

LEARN

Adobe Animate CC

for Interactive Media

Adobe Certified Associate Exam Preparation

Joseph Labrecque
with Rob Schwartz



FREE SAMPLE CHAPTER

SHARE WITH OTHERS



LEARN

Adobe Animate CC

for Interactive Media

Adobe Certified Associate Exam Preparation

Joseph Labrecque
with Rob Schwartz



LEARN ADOBE ANIMATE CC FOR INTERACTIVE MEDIA
ADOBE CERTIFIED ASSOCIATE EXAM PREPARATION
Joseph Labrecque
with **Rob Schwartz**

Copyright © 2016 by Peachpit Press

Adobe Press books are published by Peachpit, a division of Pearson Education.
For the latest on Adobe Press books and videos, go to www.adobepress.com.
To report errors, please send a note to errata@peachpit.com

Adobe Press Editor: Victor Gavenda
Senior Editor, Video: Karyn Johnson
Development Editor: Stephen Nathans-Kelly
Senior Production Editor: Tracey Croom
Technical Editor: Russell Chun
Copyeditor: Liz Welch
Compositor: Kim Scott, Bumpy Design
Proofreader: Kim Wimpsett
Indexer: James Minkin
Cover & Interior Design: Mimi Heft
Cover Illustration: atthameeni, Fotolia.com

NOTICE OF RIGHTS

All rights reserved. No part of this book with video training may be reproduced or transmitted in any form by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. The images and media files provided for download are copyrighted by the authors and Peachpit. You have the non-exclusive right to use these programs and files. You may use them on one computer at a time. Adobe product screenshots reprinted with permission from Adobe Systems Incorporated. For information on getting permission for reprints and excerpts, contact permissions@peachpit.com.

NOTICE OF LIABILITY

The information in this book is distributed on an “As Is” basis without warranty. While every precaution has been taken in the preparation of the book, neither the authors nor Peachpit shall have any liability to any person or entity with respect to any loss, damage, or injury caused or alleged to be caused directly or indirectly by the instructions contained in this book or by the computer software and hardware products described in it.

TRADEMARKS

Adobe, the Adobe logo, Adobe Certified Associate, Adobe Animate, Flash, Illustrator, Photoshop, Dreamweaver, Adobe Capture, Flash Player, and Typekit are registered trademarks of Adobe Systems Incorporated in the United States and/or other countries. All other trademarks are the property of their respective owners.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Peachpit was aware of a trademark claim, the designations appear as requested by the owner of the trademark. All other product names and services identified throughout this book are used in editorial fashion only and for the benefit of such companies with no intention of infringement of the trademark. No such use, or the use of any trade name, is intended to convey endorsement or other affiliation with this book.

ISBN 13: 978-0-134-39781-8
ISBN 10: 0-134-39781-9

9 8 7 6 5 4 3 2 1

Printed and bound in the United States of America

Toward enduring remains.

This page intentionally left blank

Acknowledgments

Flash Professional is an application that has personally (and professionally!) always been a joy to use. It is such a uniquely rich and creative environment within which to build so many different types of projects...with the ability to work with animation, design, and interactive aspects—all within a single creative environment. Like many others, I began my journey with all things “Flash” under Macromedia and have continued with it during the transition to Adobe up to the present day. Flash Professional reached superstar status at one point, and now with its rebranding as Animate CC, stands poised to reclaim its crown as the premier animation and interactive application available. We are entering a new era with Animate CC, and you, the reader of this book, get to be at the forefront.

No work of this scale can be attributed to a single individual. Thanks to Rob, who has contributed his writing to this book in order to make present a great amount of design and theory knowledge. He was also instrumental in orchestrating much of the overall work toward our unified goals with this series. Thanks to Victor at Peachpit for getting this new series of books together and working with the authors at BrainBuffet and to ensure a solid outcome. Thanks also to my editor, Stephen, and my reviewer, Russell. It would be impossible to do this correctly without you all. I’d also like to express my thanks to the wider Peachpit, Adobe Press, and BrainBuffet teams...and special thanks to Ajay, Rich, Chris, and everyone else at Adobe who is working so hard to make Animate CC what it is today—and what it will become in the future!

