ORACLE

Academy

Java Fundamentals

2-3

Procedures and Arguments

ORACLE
Academy

Objectives

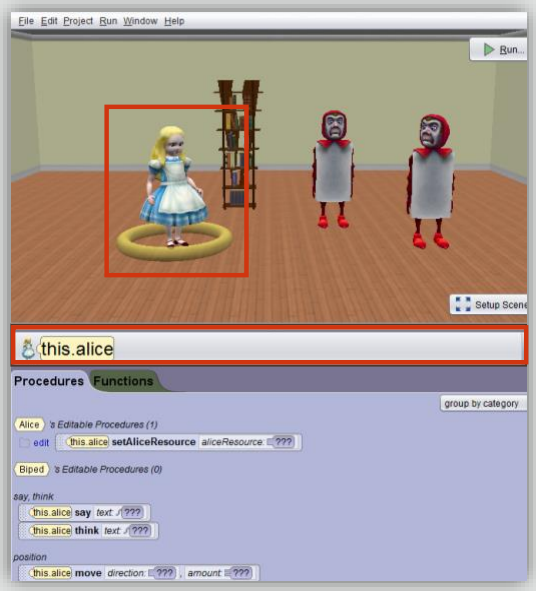
- This lesson covers the following objectives:
  - Toggle between, and describe the visual differences between, the Scene editor and the Code editor
  - Locate and describe the purpose of the methods panel and the procedures tab
  - Use procedures to move objects
  - Add Java programming procedures to the Code editor
  - Demonstrate how procedure values can be altered
  - Create programming comments
  - Reorder, edit, delete, copy, and disable programming statements
  - Test and debug an animation

Display the Code Editor

- Click Edit Code (from the scene editor) to display the Code editor
- The Code editor is where you program your animation

Select Instance

- First, select the instance that you want to program
- This ensures that you are creating a programming instruction for the correct instance
- Select an instance by clicking on the instance in the small scene window or by using the instance pull down menu below the small scene window

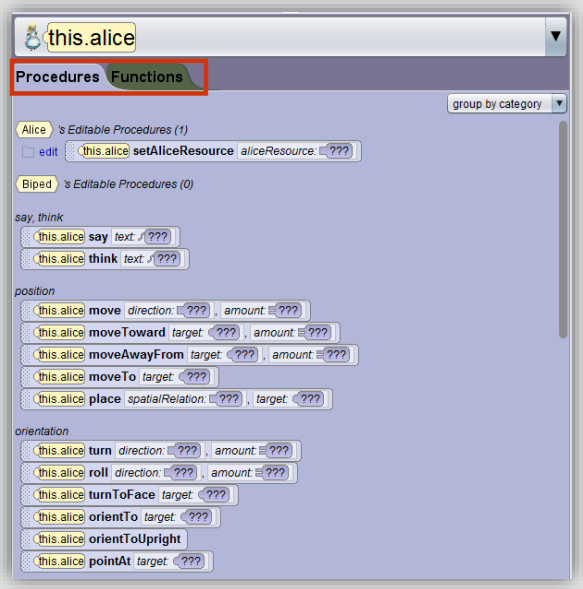


When you click on the object in the scene window always ensure that the object and not the world itself has been selected. You can check this by reading the name that is displayed in the drop down menu.

Methods Panel

The Methods Panel contains two tabs:

- Procedures:
  - All pre-defined procedures for an object
- Functions:
  - All pre-defined functions for an object

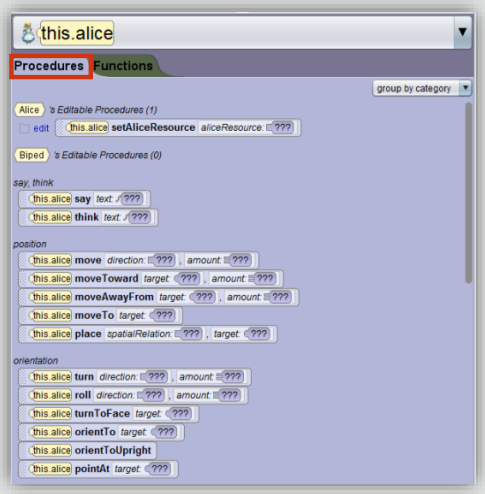


These tabs are where all of your programming statements in Alice will come from. Procedures for actions and functions for information.

