ORACLE ACADEMY

Tutorial 4: Declare Procedures

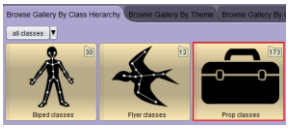
Overview	In this tutorial, you will practice declaring your own procedures to simplify animation movement.
Key Concepts Learned	• Define and compare an animation and a scenario • Demonstrate how to declare a user-defined procedure • Identify and use procedural abstraction techniques to simplify animation development
Difficulty Level	Intermediate: This tutorial is appropriate for someone who has previously used Alice 3 to: • Add and position objects in a scene • Code programming statements in the Code editor • Randomize movements
Duration	1 hour
Notes	This tutorial was built using Alice 3.1.81.

Part 1: Define the Scenario

Review the scenario and corresponding animation. In this tutorial, you are going to create an animation from the scenario defined below.

Scenario	**Animation**
An elephant is thirsty and needs to find water.	The elephant walks to a local watering hole to get a drink of water.

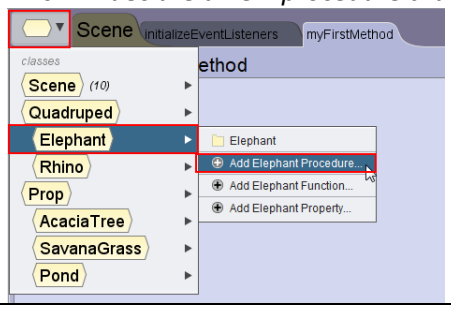
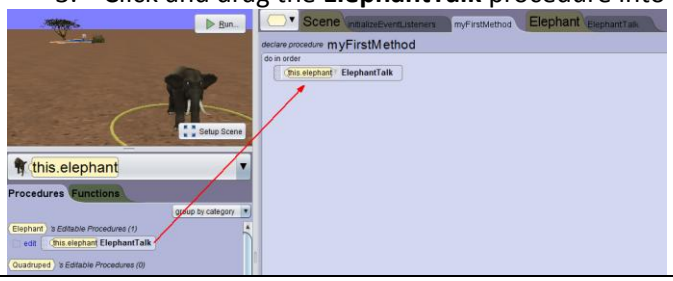
Part 2: Create the Scene

1.	Open Alice 3 and select the **DESERT** template.
2.	Save the animation project with the file name **Tutorial 4**. *REMEMBER: SAVE FREQUENTLY AS YOU WORK!*
3.	In the Gallery, go to the **Browse Gallery by Theme** tab  . Select the **Africa** theme.
4.	Add the following objects to the scene: • 1 African Elephant • 1 Rhino • 1 Acacia Tree • 3 Savana Grass
5.	Go to the **Browse Gallery by Class Hierarchy** tab. Go to the **Props** folder. 
6.	Add a **Pond** object to the scene.
7.	Position the objects in the scene. When you are finished, your scene should look like this:

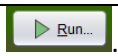
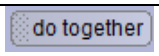
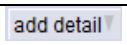
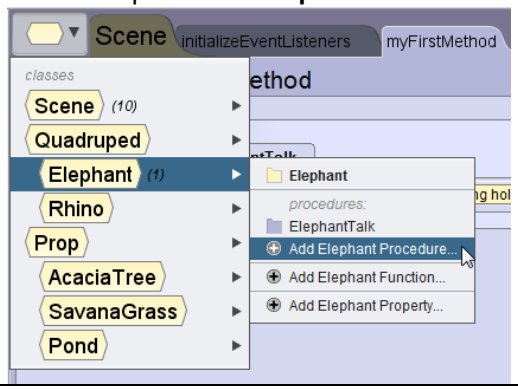
Copyright © 2014, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

	 Add additional objects to the scene to make it unique, such as additional plants, trees, or animals.