In closing, I need to acknowledge the patience and support of my wife, Leslie, and of our daughters; Paige and Lily. It’s a hard thing to manage a business and a family while working full time, teaching, and writing...but they make it all much, much easier.

I hope you all enjoy this book.

This page intentionally left blank

About the Authors

Joseph Labrecque regularly teaches classes on Flash Professional/Animate CC, web animation, graphics production, and mobile application design for the University of Denver where he is also employed by the University of Denver as a senior interactive software engineer. In that role, Joseph specializes in the creation of expressive desktop, web, and mobile solutions. His work incorporates a strong focus on the Adobe Flash Platform alongside more general web standards initiatives involving the use of HTML5, CSS, JavaScript, and related technologies.

He is also the proprietor of Fractured Vision Media, LLC; a digital media production company, technical consultancy, and distribution vehicle for a variety of creative works. Joseph authors video courses and written works through organizations which include Adobe Press, Peachpit, Lynda.com, Train Simple, Packt, O'Reilly, BrainBuffet, and video2brain. He speaks regularly at conferences such as Adobe MAX, FITC, 360|Flex/Stack, D2WC, and for a variety of local community groups.

Joseph is an Adobe Education Leader (AEL) and Adobe Community Professional. He received the Adobe Education Impact Award in 2010 and currently serves on the AEL Advisory Board. He is also an Apache Flex Committer.

Rob Schwartz (author of online chapters 9 and 10) is an award-winning teacher (currently at Sheridan Technical College in Hollywood, FL) with over 15 years experience in technical education. Rob holds several Adobe Certified Associate certifications, and is also an Adobe Certified Instructor. As an Adobe Education Leader, Rob won the prestigious Impact Award from Adobe, and in 2010 Rob was the first winner of the Certiport Adobe Certified Associate World Championship. Find out more about Rob at his online curriculum website at brainbuffet.com.



Contents

- Getting Started x
- 1 Welcome! 3**
 - About Adobe Learn Books 4
 - Animate CC vs. Flash Professional CC 4
 - Getting Acquainted with Animate CC 6
 - Major Interface Features 9
 - Let's Get Going! 13
- 2 Design a Vector Animation 15**
 - Creating the Project Document 16
 - Drawing Vector Assets 23
 - Working with Vector Shapes 29
 - Time to Animate! 32
 - Modify Animation through Easing 36
 - Publishing the Animation for Flash Player 39
 - Challenge! 45
 - Conclusion 45
- 3 Construct an Internet Meme Image 47**
 - Starting the Project 48
 - Using External Graphics 51
 - Managing Bitmap Images 54
 - Creating Black Vector Shapes 58
 - Working with Text Elements 59
 - Generating an Image 64
 - Challenge! 67
 - Conclusion 67
- 4 Compose an Animated HTML5 Greeting 69**
 - Creating a New HTML5 Canvas Document 70
 - Laying Out the Background Elements 73
 - Drawing the Flowerpot 77
 - Editing the Flower Graphic Symbol 81
 - Animating the Flower Stalk with Inverse Kinematics 85
 - Designing and Animating the Flower Head 88
 - Create a Personalized Text Message 92
 - Using JavaScript Code to Stop the Playhead 96
 - Publishing Your Project for HTML5 Canvas 98
 - Challenge! 101
 - Conclusion 101
- 5 Generate a Promotional Video 103**
 - Video Project Setup 104
 - Creating Dynamic Backgrounds with Animated Shape Masking 106
 - Design a 3D Rotating Headshot 109
 - Animating Your Name with Motion Presets 118
 - Dynamic Animation with Variable-Width Motion Guides 121
 - Rendering and Converting the RAW Video 127
 - Video in Animate CC 130
 - Challenge! 135
 - Conclusion 135

