

ACTIONSRIPT 3.0 FOR ADOBE® FLASH® CS4 PROFESSIONAL

CLASSROOM IN A BOOK®

Instructor Notes

ActionScript 3.0 for Adobe® Flash® CS4 Professional Classroom in a Book®

© 2009 Adobe Systems Incorporated and its licensors. All rights reserved.

If this guide is distributed with software that includes an end user agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe Systems Incorporated. Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Any references to company names in sample files are for demonstration purposes only and are not intended to refer to any actual organization.

Adobe, the Adobe logo, ActionScript, Adobe AIR, Adobe Dreamweaver, Adobe Flash, Adobe Flash Player, Adobe Flex, Adobe Media Encoder, Adobe Pixel Bender Toolkit, and Classroom in a Book are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Apple, Mac OS, and Macintosh, are trademarks of Apple, registered in the U.S. and other countries. Microsoft, Windows, and Internet Explorer are either registered trademarks or trademarks of Microsoft Corporation in the U.S. and/or other countries. Java and JavaScript are registered trademarks of Sun Microsystems. All other trademarks are the property of their respective owners.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110-2704, USA

Notice to U.S. Government End Users. The Software and Documentation are “Commercial Items,” as that term is defined at 48 C.F.R. §2.101, consisting of “Commercial Computer Software” and “Commercial Computer Software Documentation,” as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. For U.S. Government End Users, Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

Adobe Press books are published by Peachpit, a division of Pearson Education located in Berkeley, California. For the latest on Adobe Press books, go to www.adobepress.com. To report errors, please send a note to errata@peachpit.com. For information on getting permission for reprints and excerpts, contact permissions@peachpit.com.

Writer: Chris Florio
Project Editor: Rebecca Freed
Development Editor: Robyn G. Thomas
Production Editor: Tracey Croom
Copyeditor: Wendy Katz
Technical Editor: Angela Nimer
Compositor: Kim Scott, Bumpy Design
Indexer: Jack Lewis
Cover design: Eddie Yuen
Interior design: Mimi Heft

Book:
ISBN-13: 978-0-321-57921-8
ISBN-10: 0-321-57921-6

Instructor notes:
ISBN-13: 978-0-321-61925-0
ISBN-10: 0-321-61925-9

INSTRUCTOR NOTES

Getting Started

The Adobe ActionScript 3.0 for Flash CS4 Professional Classroom in a Book™ course presents students with tips, techniques, and solutions for using the Adobe ActionScript 3.0 for Flash CS4 software. The Instructor Notes are intended to complement the information in the *Adobe ActionScript 3.0 for Flash CS4 Professional Classroom in a Book*.

The course information organization follows the sequence of instruction in each lesson in the book.

About the workbooks

It is recommended that each student in the class have an individual copy of the *Adobe ActionScript 3.0 for Flash CS4 Professional Classroom in a Book*. Students will use this book as you lead them through projects. Additionally, students can use the book as a self-paced tutorial.

Course strategy

The book contains an “Introduction to ActionScript 3.0” and 14 lessons, some of which may take a while to complete. If you’re teaching a 14-session class, you can teach approximately one lesson from this book per class with the introduction and Lesson 1 combined into a single session. Depending on the length of the class sessions, students may need to complete some of the lessons as homework assignments. All the lessons introduce new material progressively and should be approached sequentially.

If you are teaching a shorter session class, you may want to follow the lesson sequence for the earlier lessons and assign the later lessons as extra credit out of class assignments. We recommend that, at an absolute minimum, you teach Lessons 1-8, but of course completing all the lessons makes a far superior experience.

- Lessons 1–8 which introduce many essential concepts and techniques.
- Lessons 9 and 10 work together to introduce techniques for controlling sound with ActionScript and working with XML in ActionScript.
- Lesson 11 covers some of the video features in ActionScript and expands on the XML techniques introduced in Lesson 10.

- Lesson 12 covers a number of new features in Flash CS4 including the new IK tools and Pixel Bender filters as well as working with Bitmap data in ActionScript.
- Lesson 13 covers printing and sending email with ActionScript.
- Lesson 14 is a brief overview of developing Adobe AIR applications with Flash CS4 and ActionScript 3.0.

Before beginning a lesson, encourage students to browse through it, and read any sidebars to become acquainted with what will be covered.

Introduction to ActionScript 3.0

This introductory chapter provides a brief overview of the history of Flash and ActionScript as well as some information on the changes in ActionScript 3.0 from earlier versions of the language. The amount of time you spend on this material should vary depending on the experience of your students. If you are not already aware of whether they have experience with ActionScript 1.0 or ActionScript 2.0 this would be a good opportunity to discuss their levels of experience.

The introduction includes some valuable information about the various formats for viewing Flash projects online and offline. It also addresses questions your students may have about the differences between the Flash and Flex applications.

Lesson 1: Navigating the Flash Timeline

This lesson introduces the techniques for adding code to the Flash timeline to control playback; it also introduces a few essential ActionScript 3.0 programming concepts. Be sure to emphasize the pros and cons of developing on the timeline in Flash. Many students may do all their Flash and ActionScript development on the timeline while others may be learning ActionScript in order to move on to object-oriented application development with ActionScript, which does not use the timeline at all. Most Flash and ActionScript users are somewhere between never using the timeline and exclusively using the timeline. You can emphasize the value of knowing when to use the timeline and when to use code exclusively. This is a discussion that can be revisited as your students become more comfortable with ActionScript.

Placing code on the Flash timeline

Page 17

Depending on the students' experience, you may want to review the concept of keyframes and emphasize that no ActionScript can be stored on any frame in a Flash timeline unless a keyframe has been added first.