Procedures Tab

- The Procedures tab displays pre-defined procedures for the selected instance, as well as procedures of your own that you define

A procedure is a piece of program code that defines how the object should execute a task. Alice 3 has a set of procedures for each class; however, users can create ("declare") new procedures.

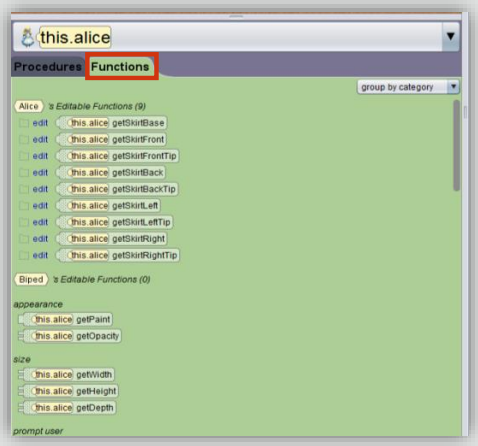


You will cover creating your own procedures later in the course. Alice 3 has lots of built in procedures that will enable you to manipulate your objects.

Functions Tab

- The Functions tab displays pre-defined functions for the selected instance, as well as functions of your own that you define

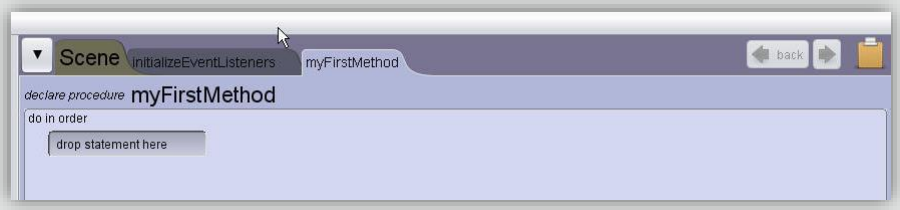
A Function computes and answers a question about an object, such as, "What is its width or height?", or "What is its distance from another object?" Alice 3 has a set of functions for each class; however, users can declare new functions.



A function is used to return information about your object. These have to be used in conjunction with procedures to create programming statements.

Code Editor Tabs

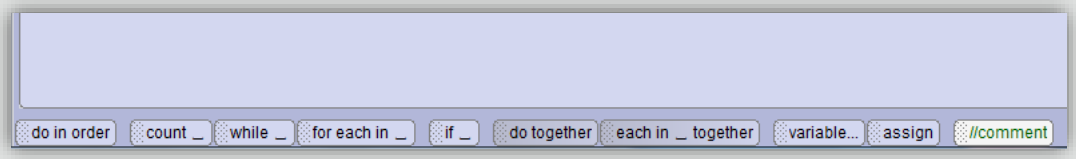
- The Class Menu displays to the left of the Scene tab
- You create your programming instructions on the myFirstMethod tab
- By default, Alice creates a Do In Order control statement in the myFirstMethod procedure
- The area labeled drop statement here is the location onto which you will place programming instructions



Alice executes code in a procedural order. The code at the top will execute first and then the next lines of code in turn till it reaches the bottom of the code.

Control Statements

- At the bottom of the myFirstMethod tab you will find the Alice 3 control statements



These control statements can be dropped into the code editor by clicking and dragging them into whatever position you want.

Object Movement

- Object movement is egocentric:
  - Objects move based on the direction they face
- An object can move in six directions:
  - Up
  - Down
  - Forward
  - Backward
  - Right
  - Left

An object as well as all of its sub-parts can move in these directions.

Examples of Movement Procedures

Procedure	Description
Move	Moves the object in any one of its six directions
Move Toward	Moves the object toward another object
Move Away From	Moves the object away from another object
Move To	Moves the object from its current position to the center point of the target object
Move and Orient To	Moves the object from its current position to the center point of the target object and adjusts the object's orientation to match the orientation of the target object
Delay	Halts an object's movement for a certain number of seconds Delay can be used to slow down the execution of an animation
Say	Creates a call-out bubble with text that makes the object appear to talk

Examples of Rotation Procedures

Procedure	Description
Turn	Rotates an object left, right, forward, or backward on its center point Left and right turn on the object's vertical axis; forward and backward turn on the object's horizontal axis
Roll	Rolls an object left or right on its center point using the object's horizontal axis