6 Tick-Tock...Build a Virtual Clock 137

- Working with ActionScript 3.0 138
- Creating an Adobe AIR for Desktop Document 140
- Building the Basic Clock Visual Elements 144
- Designing the Clock's Animated Movie Clip Assets 149
- Using Imported Audio in an ActionScript Project 155
- A Primer on ActionScript 159
- Writing ActionScript to Control the Clock 162
- Testing an ActionScript Project 168
- Publishing the Clock for Desktop Installation 169
- Challenge! 171
- Conclusion 171

7 Develop a Mobile Quiz App 173

- Creating an Adobe AIR for Mobile Document 174
- Design the App Interface Elements 176
- Writing the Application Classes 182
- Creating a Bank of Questions 187
- App Logic and Flow 195
- Publishing the App for Distribution 202
- Challenge! 203
- Conclusion 203

8 Wrapping It Up! 205

- Evaluating Interactive Media Content 206
- Where to Go Next 211

ACA Objectives Covered 212

Glossary 217

Index 229

9 Leveling Up with Design online*

10 Working with Outsiders *

**To access your free copy of this book's Web Edition containing more than 6 hours of video, see the instructions on pp. xiii–xiv.*

Getting Started

We design and develop a number of exciting projects which consist of vector shapes, bitmap images, text, and code to help you learn the basics of interactive media creation with Adobe AnimateCC—along with other design and project management skills that you will need for a creative career in interactive media. Adobe Animate CC (formerly known as Adobe Flash Professional CC) is a powerful program for building a wide assortment of animated and interactive project types with sophisticated transitions, effects, and interactive elements. You can use Animate CC to export projects to an ever-growing array of platforms and targets including web browsers, desktop computers, and mobile devices through a variety of native application and web technologies.

About this book and video

Learn Adobe Animate CC for Interactive Media was created by a team of expert instructors, writers, and editors with years of experience in helping beginning learners get their start with the cool creative tools from Adobe Systems. Our aim is not only to teach you the basics of the art of interactive media with Animate CC, but to give you an introduction to the associated skills (like design principles and project management) that you'll need for your first job.

We've built the training around the objectives for the Interactive Media Using Adobe Animate CC Adobe Certified Associate Exam (exam title not final at press time) and if you master the topics covered in this book and video you'll be in good shape to take the exam. But even if certification isn't your goal, you'll still find this training will give you an excellent foundation for your future work in video. To that end, we've structured the material in the order that makes most sense for beginning learners (as determined by experienced classroom teachers), rather than following the more arbitrary grouping of topics in the ACA Objectives.

To aid you in your quest, we've created a unique learning system that uses video and text in partnership. You'll experience this partnership in action in the Web Edition, which lives on your Account page at peachpit.com. The Web Edition contains 6 hours of video—the heart of the training—embedded in an online eBook that supports the video training and provides background material. The eBook material is also available separately for offline reading as a printed book or an eBook in a variety of formats. The Web Edition also includes about a hundred interactive

review questions you can use to evaluate your progress. Purchase of the book in *any* format entitles you to free access to the Web Edition (instructions for accessing it follow later in this section).

Most chapters provide step-by-step instructions for creating a specific project or learning a specific technique. Other chapters acquaint you with other skills and concepts that you'll come to depend on as you use the software in your everyday work. Many chapters include several optional tasks that let you further explore the features you've already learned.

Each chapter opens with two lists of objectives. One list lays out the learning objectives: the specific tasks you'll learn in the chapter. The second list shows the ACA exam objectives that are covered in the chapter. A table at the end of the book guides you to coverage of all of the exam objectives in the book or video.

Conventions used in this book

This book uses several elements styled in ways to help you as you work through the projects.

Text that you should enter appears in bold, such as:

In the Link field in the Property inspector, type **http://www.adobe.com/animate**.