Working with labels

Pages 17–18

When demonstrating how to use labels in the Flash timeline, emphasize the value of adding labels. Using labels helps clarify the project but also makes it much easier to update the navigation in a project when changes are made to the timeline.

Looping playback with ActionScript

Page 18

When demonstrating the `gotoAndPlay()` and `gotoAndStop()` methods, emphasize their usefulness for navigating timelines in Flash. You may want to mention the number of variations that are available for these methods that allow for navigating between scenes, navigating other timelines, navigating to the next and previous frames, and other navigational tasks.

Keeping track of the number of loops

Pages 19–21

This section introduces the essential concept of storing data in a variable and modifying its value. It would be worth taking the necessary time to discuss the importance of variables in ActionScript, and talk about the various data types that variables can store. How in-depth you go into this topic at this point may depend on the experience level of your students.

Using a conditional statement to control the timeline

Pages 22–24

Conditional statements are another essential concept in this lesson that you may want to place particular emphasis on. It may be worth giving some students other examples of ways that conditional statements can be used in websites, games, e-commerce applications, training programs, and so on. Focus on uses that may be of particular interest to your students.

Updating the text on the home frame

Page 24

As you introduce the idea of manipulating text fields with ActionScript, talk about the wide range of possibilities for creating, formatting, and controlling text with code. Let students know that they will be covering many of these possibilities in coming lessons.

Questions

- 1 What is the keyboard shortcut to open the Actions panel on a Windows machine? On a Macintosh?
- 2 How do you create a label name for a frame in the timeline?
- 3 What are the rules for naming a variable?

Answers

- 1 The keyboard shortcut to open the Actions panel on a Windows machine is F9, and on a Macintosh it is Option+F9.
- 2 To add a label to a frame in the timeline, select the frame and add a keyframe to the frame if there is not one there already. Then, with the frame still selected and the Properties panel visible, type a new label name into the label area of the Properties panel, and press Enter.
- 3 You can choose any name for a variable, as long as you follow these three rules:
 - Do not use spaces in your variable names; use underscores instead.
 - Except for underscores or dashes, do not use special characters; stick to letters and numbers.
 - Do not start your names with a number.

Lesson 2: Working with Events and Functions

Understanding the event model in ActionScript 3.0 is probably the biggest step in mastering the basics of the language and being able to create rich interactive applications. Lesson 2 introduces the event model using mouse click events. Emphasize to your students the wide range of events that are possible in Flash and that by understanding the techniques covered in this lesson, they will have learned the process used to work with all events in ActionScript.

This lesson also introduces functions. You can let students know that when they are comfortable with the material in the first two lessons including variables, conditional statements, events, and function, they will have an understanding of the basic building blocks of not only the ActionScript language but also of many other programming languages.

Creating event listeners and event handling functions

Pages 30–31

An event is one of the core concepts in ActionScript. Many students grasp this quickly while others struggle with it for a considerable time. Take as much time with this section as you feel is necessary to be sure that your students grasp the concepts. Clarity on the materials in this section will help in all the coming lessons.

This material will be reinforced many times in the coming lessons.

Creating navigation with a button click

Page 32

When introducing the CLICK event, you may want to let students know that it is just one of many possible mouse events that they may use in ActionScript. This section shows how to give objects instance names in the Flash interface. Emphasize to your students the importance of instance names. Forgetting to name object instances is one of the most common mistakes for new Flash developers.

Adding functions to respond to button clicks

Page 34

Mastering the syntax of functions is critical for ActionScript students, but often it takes time to grasp the meaning of some syntax. How much time you take explaining the concepts of parameters and return types can depend on the level of your students. Sometimes it is best to just go over these concepts briefly and return to them when students have more experience.

Questions

- 1 What is a function?
- 2 How do you name a button, component, or movie clip instance in the Flash CS4 interface?

Answers

- 1 A function is a block of code that performs one or more, usually related, tasks. You create functions using the keyword `function` followed by the function name and a pair of parenthesis that may contain parameters. The tasks to be performed are contained within curly braces.
- 2 To give an object an instance name in the Flash interface, with the Properties panel visible, select the object. In the Properties panel, type a name in the instance Name field and press Enter.

Lesson 3: Creating Animation with ActionScript

This lesson lays a foundation for creating motion with ActionScript. The lesson introduces the `ENTER_FRAME` event and shows how to use it to trigger functions that animate the properties of a movie clip. The lesson also introduces the `Tween` and `Easing` classes.

Controlling MovieClip properties with ActionScript

Pages 46–47

If your students are comfortable with the design tools in Flash, then the material in this lesson can be presented as techniques that let them use ActionScript to do what they are already doing in the interface. Reinforce that there are situations when animation in the interface makes sense and when using ActionScript to create animation is preferable.

Changing a property value

Pages 47–48

The ability to change a `MovieClip` property in code allows for many creative possibilities. Give students a chance to experiment with altering `MovieClip` properties either in class or in homework assignments. You may want to show the full list of `MovieClip` properties in the Flash Help files.

Increasing or decreasing a property's value

Page 48

This section introduces operators that increment and decrement `MovieClip` properties. Depending on your students' experience, you may want to discuss operators in ActionScript in more detail and talk about some of the other operators available in ActionScript 3.0.

Using an `ENTER_FRAME` event to animate a `MovieClip` property

Pages 49–50

This section introduces the very useful `ENTER_FRAME` event. Point out that this event repeats at the current frame rate, independent of what is happening on any timelines. The `ENTER_FRAME` event is extremely useful for situations where code needs to be repeated constantly. You can point out that there are also other possible techniques (such as the `Timer` class) for repeating code.

