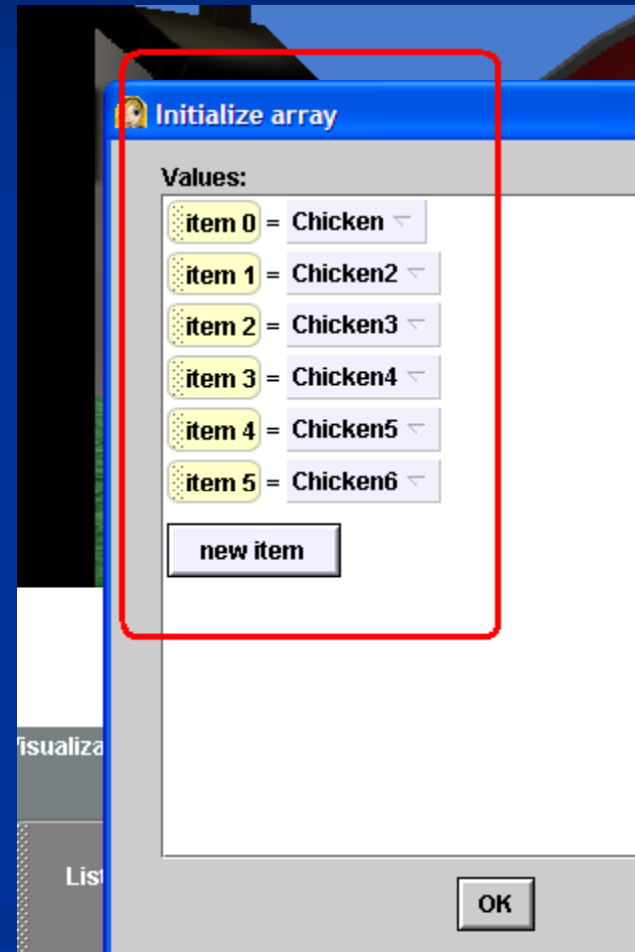
Starting the Array

- We are going to use our Array for the chickens
- When you add the Array to your world this box called “Initializing Array” will appear
- Count the number of chickens you have and select the button “New Item” for that many.
- Remember it will start at 0



Array continued

- Once you have entered all of your chickens into the Array your screen should look like this.
- If it does, click ok. If not, hit “undo” at the top left and try again
- Once you hit ok your chickens should be neatly lined up on the Array.



Arrays

- Click on your “Array Visualization” and then “properties”. Set “Is showing to False” to hide your Array Visualization so that the chickens do not look like they are standing on anything.
- However, the Array is still there and functional