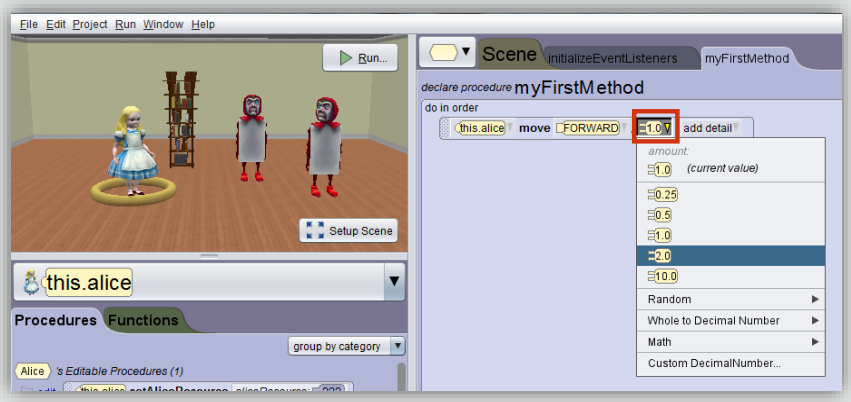
Turn and roll are the two procedures that get mixed up most when starting out with Alice. Choosing the wrong one generally creates some "interesting" animations.

Create a Programming Instruction

- From the Procedures tab, click and drag the desired procedure into myFirstMethod in the Code editor

Select and Set Argument Values

- After the programming statement is created, use the drop-down menus to set the values for each argument
- To access the argument drop-down menu, click on the down-pointing triangle to the right of the argument value



ORACLE Academy

Any statement that has been added to the code editor can have its argument values changed at any time.

Execute the Program

- Click the Run button to execute the programming instructions
- Run the program frequently to test for the desired results, and alter the values of the arguments as necessary

ORACLE Academy

As in all programming environments never create an extensive amount of code before testing that it works. You should code small animations and test them regularly.

Arguments

- Arguments are selected after the procedure is dropped onto the Code editor
- Argument types may include:
  - Object
  - Direction
  - Direction amount
  - Time duration
  - Text

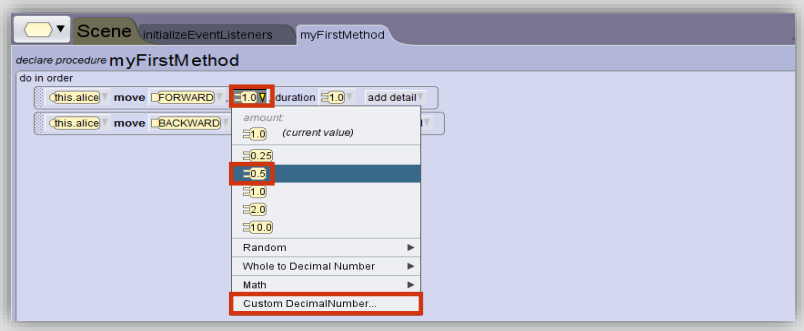
An argument is a value that the procedure uses to complete its task. A computer program uses arguments to tell it how to implement the procedure.

ORACLE Academy

Arguments in Alice teach about the use of variables within other programming environments. Their values can be changed by the code at any time.

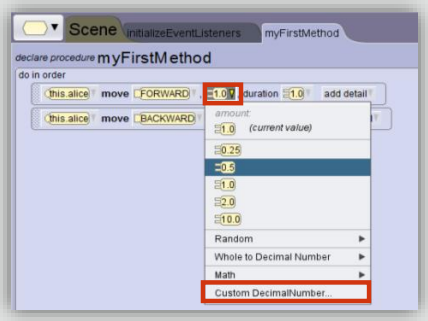
Argument Menu

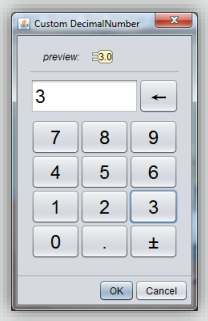
- The argument menu offers default argument values to select from
- If none of the defaults are suitable, select the Custom DecimalNumber… option so that you can specify a more accurate argument value



Steps to Edit Arguments

- Next to the argument's value, click the arrow to display the menu of values
- Select a new value
- Use the Custom DecimalNumber… option to specify a value that differs from the default list of values





ORACLE Academy

JF 2-3
Procedures and Arguments

Using these methods you should be able to give your arguments any value that is within the scope of the argument's type.

Arguments as Placeholders

- When a procedure is dropped into the Code editor, all argument values must be specified
- There will be times that you select an argument value as a temporary placeholder value that will be replaced later

As you learn to build more complex coding structures you will increasingly use placeholder values within Alice 3.