Terms that are defined in the Glossary appear in bold and in color, such as:

The **web font** that's used in the header of the page is just what the client is looking for. That's a great thing.

Links to videos that cover the topics in depth appear in the margins.

The ACA objectives covered in the chapters are called out in the margins beside the sections that address them.

Notes give additional information about a topic. The information they contain is not essential to accomplishing a task but provides a more in-depth understanding of the topic.

NOTE Notice the effect of the vanishing point, which is used to provide perspective to the artwork as it's moved through the simulated 3D environment.

 **Video 5.1**
Animating the Mask

 ACA Objective 2.1

OPERATING SYSTEM DIFFERENCES

In most cases, Animate CC works the same in both Windows and Mac OS. Minor differences exist between the two versions, mostly due to platform-specific issues. Most of these are simply differences in keyboard shortcuts, how dialogs are displayed, and how buttons are named. In most cases, screen shots were made in the Mac OS version of Animate CC and may appear somewhat differently from your own screen.

Where specific commands differ, they are noted within the text. Windows commands are listed first, followed by the Mac OS equivalent, such as Ctrl+C/Command+C. In general, the Windows Ctrl key is equivalent to the Command (or “Cmd”) key in Mac OS and the Windows Alt key is equivalent to the Option (or “Opt”) key in Mac OS.

As lessons proceed, instructions may be truncated or shortened to save space, with the assumption that you picked up the essential concepts earlier in the lesson. For example, at the beginning of a lesson you may be instructed to “press Ctrl+C/Command+C.” Later, you may be told to “copy” text or a code element. These should be considered identical instructions.

If you find you have difficulties in any particular task, review earlier steps or exercises in that lesson. In some cases if an exercise is based on concepts covered earlier, you will be referred back to the specific lesson.

Installing the software

Before you begin using *Learn Adobe Animate CC for Interactive Media*, make sure that your system is set up correctly and that you’ve installed the proper software and hardware. This material is based on the original 2015 release of Adobe Animate CC and is designed to cover the objectives of the Adobe Certified Associate Exam for that version of the software.

The Adobe Animate CC software is not included with this book; it is available only with an Adobe Creative Cloud membership which you must purchase or it must be supplied by your school or other organization. In addition to Adobe Animate CC, some lessons in this book have steps that can be performed with Adobe Media Encoder and other Adobe applications. You must install these applications from Adobe Creative Cloud onto your computer. Follow the instructions provided at helpx.adobe.com/creative-cloud/help/download-install-app.html.

ADOBE CREATIVE CLOUD DESKTOP APP

In addition to Adobe Animate CC, this training also requires the Adobe Creative Cloud desktop application, which provides a central location for managing the dozens of apps and services that are included in a Creative Cloud membership. You can use the Creative Cloud desktop application to sync and share files, manage fonts, access libraries of stock photography and design assets, and showcase and discover creative work in the design community.

The Creative Cloud desktop application is installed automatically when you download your first Creative Cloud product. If you have Adobe Application Manager installed, it auto-updates to the Creative Cloud desktop application.

If the Creative Cloud desktop application is not installed on your computer, you can download it from the Download Creative Cloud page on the Adobe website (creative.adobe.com/products/creative-cloud) or the Adobe Creative Cloud desktop apps page (adobe.com/creativecloud/catalog/desktop.html). If you are using software on classroom machines, be sure to check with your instructor before making any changes to the installed software or system configuration.

CHECKING FOR UPDATES

Adobe periodically provides updates to software. You can easily obtain these updates through the Creative Cloud. If these updates include new features that affect the content of this training or the objectives of the ACA exam in any way, we will post updated material to peachpit.com.

Accessing the free Web Edition and lesson files

Your purchase of this book in any format includes access to the corresponding Web Edition hosted on peachpit.com. The Web Edition contains the complete text of the book augmented with hours of video and interactive quizzes.

To work through the projects in this product, you will first need to download the lesson files from peachpit.com. You can download the files for individual lessons or download them all in a single file.