Creating a variable to store a MovieClip reference

Page 50–51

This is the first time that students are using a variable to store a reference to a MovieClip instance. Remind them that variables can store many types of data in ActionScript. For more experienced students, you may want to give a more detailed discussion of data types at this point.

Changing MovieClip properties with buttons

Pages 51–53

This section combines many of the concepts that have been introduced so far to change a MovieClip's properties when a button CLICK event takes place. For many ActionScript beginners, becoming confident with these techniques provides an overall level of comfort with learning the language. If you have students with doubts about their ability to work with code, it may be worth taking a little extra time with these sections to make sure your students are secure with this material.

Creating animation using ActionScript tweens

Pages 53–54

The Tween class is fun to work with and is also a good opportunity to clarify how parameters work. Often, students quickly understand the purpose of the parameters of Tween instances, which leads to a grasp of working with parameters throughout the ActionScript 3.0 language.

Importing the Tween and easing classes

Page 55

For simplicity, the lessons in the book only import classes when necessary. If your students are learning ActionScript to eventually develop object-oriented applications, you may want to encourage them to import all the classes that they use.

Adding ActionScript tweens

Pages 55–58

Often, students enjoy playing with ActionScript tweens, and you may want to show them some additional easing options with ActionScript.

Questions

- 1 Describe the ActionScript syntax to import an external class into a Flash project.
- 2 When is the syntax to describe the current width of the Flash stage? The current height?
- 3 What is the range for the `alpha` property of a `MovieClip`?

Answers

- 1 To import an ActionScript class, you use the keyword `import` followed by the path to the class that you wish to import. For example to import the `Tween` class you would type:

```
import fl.transitions.Tween;
```
- 2 The width of the Flash stage can be described as `stage.stageWidth`, the current height can be described as `stage.stageHeight`.
- 3 You can set the `alpha` property of a `MovieClip` can be set to a range between 0 (completely transparent) and 1 (completely opaque).

Lesson 4: Creating ActionScript in External Files

Up to this point, all the lessons have created ActionScript in frames of the Flash timeline. In this lesson, ActionScript will be created in an external `.as` file and integrated into a Flash project. Some students may wind up pursuing OOP development in Flash and wind up working extensively or even exclusively with ActionScript files; others may choose to always work in the timeline.

Use this lesson as an opportunity to explain the options for writing ActionScript and make it clear that there is no universal wrong or right approach but instead they should try to choose the approach that works best for each individual project.

Creating an ActionScript file

Pages 64–65

Emphasize that the tools in the Flash application are not the only option for creating ActionScript files. Show them the pros and cons of using Flex Builder or other development environments.

Basic structure of an ActionScript 3.0 class file

Pages 65–69

Frequently, when students begin to grasp the way that class files work in ActionScript, their understanding of Flash expands. Take as much time as you can to fully explain the structure and use of class files. Make sure the students understand the concept of packages, constructor functions, methods, and their parameter.

This section also introduces the drawing API. If your students are more advanced you may want to show them more features of the drawing API.

Creating instances of a class file in Flash

Pages 70–73

This section demonstrates the very important technique of creating variables to store instances of custom classes. It also covers adding display objects to the stage with the `addChild()` method.

Overriding the parameters of each Ellipse instance

Pages 73–74

This is a concept that often takes new users awhile to fully understand. Explain the difference between required and optional parameters and how to set the parameters of a constructor function when creating a new instance of a class.

Turning the `makeShapes()` function on and off

Pages 74–75

This section introduces the `MOUSE_MOVE` event as well as the technique for removing an event listener.

Randomizing the color of the ellipses

Pages 75–77

This section introduces the `random` method of the `Math` class and uses it to select a random hexadecimal number.

Questions

- 1 What is the suffix for an ActionScript file?
- 2 What is the ActionScript keyword at the start of an ActionScript file that describes the path to that file?
- 3 How does ActionScript indicate that a number is a hexadecimal value?

Answers

- 1 An actionscript file is a plain text file whose file name ends with the .as suffix.
- 2 The package keyword is used at the beginning of a .as file to describe the path to that file.
- 3 ActionScript uses the characters 0x to precede a number to indicate it is a hexadecimal number.

Lesson 5: Using ActionScript and Components to Load Content

Most large Flash projects are made up of a number of SWF files in addition to supporting content and data that is loaded at runtime.

The main goal of this lesson is to integrate multiple files into a single Flash project. This lesson is particularly important for students who will be using Flash to design websites with limited interactivity but a lot of content.

Creating an instance of the List component and setting its parameters

Pages 84–86

This lesson introduces students to a number of useful UI components in Flash. Use the List component to illustrate how you can set the properties of a component in either the Component Inspector panel or with ActionScript. For users of older versions of Flash, you may want to point out that the Parameters panel is no longer available and that the Component Inspector should be used instead.

Adding an instance of the UI Loader component

Pages 86–87

The UI Loader instance is one of the most frequently used components by many web developers. This component appears again in Lesson 6, “Creating Preloaders in ActionScript 3.0.”

Examining the gallery.fla file

Page 89

Be sure that students understand the setup of the gallery file, especially the way that the thumbnails MovieClip is designed with buttons inside of it.

Adding event listeners to the thumbnails

Page 90

Be sure that students understand the process of targeting buttons that are nested in MovieClips.

Loading text from an external file

Page 91

This section introduces the `URLLoader` class and uses its methods to load text from external files.

It also shows how to set properties of a text field with `ActionScript`.

Adding a scroll bar to the text field

Pages 94–95

This section shows how to link a `UIScrollbar` component to a text field to create scrolling text. If a student's scrollbar is not working, instruct them to check whether the `multiline` property is selected for their text field in the Properties panel.