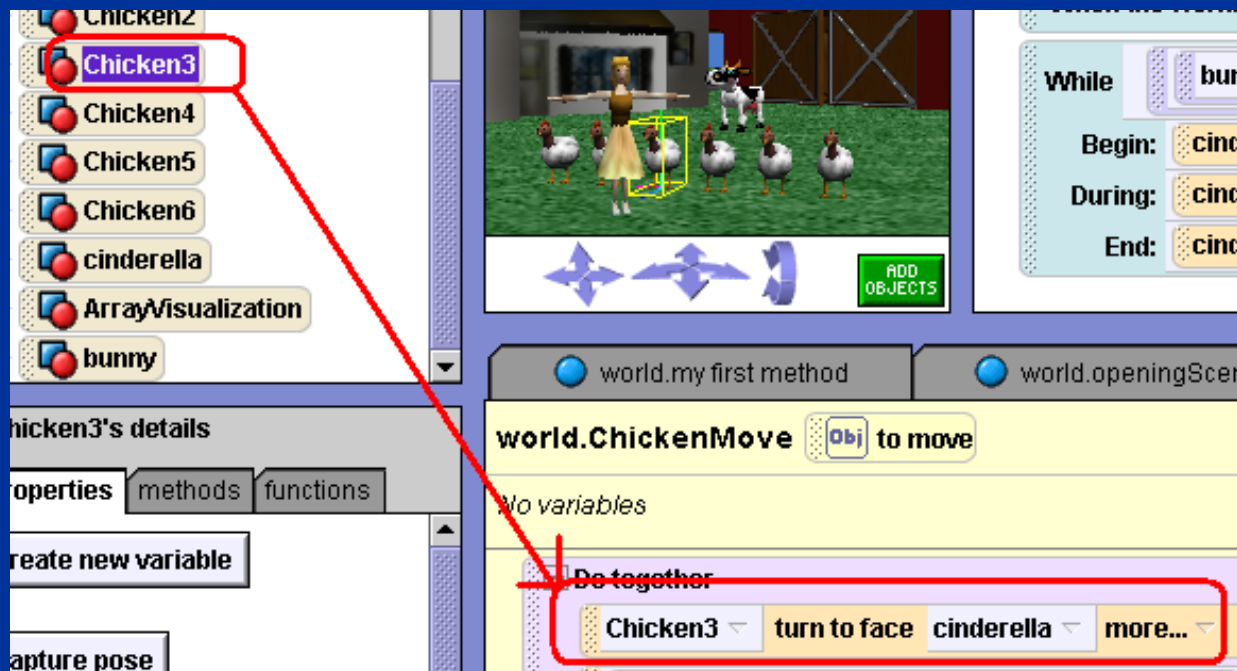
Array Continued

- This is how our code will look when we are done.
- Note the small red circle on the left
- This is how your array will appear in your tool bar.
- The large circle on the right is the beginning of how we will use our Array to animate our chickens.



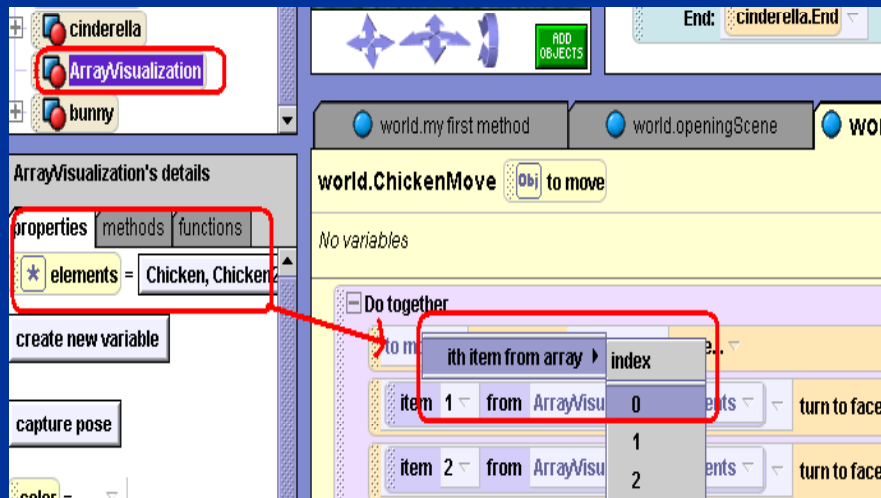
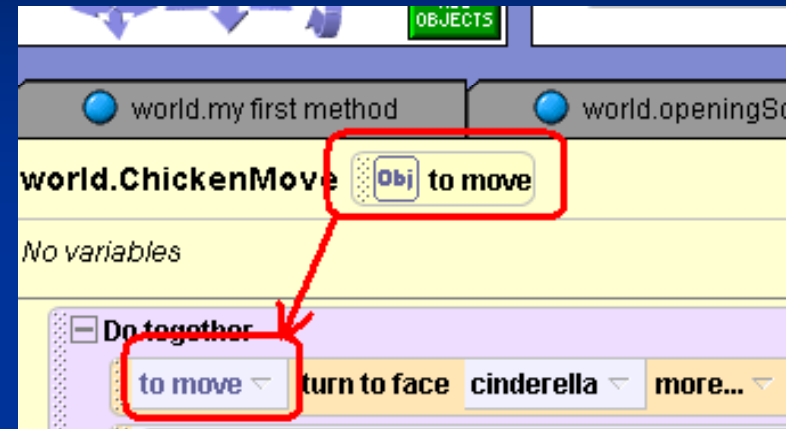
Arrays Continued

- First, let's have our chickens all turn to face Cinderella. Again you can code this with any chicken. See below