If you purchased an eBook from peachpit.com or adobepress.com, the Web Edition will automatically appear the Digital Purchases tab on your Account page. Click the Launch link to access the product. Continue reading to learn how to register your product to get access to the lesson files.

If you purchased an eBook from a different vendor or you bought a print book, you must register your purchase on peachpit.com:

- 1 Go to www.peachpit.com/register.
- 2 Sign in or create a new account.
- 3 Enter ISBN: **9780134397818**.
- 4 Answer the questions as proof of purchase.
- 5 The **Web Edition** will appear under the Digital Purchases tab on your Account page. Click the Launch link to access the product.

The Lesson Files can be accessed through the Registered Products tab on your Account page. Click the Access Bonus Content link below the title of your product to proceed to the download page. Click the lesson file links to download them to your computer.

Project fonts

All fonts used in these projects are either part of standard system installs or can be downloaded from Typekit, an Adobe service which is included with your Creative Cloud membership.

Additional resources

Learn Adobe Animate CC for Interactive Media is not meant to replace documentation that comes with the program or to be a comprehensive reference for every feature. For comprehensive information about program features and tutorials, refer to these resources:

Adobe Animate CC Learn & Support: helpx.adobe.com/animate is where you can find and browse Help and Support content on Adobe.com. Adobe Animate Help and Adobe Animate Support Center are accessible from the Help menu in Animate CC. Help is also available as a printable PDF document. Download the document at helpx.adobe.com/pdf/animate_reference.pdf.

Adobe Forums: forums.adobe.com/community/animate lets you tap into peer-to-peer discussions, questions, and answers on Adobe products.

Adobe Animate CC product home page: adobe.com/products/animate provides information about new features and intuitive ways to create professional layouts for print, tablets, and eBooks.

Adobe Add-ons: creative.adobe.com/addons is a central resource for finding tools, services, extensions, code samples, and more to supplement and extend your Adobe products.

Resources for educators: adobe.com/education and edex.adobe.com offer a treasure trove of information for instructors who teach classes on Adobe software at all levels.

Adobe certification

The Adobe training and certification programs are designed to help designers and other creative professionals improve and promote their product-proficiency skills. The Adobe Certified Associate (ACA) is an industry-recognized credential that demonstrates proficiency in Adobe digital skills. Whether you're just starting out in your career, looking to switch jobs or interested in preparing for success in the job market, the Adobe Certified Associate program is for you! For more information visit edex.adobe.com/aca.

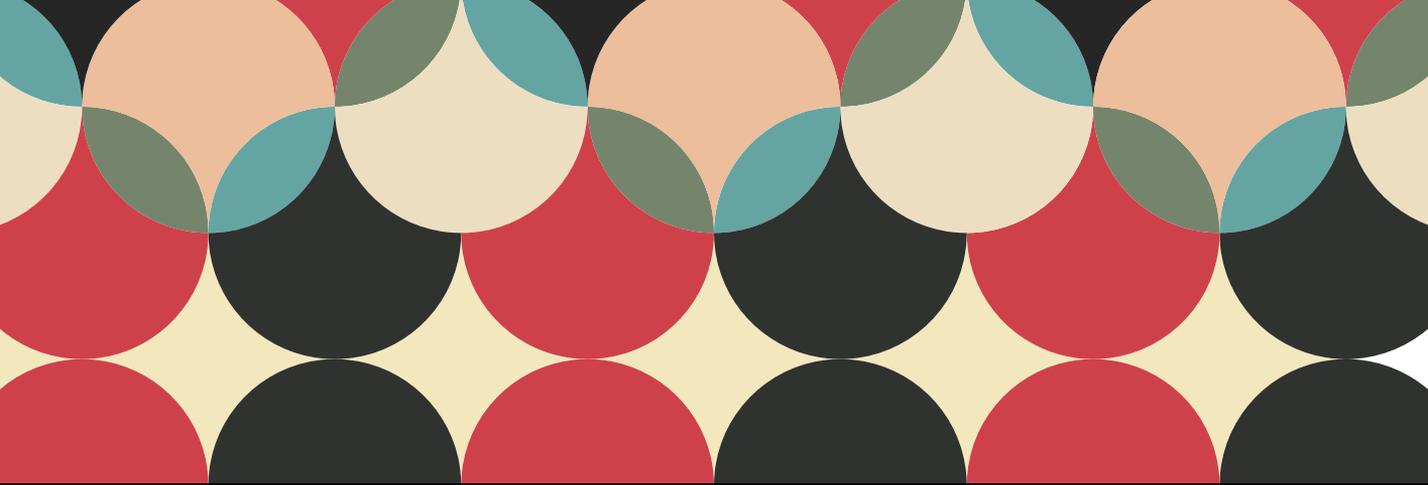
Resetting the preferences to their default settings