Arguments as Placeholders

- For example
  - You may want an object to move forward but you are not sure how far
  - Select a placeholder value of 2 meters, run the animation, determine that a different value is needed, and then edit the value
  - You can also specify a placeholder value that you will replace later with a function or a variable

As you learn to build more complex coding structures you will increasingly use placeholder values within Alice 3.

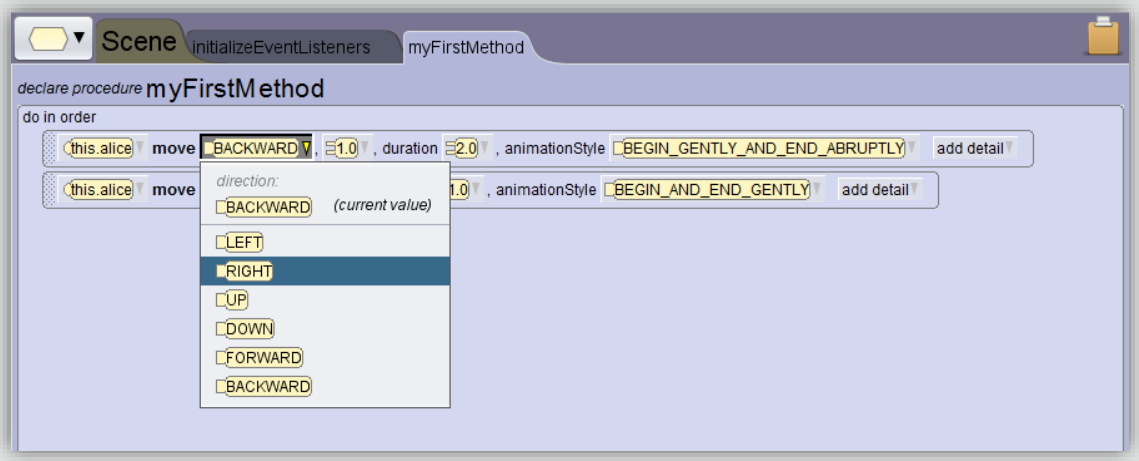
Steps to Reorder Programming Statements

- Use "**Drag and Drop**":
  - Select the handle of the programming statement
- Drag the programming statement to its new position
- Drop the programming statement into its new position by de-selecting the handle*

- ***Note**: A green position indicator line will appear to help you align the programming statement to the desired position

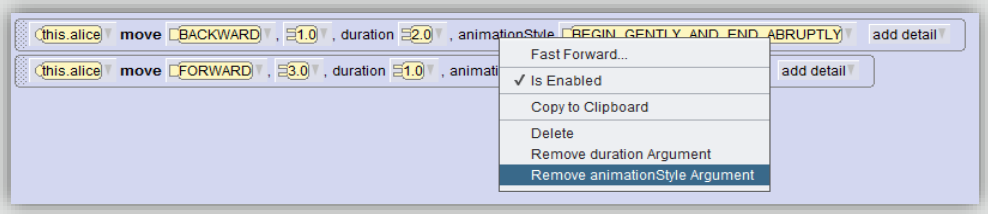
Edit Programming Statements

- Use the drop-down lists to edit the values of a programming statement

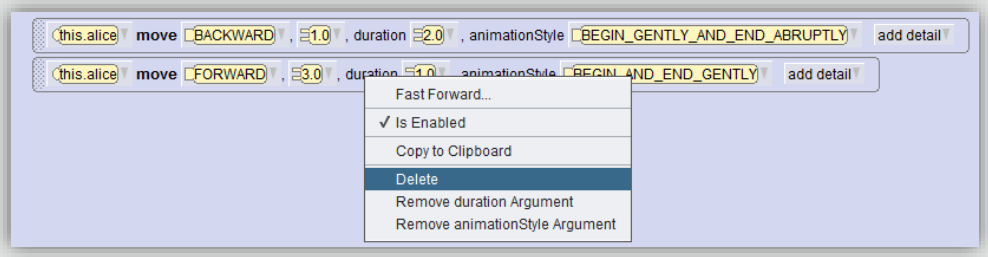


Delete Programming Statements

- Right-click programming statements to remove parts of the statement



- Or delete the entire statement



ORACLE Academy

Removing parts of the statement saves you having to delete and recreate code statements from the very beginning. This is a very useful tool within Alice that saves a lot of time.
If you delete a statement and then regret it you can always use undo (CTRL + Z) to bring it back.