Arrays Continued

- Now create a parameter
- You can name it whatever you would like
- Drag it into your chicken command (picture on right)



- Now go to Array Visualizations, properties, and drag the “elements” tab down to your parameter.
- Select 0, repeat this step for each chicken in your Array

Complete Turn to Face Code

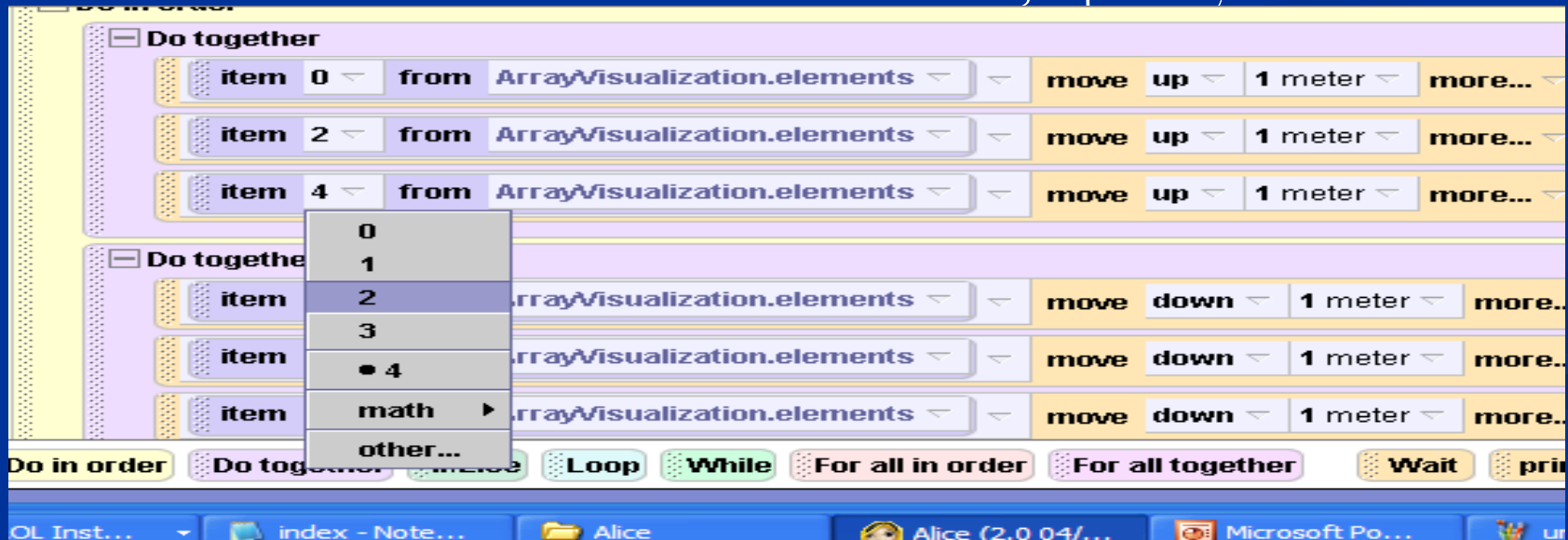
- Note that each element in the Array is coded separately but they are all doing the same thing. Now let's get them to do different things.

The image shows a Scratch code editor with a 'Do together' loop block. The loop contains six identical code blocks, each with a different element index from an array. Each block consists of a 'from' block and a 'turn to face' block.

Iteration	Element	Source	Action	Target
1	item 0	ArrayVisualization.elements	turn to face	cinderella
2	item 1	ArrayVisualization.elements	turn to face	cinderella
3	item 2	ArrayVisualization.elements	turn to face	cinderella
4	item 3	ArrayVisualization.elements	turn to face	cinderella
5	item 4	ArrayVisualization.elements	turn to face	cinderella
6	item 5	ArrayVisualization.elements	turn to face	cinderella