Questions

- 1 What is the parameter of a List component that stores the List's labels and data?
- 2 What ActionScript class can be used to load text or data into a file?
- 3 Name a property of a text field that can be set with ActionScript but not in the Properties panel of the Flash interface.

Answers

- 1 The `dataProvider` parameter of a list component stores the components labels and data.
- 2 The `URLLoader` class can be used to load text or data into a file.
- 3 The `backgroundColor` property of a text field can be set with ActionScript but not in the Properties panel of the Flash interface.

Lesson 6: Creating Preloaders in ActionScript 3.0

This lesson focuses on the tools for monitoring the load progress and giving the user feedback while the content of a Flash project is loading.

The lesson covers over a number of tools in the Flash testing environment as well as the basics of using a preloader to track the loading of content into Flash Player, give the user feedback while loading content, and respond appropriately when the content is fully loaded.

Tools in the testing environment

Pages 100–101

By taking advantage of the tools in the testing environment, Flash developers can work with their projects offline and still test and optimize their files to create an ideal online experience for the user.

Bandwidth Profiler

Pages 101–103

The Bandwidth Profiler is an extremely useful tool that gives the developer visual feedback on the size of the content in the frames of a Flash timeline. This feedback helps identify places that may cause potential problems downloading Flash files.

Simulate Download

Pages 103–104

The Simulate Download tool is used in conjunction with the Download Settings (View > Download Setting) in the testing environment. Choosing Simulate Download lets you experience what it would be like to stream and view a Flash project at the connection speed currently set in the Download Settings.

Creating a text field and progress bar to track loading

Pages 104–106

Preloaders can have many designs, but this lesson focuses on the most common type of preloader, which consists of a progress bar and a text field that give the user feedback on the download progress.

Inserting ActionScript to track progress of the UILoader component

Pages 106–107

The preloader in this lesson uses the built-in methods and properties of the UILoader class. Let students know that preloaders can track the loading of any content not just content loaded into a UILoader component.

Adding event listeners for PROGRESS and COMPLETE

Page 108

The PROGRESS and COMPLETE events of the UI Loader class make the process of creating preloaders much easier. Again, emphasize that they can create preloaders without these events simply by tracking the BytesLoaded and BytesTotal properties of any Flash content.

Controlling the frames of a MovieClip to reflect loading progress

Pages 112–114

This section shows a simple example of an alternative way to give the user feedback while content is loading. If there is time, have your students try to create their own preloaders with their own content.

Adding ActionScript for the loadAnimation clip

Pages 115–119

It may take time for some of your students to understand the math in steps 10 and 11. Go over it a few times with them, if they are having trouble, and make it clear to them that simple operations like this become more intuitive with experience.

Questions

- 1 What is the purpose of a preloader in Flash?
- 2 Name one way to enter the Flash testing environment.
- 3 Which user interface (UI) component is used to give visual feedback on the loading progress.

Answers

- 1 A preloader tracks the loading of content into Flash Player, to give the user feedback, when needed, on that loading, and to respond to the results of the loading.
- 2 You can enter the Flash testing environment by choosing Control > Test Movie from within Flash.
- 3 You can use the ProgressBar component to give a user visual feedback on the loading progress.

Lesson 7: Using Arrays and Loops in ActionScript 3.0

This lesson introduces two very important ActionScript tools: the Array and the for loop. Explain that these are important tools in many programming environments. Now students should think of themselves as going beyond the beginning steps in ActionScript 3.0. Assure your students that once they are comfortable working with Arrays and for loops, there are many types of functionality that will be possible in their projects. You may want to give them some examples that are relevant to their specific areas of interest.

Adding MovieClip instances to the stage from the Library

Page 126

Adding clips from the library to the stage with ActionScript is a good use of the hybrid design/programming nature of Flash.

It is not covered in the lesson but an alternative way to accomplish the technique covered in this exercise is to use the `embed` tag which is available in Flash for the first time in Flash CS4. (see Flash Help files for details).

Setting the linkage properties of a MovieClip

Pages 126–129

Setting linkage properties is a step that many beginners forget, so be sure to emphasize to students the importance of setting the linkage properties if the MovieClip needs to be added to the stage at runtime with Actionscript.

Adding a new instance of the Block class to the stage

Page 129

Emphasize that the code for making instances of clips in the library is identical to the code for making instances of other ActionScript classes.

Generating multiple instances with a for loop

Pages 129–130

Many students, who understand the basics of working with conditional statements, take a while to grasp the use of for loops.

Often, writing out a long `if` statement that does the same thing as a for loop helps clarify the syntax of a for loop and illustrate the convenience of using a for loop instead of using the longer technique of an `if` statement.

Creating the for loop

Pages 130–131

When going through the syntax of a for loop, emphasize the semicolons between statements within the parenthesis. Many new programmers make the understandable mistake of using commas between the statements.

Modifying a for loop's behavior with variables

Pages 131–133

Using variables to modify a for loop's behavior is a very important technique that opens up many possibilities especially for developing games in Flash and for cycling through objects and data.

Adding event listeners to the Block instances

Pages 133–135

Point out that each Block instance is getting its own individual event listener.

Adding the ENTER_FRAME listener

Page 135

Take this opportunity to check the student's comfort level with the ENTER_FRAME event. At this point, using EVENT_FRAME should be starting to become familiar to them.

Using an event's target property

Pages 135–136

The target property is a very useful way of having multiple objects trigger the same function. This section also introduces the concept of casting to a data type.

Creating the rotateUp() function

Pages 136–137

The rotateUp() function uses a conditional statement to limit the amount that each Block rotates. This section also introduces the new 3D properties of a MovieClip in Flash CS4.