Edit and Test the Program

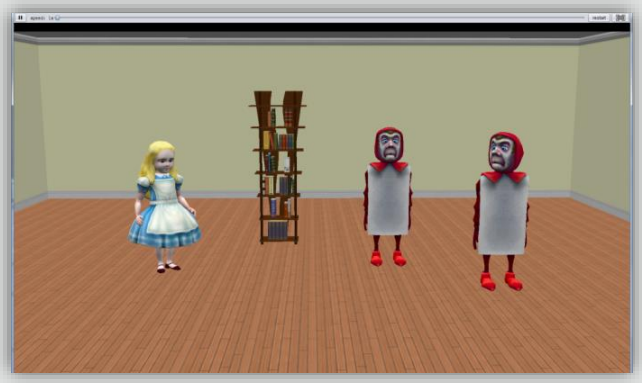
- Run the animation to test it, and edit the code as necessary
- It may take several cycles of testing and editing to get the animation to run as desired

ORACLE Academy

You need to retest the animation after every change. Something that worked perfectly well before may not work anymore because of a change you made elsewhere.

Debugging and Testing

- Debugging and Testing is the process of running the animation many times, and adjusting the control statements, procedures, and arguments after each execution
- Save often while debugging your program



Insert Temporary Programming Statements to Help with Debugging

- You can insert temporary programming statements into your code to help with debugging
- For example:
  - Imagine that you have an object that is not moving forward
- Temporarily enter a Say programming statement to announce that the object is about to move forward

Using testing statements can be very useful as it lets you get information about the flow of the code and by displaying it to screen you can evaluate the result.

Insert Temporary Programming Statements to Help with Debugging

- Test the program to see whether or not the Say programming statement executes
- If the Say statement executes, but the object does not move, this indicates one type of problem
- If both the Say and Move statements do not execute, this may indicate another type of problem

Using testing statements can be very useful as it lets you get information about the flow of the code and by displaying it to screen you can evaluate the result.

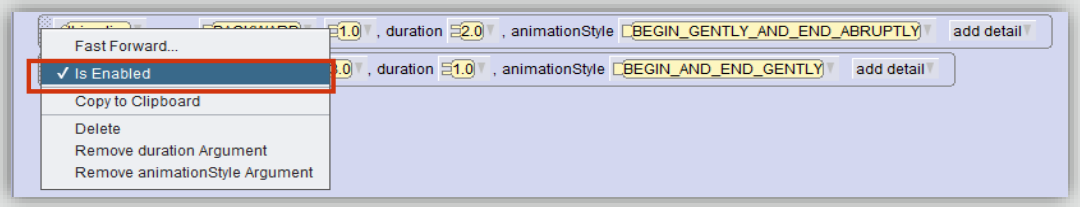
Disabling Programming Statements

- Programming statements can be disabled in the Code editor
- Disable programming statements to:
  - Help isolate portions of code during testing
  - Help you focus on programming, testing, and debugging a specific instruction

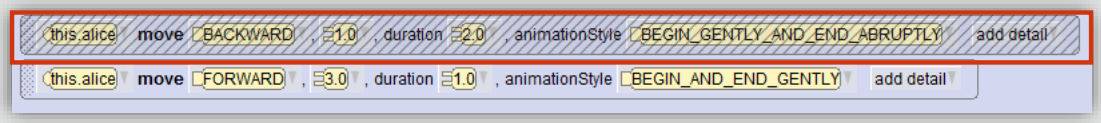
Disabling code blocks can save a lot of time so that you are not constantly re-testing code that has already been fully tested.

Steps to Disable a Programming Statement

- Right-click on the programming statement
- Unselect "Is Enabled" from the drop-down list



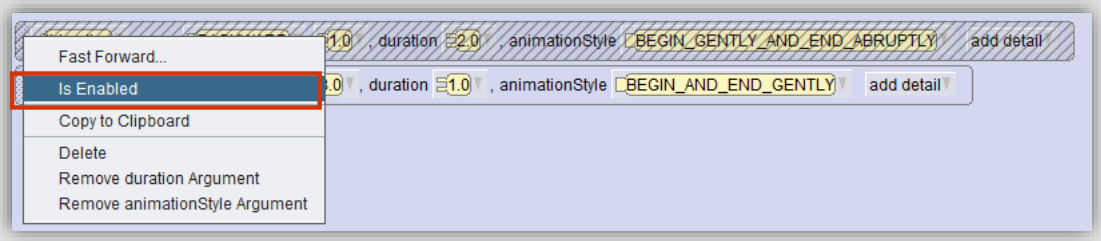
- The color of the programming statement will change to gray hash marks to show that it is disabled