Animate CC lets you determine how the program looks and behaves (like tool settings and the default unit of measurement) using the extensive options in Edit > Preferences (Windows) or Animate CC > Preferences (Mac OS). To ensure that the preferences and default settings of your Adobe Animate CC program match those used in this book, you can reset your preference settings to their defaults (the Animate CC Help file calls this procedure “re-creating your preferences”). If you are using software installed on computers in a classroom, don't make any changes to the system configuration without first checking with your instructor.

To reset your preferences to their default settings, follow these steps:

- 1 Quit Adobe Animate CC.
- 2 Hold down the Ctrl + Alt + Shift keys (Windows) or Cmd + Option + Shift keys (Mac OS).
- 3 Continue to hold the keys and start Adobe Animate CC.
- 4 When the Delete Adobe Animate Preferences dialog box appears, release the keys and click OK.

The file containing your preferences will be deleted.



CHAPTER OBJECTIVES

Chapter Learning Objectives

- Work with HTML5 Canvas and WebGL publish targets
- Access the Brush Library
- Use the Paint Brush tool
- Create armatures with the Bone tool
- Work with Movie Clip symbols and instances
- Manage HTML5 Canvas publish settings

Chapter ACA Objectives

For more detail on ACA Objectives in this book, see the table on pages 212–215.

DOMAIN 2.0
UNDERSTANDING INTERACTIVE MEDIA DESIGN
2.4

DOMAIN 3.0
UNDERSTANDING ADOBE FLASH PROFESSIONAL
3.2, 3.6, 3.7

DOMAIN 4.0
CREATING INTERACTIVE MEDIA CONTENT USING
FLASH PROFESSIONAL
4.1, 4.3, 4.4, 4.5

DOMAIN 5.0
TESTING, PUBLISHING, AND EVALUATING
INTERACTIVE MEDIA ELEMENTS USING
FLASH PROFESSIONAL
5.4

CHAPTER 4

Compose an Animated HTML5 Greeting

This chapter will build on everything you’ve learned so far, but we’ll take things in a different direction from previous projects by authoring content that runs natively in the browser without the need for Flash Player. We’ll also delve deeper into aspects of the Library and symbols while introducing some new tools.

NOTE *Adobe has rebranded Flash Professional CC as Animate CC! They are essentially the exact same application—with absolutely no differences in capability aside from features introduced in the regular update schedule. All future updates to Flash Professional CC will be known by the name Animate CC. If desired, subscribers can run multiple versions of the product—branded as either Flash Professional or Animate—but I highly recommend using the latest available version of Animate CC to take advantage of all new features.*

This project involves the traditional web-based greeting card (**Figure 4.1**), but with a twist: the client wants you to design the card in such a way that it can be accessed via a web browser on a desktop, mobile device, or other similar means. In other words, since the browser-based Flash Player is widely available only on desktop browsers, in order to hit mobile targets we’ll need to deploy using a native **HTML5** target. Luckily, Adobe Animate CC includes a number of target-platform document types suited specifically to this task.