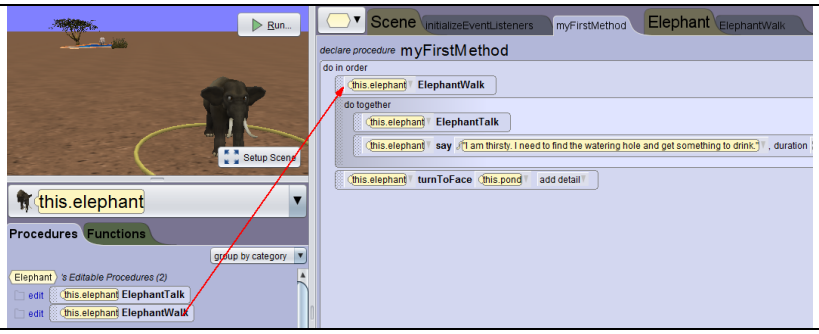
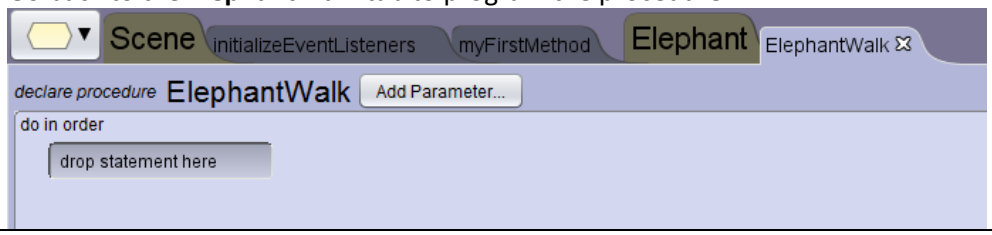
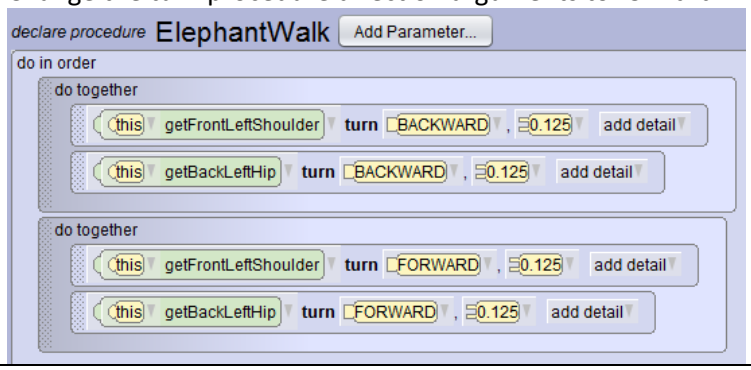
Part 3: Program the Animation

1.	Go to the Code editor. *First, you will declare a procedure for the elephant's talking motion. A procedure must be declared so that the elephant will move its trunk while it is saying something.*
2.	Select the **Elephant** from the instance menu.
3.	From the **Class menu**, select the **Elephant class**, then **+Add Elephant Procedure**. *This will declare a new procedure that all elephant objects of the Elephant class may use.* 
4.	Name the procedure **ElephantTalk**. Click **OK**.
5.	*Before coding the procedure, you need to add the new ElephantTalk procedure to the myFirstMethod tab. This is so you can click the Run button to test how the procedure functions as you write the code for it.* 1. Click the **myFirstMethod** tab. 2. Select the Elephant from the Instance menu. 3. Click and drag the **ElephantTalk** procedure into myFirstMethod. 

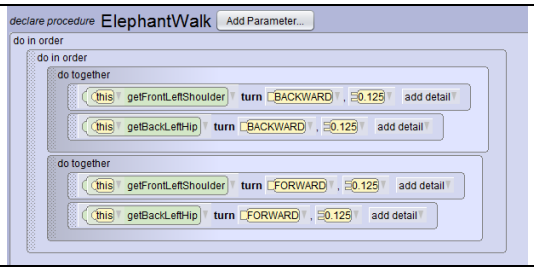
Copyright © 2014, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