Introducing the array

Page 140

Many beginners do not appreciate the usefulness of arrays the first time that they are exposed to them. It is sometimes beneficial to give multiple examples of using arrays in areas that the students are interested.

Using the push() method

Pages 141–142

When demonstrating the push() method, you may want to give an overview of some of the other methods of the array class that are not covered in this lesson.

Checking the properties of elements in the rArray

Pages 142–145

Point out that cycling through the properties of a series of objects in an array with a for loop is a very common and useful technique in ActionScript. Again, it may help to give additional examples.

Questions

- 1 Once a MovieClip in the library has its linkage properties set, how do you create instances of it with ActionScript? How is that instance added to the stage?
- 2 What properties of a MovieClip can you use to rotate it in three dimensions?
- 3 What do you use the keyword as used for in ActionScript 3.0?

Answers

- 1 You create a new instance of a MovieClip in the library with the keyword new just like an instance of any other class. For example, to make an instance of an object in the library with a linkage identifier of Ship you could write:

```
var ship1:Ship = new Ship
```

to add the object to the stage you would write:

```
addChild(ship1)
```
- 2 You can rotate a MovieClip in three dimensions using its rotationX, rotationY, and rotationZ properties.
- 3 You use the keyword to change or recast an object's data type.

Lesson 8: Creating a Radio Button Quiz in an ActionScript File

This lesson will probably take longer than the other lessons. It covers a range of techniques that are used for the first time when working with Radio Buttons, using a `switch` statement, communicating between an ActionScript file and the timeline, and using the `TextFormat` class.

This lesson also makes extensive use of arrays and `for` loops.

Creating a new ActionScript file

Pages 152–153

Most of the work the student will do in this lesson will be in a new ActionScript file called `RadioButtonsQuiz.as`.

Setting up the RadioButtonQuiz class

Page 153

This is a good opportunity to review the basic syntax of a class file.

Importing classes into the RadioButtonQuiz class

Page 154

This file makes use of many more classes than the ActionScript file in Lesson 4. Be sure to emphasize the necessity of importing every class that is used. Use the completed file for the lesson to show the error message that is generated when one of the `import` statements is removed or commented out.

Creating the RadioButtonQuiz class

Page 154

This section includes an important sidebar on the Sprite data type. Be sure that your students are clear on the relationship between `MovieClips` and `Sprites` and when to use each.

Reviewing the format of the quiz

Page 155

This section introduces radio buttons, which are used extensively in this lesson.

Adding variables to the RadioButtonQuiz class

Pages 156–157

This section introduces the concept of class variables as well as the public and private access modifiers.

Creating the constructor function

Pages 158–159

Review the concept of the constructor function. Point out that it is common for a constructor function to exclusively call other functions.

Creating the quizInit() function

Pages 159–160

This function creates and sets up the text field that will give the user feedback as they take the quiz. It also sets up the initial RadioButtonGroup.

Creating a RadioButtonGroup

Pages 160–161

Make sure that the relationship between RadioButtons and a RadioButtonGroup is clear to your students.

Formatting text with ActionScript

Pages 161–163

Many students who have experience with web design understand the TextFormat class by comparing it to using CSS in an HTML web page.

Creating the quiz

Pages 163–164

The three functions created in the following sections are the heart of this lesson and may take a while for the students to fully understand. Review each function as many times as possible to give students a chance to grasp the full functionality.

Creating the addQuestion() function

Page 164

The bulk of this function involves a long `switch` statement that sets up the questions and possible answers in the quiz. Explain that `switch` statements are just an alternative way to write a long conditional statement. Some developers prefer them to `if` statements, and some rarely use them.

Adding the `setRadioButtons()` method

Pages 170–171

This function uses a number of parameters and is a good opportunity for students to become more comfortable with method parameters.

This section also introduces the idea of variables with a local scope within a function.

Calling the `createRadioButton()` method

Page 172

The `createRadioButton()` method, which is created in a coming step, is called here four times to create the four radio buttons that will appear on stage for each question in the quiz. The parameters of this method are what determine the label and position and `RadioButtonGroup` of each button.

Creating the answers array

Page 173

An array is added to the `setRadioButtons()` method to store all the possible answers to the current question as well the correct answer and the question number. Storing this information in an array allows it to be passed to other places.

Adding the `createRadioButton()` method

Pages 173–175

The `createRadioButton()` method creates radio buttons using the parameters passed to it.

Checking to see if a condition is false

Pages 175–177

This section introduces the operator `!=` as well as the `removeChild()` method. These are used together to check if a set of radio buttons is onstage, and if so, remove them before adding new radio buttons for the next question in the quiz.

Creating `checkAnswer()`

Pages 177–179

This function checks if the answer the user selected from the choices in the radio button labels is the correct one. If the answer is correct it will increment the `quizCount` variable and call the `addQuestion()` method. If the answer is incorrect, it will give the user feedback in the `resultText` field.

Calling a function in the timeline from a class file

Pages 179–182

You can point out that it is fairly uncommon to call functions in the Flash timeline from an external class file, but the techniques can occasionally be very useful as in this example.

Integrating your new class into the Flash project

Page 183

Students will integrate the code written so far with the `lesson08_start.fla` file. They will do this by making an instance of the `RadioButtonQuiz` class in the `.fla` file.

Adding an instance of the `RadioButtonQuiz` class

Pages 183–184

Point out that the code to create an instance of a custom class is the same as the code to create an instance of a built-in class.

Adding the `RadioButton` to the library

Page 185

For the radio button code in the `ActionScript` file to work, the `RadioButton` component needs to be available. You make a component available by placing a copy in the library of the `lesson08_start.fla` file.