 **Video 4.1** Project Overview

NOTE

All of the videos for this book were recorded with a version of Animate CC branded as Flash Professional CC. It is important to note that everything that we demonstrate in the recordings can be applied to Animate CC—even if we refer to the application as Flash Professional. They are essentially the exact same application—with absolutely no differences in capability.

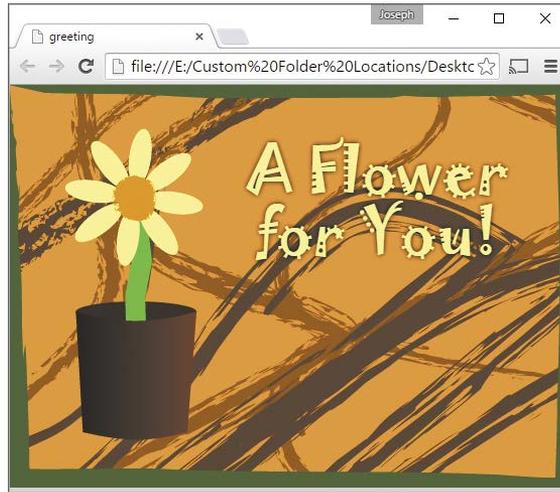


Figure 4.1 The completed web-based greeting card that you'll produce with Animate CC in this chapter

Creating a New HTML5 Canvas Document

★ ACA Objective 4.1

▶ **Video 4.2** *Creating a New HTML5 Canvas Document*

Content created in Animate CC can now be published directly to a number of modern HTML5 formats, including **Canvas** and **WebGL**. The primary document type that allows this publish target is HTML5 Canvas. When creating content using this publish target, Animate CC leverages the CreateJS JavaScript libraries to output an entire animation or interactive project for the HTML5 canvas element.

Setting up the project document

NOTE

The HTML5 canvas element is also used to display WebGL content.

The first thing you'll do is create a new document. On the Welcome screen, or via File > New in the application menu, choose HTML5 Canvas as your document type.

CREATEJS

CreateJS is a suite of modular JavaScript libraries and tools that work together or independently to enable rich interactive content on open web technologies via HTML5. See www.createjs.com to learn more.

Animate CC opens a new document, which initially will look exactly the same as the documents we've already worked with. You should note a number of differences, however:

- First, notice that in the General tab of the New Document dialog box, the name of the file is appended with the (Canvas) qualifier (**Figure 4.2**). This lets you know that you're working in that particular document type. ActionScript-based projects have no such qualifier.