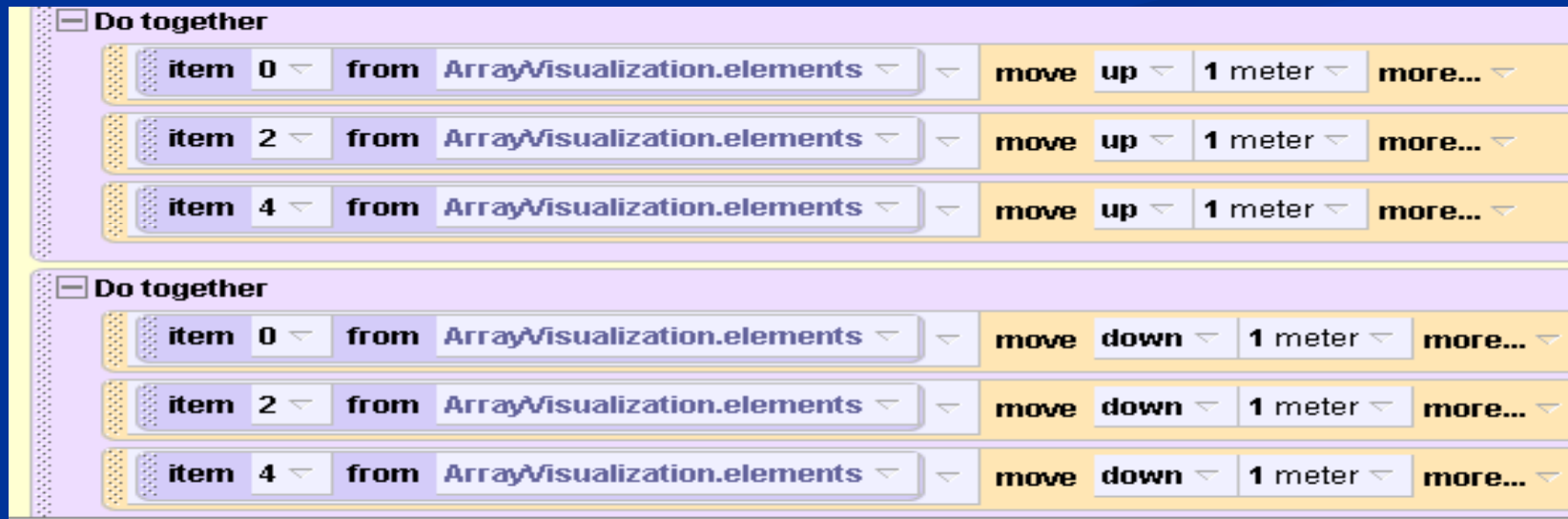
Getting Chickens to do Different Things

- To get the chickens to move up and down at different times you will follow the same steps that you just performed with "Turn to Face" in the Cinderella slide.
- See the code below
- Note that we only selected chickens 0, 2 and 4 to perform this task.
- After making the even number chickens move up and down, add code to make the odd number move, separately.



Arrays Continued

- Once you have finished plugging in the Array your code should look like this
- Note that only chickens 0, 2, and 4 are going to move.
- This is the beauty of the Array
- It allows you to only move selected objects in your Array
- Whereas with the List you must move everything



Arrays

- This is the visual
- Note how Chickens 0, 2 and 4 are moving down while chickens 1, 3, and 5 are moving up.
- You can program each chicken individually or in any number you want to



Putting it all together

- For the Final construction I created a BDE event and did the same trick that was used in the vehicle tutorial. Put an invisible bunny in the world and have him change colors. While he is one color Cinderella conducts, while he is a separate color she does not.
- The bunny will turn one color while the chickens begin to dance, this will trigger Cinderella to start conducting. When the chickens finish, the bunny will turn a different color, which will signal Cinderella to stop.

Final construction

Events create new event

When the world starts, do world.my first method CHICKENS! = Chicken

While bunny . color == is true

Begin: cinderella.Conduct

During: cinderella.Conduct

End: cinderella.End

■ *And that's all folks!*

Recap

- When we put all of these elements together you can see how Lists and Arrays can be used interchangeably to animate your Alice world.
- The 'For all together' command in Lists allows you to animate a group of objects to do something together.
- The 'For all in order' command in Lists allows you to animate the whole group of objects to do the same thing, but one at a time.
- The Array allows you to have a group of objects do different things.