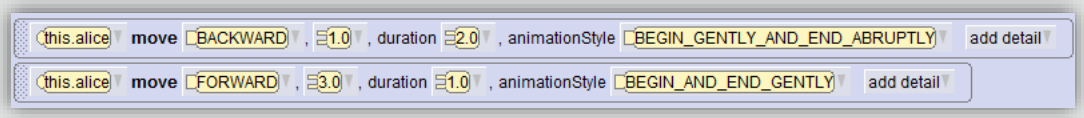
Always re-enable all code for final testing.

Steps to Re-Enable a Disabled Programming Statement

- Right-click on the programming statement that has been disabled
- Select "Is Enabled" from the drop-down list



- This re-enables the code and removes the grey lines



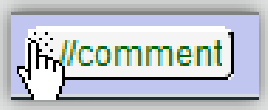
Programming Comments

- Including programming comments in an animation helps humans understand the flow of the programming
- Comments:
  - Describe the intention of the programming instructions
  - Do not affect the functionality of the program because they are ignored during its execution
  - Are typically placed above the block of programming statements that it describes
  - Are often written first, in large programs, as an outline of the programming instructions

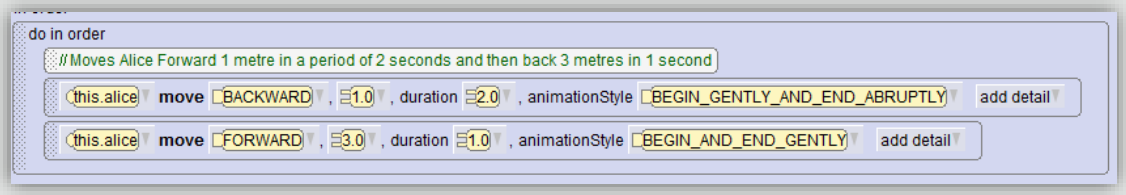
Comments are crucial to understanding how code works. It is a good habit to get into where you add comments as you code. Never leave commenting to the end, it is much easier to add comments as you code.

Steps to Enter Comments in a Program

- Drag and drop the comments tile above a code segment
- The comment tile is located at the bottom of the myFirstMethod

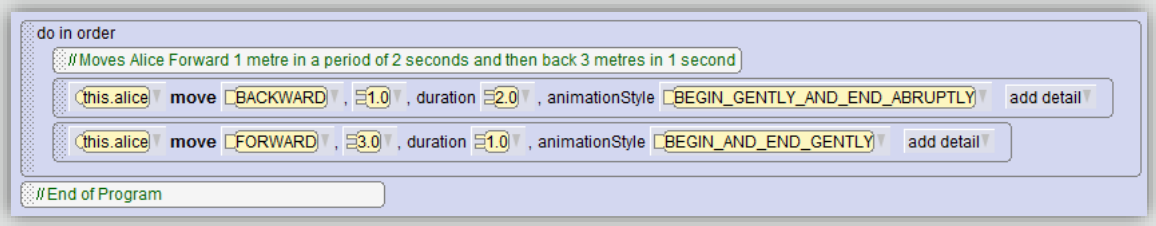


- Type comments that describe the sequence of actions in the code segment



Using Comments to Organize Your Program

- Comments can be an excellent tool for organizing the development of a program
- For large programs, create a comment that indicates the end of the program
- An "end of program" comment helps to minimize scrolling when adding programming statements to a lengthy myFirstMethod



ORACLE Academy

Comments are not compiled as part of your code so they have no affect on how your program runs. You can add as many comments as you like. They can be used to explain what code does as well as explaining the layout of your code.

Terminology

- Key terms used in this lesson included:
  - Argument
  - Code editor
  - Comments
  - Functions
  - Methods panel
  - Orientation
  - Procedure

Summary

- In this lesson, you should have learned how to:
  - Toggle between, and describe the visual differences between, the Scene editor and the Code editor
  - Locate and describe the purpose of the methods panel and the procedures tab
  - Use procedures to move objects
  - Add Java programming procedures to the Code editor
  - Demonstrate how procedure values can be altered
  - Create programming comments
  - Reorder, edit, delete, copy, and disable programming statements
  - Test and debug an animation


ORACLE
Academy