Questions

- 1 What is the main difference between a `Sprite` and a `MovieClip`?
- 2 What event takes place when a user selects one of the radio buttons in a `RadioButtonGroup`?
- 3 Name three properties of the `TextFormat` class.

Answers

- 1 A `Sprite` is similar to a `MovieClip` except that a `MovieClip` has a timeline and a `Sprite` does not.
- 2 The `CHANGE` event takes place when a user selects one of the radio buttons in a `RadioButtonGroup`.
- 3 There are many properties in the `TextFormat` class including the three covered in this lesson which were: `font`, `size`, and `color`.

Lesson 9: Controlling Sound with ActionScript

This lesson will focus on three sound related classes; the `Sound`, `SoundChannel`, and `SoundTransform` classes. These classes work together to give you control over individual sound files. This lesson uses these classes to build a simple music player application.

Adding sliders to the project

Pages 192–193

The Slider components added in this exercise will be used later in the lesson to control the volume and panning of audio files.

The Sound, SoundChannel, and SoundTransform classes

Page 193

This section introduces the relationship between the three sound-related classes. Be sure that students understand this relationship.

Creating Sound, SoundChannel, and SoundTransform instances

Page 195

This lesson uses one instance of each sound class. This section creates the variables that will store those instances.

Adding more variables

Page 196

Three more variables are needed to keep track of the song that is currently playing in the music player, as well as the current volume and pan settings.

Creating the songList array

Pages 196–197

In this lesson, an array called `songList` stores the list of songs in the music player application. In the next lesson, this project will be further developed and an external XML file will store the song list.

Setting the song titles using a for loop

Pages 197–198

A `for` loop steps through the elements of the `songList` array and places the titles of the songs into text fields. This will give students more practice using arrays and `for` loops together.

Modifying text fields with the `replace()` method

Pages 198–199

The `replace()` method is used to shorten the strings used in the title text fields. This illustrates a couple of useful techniques including manipulating text strings and evaluating values to produce the name of an object.

Making the sliders invisible until needed

Page 200

Point out to students the value of using `ActionScript` to make interface elements visible and invisible as appropriate.

Adding event listeners to the song buttons

Pages 200–201

Rather than writing a separate function for each button onstage that can choose a song, all of the buttons call the same `chooseSong()` function.

Creating the `chooseSong()` function

Page 201

The `chooseSong()` function uses a `switch` statement that takes the name of the button that was clicked to determine which song to play. Be sure that students understand the way the name property of the target is being used in this function.

Creating a `Sound` class instance and checking for existing instances

Pages 202–203

This section uses a conditional statement to check that a sound is not already playing before creating a new sound.

This step may need to be clarified for students because the conditional refers to a `SoundChannel` instance (`channel`) that has not yet been created. This will be the `SoundChannel` that the sound (`snd`) will be played through.

Creating the SoundChannel and SoundTransform instances

Pages 203–205

The SoundChannel instance will control the playing and stopping of sounds, and the SoundTransform instance will control the volume and panning controls. It is a good idea to emphasize the warning in the sidebar on page 204 regarding the range for volume settings.

Controlling the visibility of the volume and pan controls

Pages 205–206

Once a song has been chosen to play, the volume and pan sliders that were hidden in an earlier step are needed and made visible.

Adding a listener for the ID3 tags of an MP3 file

Page 207

If there is time in your class, it may be worth while to have students create their own ID3 tags using iTunes or another audio application.

Adding the Artist and Album information

Pages 209–210

This section uses the ID3 tags from the currently loaded song to populate the text fields onstage.

Adding a text format object

Pages 210–211

Now is an opportunity to review the TextFormat class. Encourage your students to experiment with their own TextFormat settings.

Adding the slider controls

Page 212

This section adds ActionScript to set the volume and pan of the currently playing song based on the value of the two sliders onstage.

Adding the volumeChange() and panChange() functions

Pages 212–213

When a slider is moved, a CHANGE event listener is used to control volume and pan settings. Be sure students understand the simple math equations that are used to translate the volume and pan settings to user-friendly numbers.

Questions

- 1 What is the ActionScript class that needs to be imported to be able to use the CHANGE event of a Slider component?
- 2 How are single line comments added in ActionScript? Multiline comments?
- 3 What is the range for the volume property of the SoundTransform class? What is the range of the pan property?

Answers

- 1 The `SliderEvent` class needs to be imported in order to use the CHANGE event of a Slider component. You import this class with the following code:

```
import fl.events.SliderEvent;
```

- 2 Single line comments in ActionScript begin with two forward slashes. Multiline comments begin with a forward slash and an asterisk and end with an asterisk followed by a forward slash as in:

```
// this code is a single line comment
```

```
/* this line and the  
lines below this  
are all  
commented out  
until the  
asterisk-and forward space here */
```

- 3 The volume property of the `SoundTransform` class has a range from 0 (silent) to 1 (full volume); values higher than 1 are valid but will overdrive the sound. The pan property has a range from -1 (left speaker) to 1 (right speaker). A pan setting of 0 is equally balanced between left and right.

Lesson 10: Working with an XML Playlist

This lesson introduces very useful techniques for using ActionScript to integrate XML data into a Flash project. Additional XML functionality is covered in Lesson 11.

Understanding the basic structure of an XML file

Pages 218–220

The amount of emphasis you place on the XML file format should depend on the experience level and priorities of your students. The brief overview of XML structure in this lesson is enough for students to be able to complete the lesson, but if your students are new to XML, it may be worth spending more time discussing the XML format and ways that it can be used with Flash and in other environments. You might point out that XML is used with everything from video applications like Adobe After Effects and Apple Final Cut Pro to database software like FileMaker.