6.	Go back to the **ElephantTalk tab**. *This is where you are writing the code for the ElephantTalk procedure. Right now, it does not have any code in it. You have to write the code that will tell the elephant how to move in order to simulate "talking".* 
7.	Select the **Elephant's neck** from the instance menu.
8.	Drag a **turn** procedure into the ElephantTalk tab. Select arguments: **BACKWARD → 0.125**.
9.	**Run** the animation  . Observe how the elephant moves its neck and head backward.
10.	Drag a **turn** procedure into the ElephantTalk tab. Select arguments: **FORWARD → 0.125**.
11.	**Run** the animation. *The ElephantTalk procedure is complete. Return to the myFirstMethod tab.*
12.	Drag a **Do Together**  into myFirstMethod. Drag the **ElephantTalk** procedure inside of the Do Together.
13.	Select the **Elephant** from the instance menu.
14.	Drag a **say** procedure into the Do Together. 1. Select **Custom TextString…** 2. Enter text: **I am thirsty. I need to find the watering hole and get something to drink.** 3. Click **OK**.
15.	Click the **Add Detail drop-down menu**  next to the say procedure and select **duration**. Change the **duration** of the procedure to 2.0.
16.	**Run** the animation.
17.	Click and drag a **turnToFace** procedure into myFirstMethod. Select the **pond**.
18.	*Now you need to declare an ElephantWalk procedure to create the walking movement for the Elephant so it looks realistic when the Elephant moves.* In the Class Menu, click **+Add Elephant Procedure**. Name the procedure **ElephantWalk**. 
19.	Go back to the myFirstMethod tab. Drag the **ElephantWalk** procedure into myFirstMethod, above all of the other procedures.

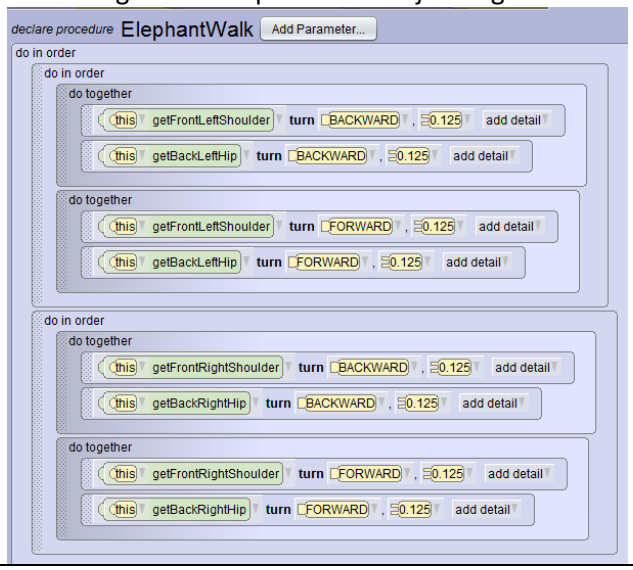
Copyright © 2014, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

	
20.	Go back to the **ElephantWalk** tab to program the procedure. 
21.	Drag a **Do Together** into the ElephantWalk tab.
22.	Select the **Elephant's Front Left Shoulder** from the instance menu.
23.	Drag a **turn** procedure into the Do Together. Select arguments: **BACKWARD** → **0.125**.
24.	Copy the **turn** procedure. Paste the procedure inside of the Do Together. Change the second turn procedure's object argument to the **Elephant's Back Left Hip**. 
25.	**Copy** the Do Together. **Paste** it below the first Do Together. Change the turn procedure direction arguments to **forward**. 
26.	Drag a **Do In Order** into the ElephantWalk tab. Drag both **Do Together** programming statements inside of the Do In Order. *This will make it easier to copy and paste the entire movement for the left side of the body to create the movement for the right side of the Elephant's body.*

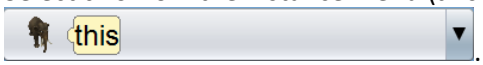
Copyright © 2014, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.



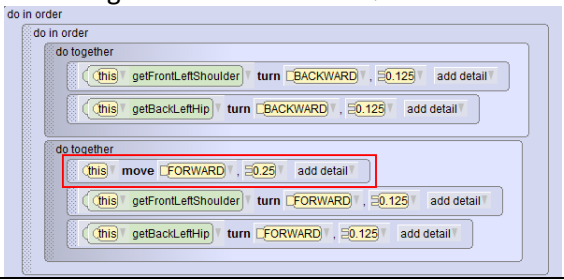
27. Copy the **Do In Order**.
Paste the **Do In Order** below.
1. Change the turn procedure object arguments from **FrontLeftShoulder** to **FrontRightShoulder**.
2. Change the turn procedure object arguments from **BackLeftHip** to **BackRightHip**.