The strong support for XML in ActionScript makes it something that many ActionScript developers frequently use in their projects.

Replacing the songList array with an XML instance

Page 222

This lesson modifies the project from Lesson 9 and replaces the array in that lesson, which stored a list of songs, with an XML file which contains a longer list of songs as well as information about the songs. Point out to students that when a list of media or data for use in a Flash project is short and consistent, then an array is a logical place to store that information, but when a list is either long or will be changed regularly, then it is much easier to store the information in an external XML file. By learning both techniques, the students can choose the best option for each project they encounter.

Loading an external playlist using the URLLoader class

Page 223

Students were already exposed to the URLLoader class in Lesson 5. This lesson uses this class to load XML data and work with it in Flash.

Responding to COMPLETE and IO_ERROR events

Pages 224–225

Emphasize the importance of checking if loaded data and content is available before executing code that makes use of it. In this lesson all the ActionScript that works with song information from the XML file is moved into the function that takes place when the XML data is completely loaded.

Accessing song title and artist from the XML data

Pages 227–229

This section introduces very useful techniques for accessing data from loaded XML files. Be sure to point out the similarities between these techniques and the way that data is accessed from ActionScript arrays.

Navigating through the song list

Pages 236–237

The ActionScript that is triggered by clicking the `more_btn` will navigate through the song list and keep track of the songs from the XML data. Make sure that students understand the logic of the `songCount` variable and how it is used with the XML data in the `moreSongs()` function.

Questions

- 1 What method of the `URLLoader` class is used to load the data in an XML file?
- 2 How can you be sure that data that is loaded with a `URLRequest` instance is available in Flash?
- 3 What is a method that creates a hyperlink in ActionScript?

Answers

- 1 The `load()` method of the `URLLoader` class is used to load data into Flash. For example you could load data from a local file called `songlist.xml` with the code:

```
xmlLoader.load(new URLRequest("songlist.xml"));
```
- 2 To be sure that data that is loaded with a `URLRequest` instance is available you could create an event listener for the `COMPLETE` event of the `URLLoader` and use the loaded data within that event handling function.
- 3 The `navigateToURL()` method creates a hyperlink in ActionScript for example:

```
navigateToURL(new URLRequest(link), "_blank");
```

Lesson 11: Using ActionScript and Components to Control Video

Flash video is a very popular technology and many developers spend much of their time developing projects that use many of the video techniques covered in this lesson. An entire volume could be dedicated to ActionScript for Flash video. This lesson introduces the basics for creating a Flash video player that can play a series of videos from an XML playlist and include captions from an XML caption file.

Adding the FLVPlayback component

Pages 243–244

For most users who work with Flash video, the FLVPlayback component is their primary tool. Let students know that there are lots of methods, properties, and events that are available for this component beyond the ones covered in this lesson.

Setting the FLVPlayback controls

Page 246

Again, stress to students the variety of ways available to customize the look and controls for an instance of the FLVPlayback component. Show them some of the online resources with information on how to use this component and customize its controls.

Adding the FLVCaptioning component

Page 254

Remind students that in order to use the FLVCaptioning component, they should use an XML file that is specifically in the Timed Text format.

Playing multiple video files from an XML playlist

Page 257

This is another opportunity to get comfortable with the XML techniques introduced in Lesson 10. XML data is again loaded with a URLLoader instance and used in Flash as a playlist, this time for the video files that will be played in the FLVPlayback component.

Creating the changeVid() function

Pages 260–261

Be sure that students are clear on the difference between the COMPLETE event of the FLVPlayback component and the COMPLETE event of the URLLoader class.

Questions

- 1 What are the two types of video compression available in the .flv file format?
- 2 What is the type of video compression used in the .f4v file format?
- 3 What property of the FLVPlayback component is used to determine which video file is played in the component?
- 4 Which property of the FLVPlayback component determines which SWF file will be used as the controls for the source video?

Answers

- 1 The FLV file format can store video compressed with either Sorenson compression or On2 VP6 compression.
- 2 The F4V file format stores video compressed using H.264 compression.
- 3 The `source` property of the `FLVPlayback` component is used to determine which video file is played in the component.
- 4 The `skin` property of the `FLVPlayback` component determines which SWF file will be used as the controls for the source video.

Lesson 12: Delving Deeper into Graphics and Animation with ActionScript

This lesson introduces a number of ActionScript techniques new to Flash CS4. Inverse Kinematics and Pixel Bender filters are two very powerful (and fun) new CS4 features. In addition, this lesson introduces the `Camera` class and `Bitmap` classes. The techniques in this lesson only scratch the surface of what is possible with all these classes, so encourage students to continue to explore them once they are comfortable with the basics.

Inverse kinematics in Flash CS 4

Page 272

Since this is an ActionScript book, there is only a brief examination of the IK tools in the Flash CS4 interface. If time allows, however, you may want to spend more time showing students how to use the Bone tool in the Tools panel to create IK systems and how to animate IK armatures in the timeline.

Switching between authortime and runtime IK

Page 274

New users of Flash CS4 often forget to switch between authortime and runtime IK in the Properties panel. Emphasize this step with students.

Using the `IKMover` class

Pages 278–279

This is the main class for creating IK animation. Most students grasp its use fairly quickly, but you may wish to have them experiment with this class in a separate file.

Adding the moveDown() function

Pages 279–282

This is the main function in this lesson, and it will be expanded on in the steps that follow.

Creating Sound and SoundChannel instances

Page 283

In Lessons 8 and 9, the concept of using multiple sound channels was introduced, but this is the first time in these lessons that two separate sounds have been used in a single project.

Accessing the user’s webcam or video camera using ActionScript

Page 285

If students do not have access to video cameras on their computers, you may want to modify this lesson to only use the alternative video file that is included.

Using the Bitmap and BitmapData classes

Pages 290–293

These two classes are very powerful and versatile, and you may want to let students know that the techniques in this lesson only use a fraction of the capabilities in these classes.

Examining the Pixel Bender Toolkit

Pages 293–295

For more experienced students, you may want to go over the code in a few of the Pixel Bender sample filters that ship with Flash. Point them to the filter writing tutorial and Pixel Bender reference file that are included with the Pixel Bender Toolkit install.

Examining the PBFilter.as file

Pages 296–297

The book covers only a few of the significant points in this included file. If time permits, you may want to go through some of the code in this file in more detail. Particularly the code that uses the FileReference class.

Manipulating a filter parameter with the Slider

Pages 301–305

The `PBFilter.as` file refers to only a few filter parameters. More experienced users could create additional sliders to add controls for other parameters or even query the loaded filter to determine its parameters.

Questions

- 1 What does IK stand for?
- 2 What ActionScript class is used to move joints in an IK armature?
- 3 What method of the `Video` class is used to place the feed from a live camera into a video object?

Answers

- 1 IK is short for Inverse Kinematics.
- 2 The `IKMover` class is used to move joints in an IK armature.
- 3 The `attachCamera()` method of the `Video` class is used to place the feed from a live camera into a video object.

Lesson 13: Printing and Sending Email with ActionScript

This lesson introduces the ActionScript techniques for sending email from Flash in two different ways, as well as how to send content from a Flash project to the user's printer.

Adding a simple email link

Pages 313–314

While this technique is initially attractive because of its simplicity, most users quickly decide that they prefer the technique for sending email from Flash using an external PHP file.

Sending email from Flash

Pages 314–315

The PHP file for this section is included in the Lessons folder. If your students have some PHP experience, you may wish to show them how they can modify this file to offer additional functionality. The lesson file is based on a copy of the PHP file

located on the author's server. If you alter the PHP file in the lesson, be sure to load it onto your own server and modify the URL that is used in the lesson.

Sending values using the URL Variables class

Pages 315–316

Make sure that students understand the difference between normal ActionScript variables and variables that are created as properties of the `URLVariables` class.

Sending URL Variables to the PHP file

Pages 317–319

The `sendToURL()` method in step 6 is the core of this section. Point out to your students that this method is useful anytime they wish to send data from a Flash project to any external URL.

Adding printing capabilities with the PrintJob class

Pages 319–320

Many new users are initially confused by the `PrintJob` class, but once they understand the relationship between the `start()`, `addPage()`, and `send()` methods, it usually becomes much clearer.

Capturing the current date and time with the Date class

Pages 320–321

This lesson uses the `Date` class in a very simple way. If time allows, you may want to show your students ways that the `Date` class can be used to format date and time information more creatively.

Questions

- 1 What is the syntax used to send email from a text field link?
- 2 What is an ActionScript method that can be used to send variables to a URL?
- 3 What does the `start()` method of the `PrintJob` class return when the user confirms the print job?
- 4 The constructor of what ActionScript class returns the current date and time?

Answers

- 1 To send email using a text field link in Flash, you type `mailto:` followed by the recipient's email address in the Link field of the Properties panel. Optional parameters can follow the address. Each parameter is preceded with a question mark as in:
`mailto:yourName@yourISP.com?subject=From Lesson 13`
- 2 You can use the ActionScript method `sendToURL()` to send variables to a URL.
- 3 When a user confirms a print job, the `start()` method returns `true`. If the user cancels the print job, the `start()` method returns `false`.
- 4 The constructor of the `Date` class returns the current date and time.

Lesson 14: Creating Adobe AIR Applications with Flash and ActionScript

This lesson gives a very brief introduction to the world of Adobe AIR development. If the number of weeks in your class does not allow for covering all the lessons in the book, then this lesson is the easiest to skip without affecting the other lessons. However, this lesson may be many of your student's first exposure to Adobe AIR, which may be valuable for them. If there is not class time for this lesson, encourage your students to go through the lesson on their own.

AIR-specific ActionScript

Page 333

When going through this section, it may be beneficial to go to the ActionScript 3.0 Language and Components Reference guide and show the various AIR-only classes that are available.

Flash publish settings for an AIR project

Pages 334–335

After discussing the publishing options for AIR files, you may want to open the resulting XML descriptor file and go through it with your students.

Creating a digital signature

Pages 337–338

If it is relevant to your students, you may want to discuss in more detail the process of purchasing a commercial digital signature. More information can be found at: www.adobe.com/devnet/air/

Listening for drag events

Page 343

This section introduces some of the very useful ActionScript functionality that is Adobe AIR specific including the drag-and-drop capabilities and the `Clipboard`, `File`, and `FileStream` classes

Questions

- 1 Name three ActionScript classes that are available for Adobe AIR only?
- 2 What ActionScript class is used to determine whether a `PrintJob` prints vector or bitmap graphics.
- 3 What is a method of the `String` class that lets you search through the characters in a string?
4. What is a method of the `FileStream` class that can read data into Flash from an external file.

Answers

- 1 There are many ActionScript classes that are available for Adobe AIR only; these include the `File`, `FileStream`, `NativeDrag`, and `NativeDragEvent` classes.
- 2 The `PrintJobOptions` class is used to determine whether a `PrintJob` prints vector or bitmap graphics.
- 3 The `indexOf()` method of the `String` class that lets you search through the characters in a string.
- 4 The `open()` method of the `FileStream` class can read data into Flash from an external file.