28. *The Elephant moves its legs in a walking motion, but it doesn't move forward. You need to enter a move procedure in each Do In Order to make the Elephant move forward.*
Select **this** from the instance menu (*this* is a Java keyword that refers to the Elephant object)

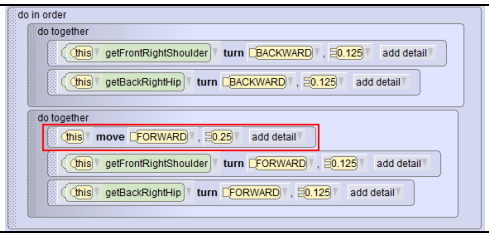
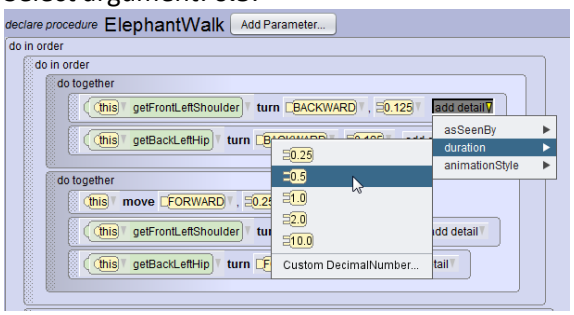
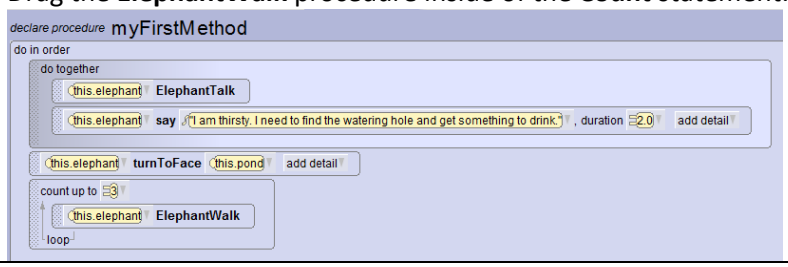


29. Drag a **move** procedure into the second **Do Together** of the first Do In Order programming statement that controls the left side of the body.
Select arguments: **FORWARD** → **0.25**.



30. Copy and paste this **move** procedure into the second Do Together of the second Do In Order programming statement that controls the right side of the body.

Copyright © 2014, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

	
31.	**Run** the animation.
32.	*The elephant moves a little slow. You can adjust the duration of each procedure to make the elephant move faster.* Click the **Add detail** drop-down menu in each procedure and select **duration**. Select argument: **0.5**.  Repeat this for all of the **turn** procedures in the ElephantWalk procedure.
33.	*The ElephantWalk procedure is complete.* Return to the **myFirstMethod** tab.
34.	Move the **ElephantWalk** procedure that is currently in the myFirstMethod tab to the bottom of the tab below the **turnToFace** procedure.
35.	**Run** the animation. *Observe how the Elephant turns to face the pond, then moves toward it a small amount.*
36.	Click and drag the **Count** statement into myFirstMethod. Select **3**. *The Count control statement allows you to repeat a procedure a specific number of times, so you do not have to keep entering the procedure into myFirstMethod repeatedly.*
37.	Drag the **ElephantWalk** procedure inside of the **Count** statement. 
38.	**Run** the animation.
39.	Revise the value in the Count statement until the Elephant walks the full distance to the pond. *Note: This may vary depending on how far your Elephant is from your pond.* Run the animation a few times. Edit the programming statements as necessary until the animation runs as you intended.

Copyright © 2014, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Part 4: Give the Animation a Unique Ending

Now that the Elephant is able to walk to the pond, you need to give the animation a surprise ending. Here are some ideas to spur your thinking:

- The Elephant jumps in the pond and takes a swim.
- The Hippo scares the Elephant away, or is nice to the Elephant.
- Another animal, such as a hyena or wolf, jumps into the scene.
- The pond floods (hint: use the **resize** procedure).

Use this opportunity to declare additional procedures if you need them.

Copyright © 2014, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.