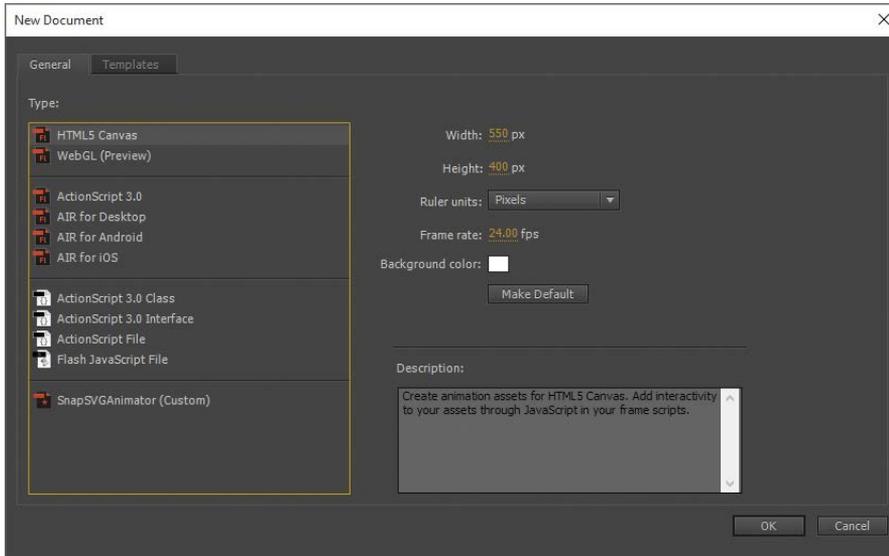


Figure 4.2 The New Document dialog box

- There are a couple of tools in the Tools panel that are disabled, including the 3D Rotation tool () and the 3D Translation tool ().
- If you select the Text tool (), you'll notice that you can only choose Dynamic Text. The other two text types are disabled.
- When viewing Document Properties, you'll see that a lot of options under the Publish area are disabled because they pertain only to ActionScript-based projects.

Otherwise, creating and animating content targeting HTML5 Canvas should be nearly identical to what you have already encountered.

NOTE

The HTML5 canvas element is a structure through which bitmap data can be drawn and changed via JavaScript. It also includes the ability to process and respond to various modes of interactivity like mouse clicks and touch events.

Preparing the Stage and timeline

Before creating any assets in the greeting card project, you'll do some work on configuring the Stage and other project fundamentals.

- 1 In the Properties panel (**Figure 4.3**), set the Stage width to 550 pixels and the height to 400 pixels. This is the normal Animate CC default, so there may not be anything to change depending on your preferences.

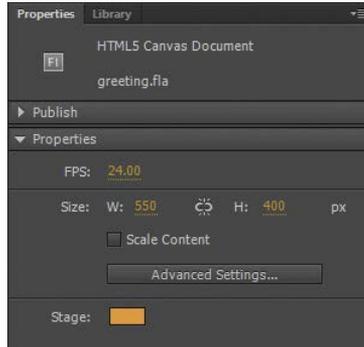


Figure 4.3 Configuring the Stage in the Properties panel

- 2 Make sure FPS is set to a value of 24 and choose a background color that suits you. I choose black most of the time, but in this project you'll be drawing a background anyhow, so it doesn't matter much.
- 3 Moving on to the timeline, rename the present layer **BG**. This is where you will draw your project background.
- 4 Use the New Layer button () to create an additional layer and name it **Frame** (**Figure 4.4**). This layer will contain assets that make up the basic background elements of your project.

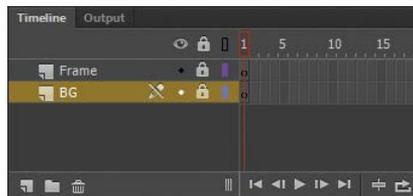


Figure 4.4 Frame layer added

- 5 Save the document in a location that makes sense to you.
Even though you're using a different document type in this project, Animate CC treats the file the same way, even saving it in the same FLA file format.

Laying Out the Background Elements

Now that you have layers for your frame and background assets, you can create these elements using the shape and drawing tools in Animate CC. You'll be using some of the vector tools you're already familiar with, but you'll also be introduced to the concept of variable-width strokes and the super-cool Paint Brush tool.

★ ACA Objective 3.2

Drawing the background

To draw the background to the greeting card project, you'll use the new Brush Library and the Paint Brush tool in Animate CC.

▶ **Video 4.3** *Drawing the Background*

- 1 To begin, lock the Frame layer to ensure that you don't draw anything in it by mistake.
- 2 Select the BG layer within the timeline and choose either the Rectangle tool () with Object Drawing Mode enabled or the Rectangle Primitive tool ().
You're now ready to draw the background.
- 3 Click and drag from the top left of the Stage to the bottom right to draw a shape exactly the same size as the Stage: 550x400.
- 4 You may need to make some adjustments to the size and position of your shape, depending on your snap settings and how steady a hand you have. Do this from the Properties panel with the object selected.
- 5 If you haven't done so already, choose a background color for your shape. I'm using #FF9933 in the example shown in **Figure 4.5**, but you can choose whatever you like.



Figure 4.5 I'm using a black background for my shape.

NOTE

You can always turn off either Stroke or Color by choosing this option for either color value.

- 6 You don't need any Stroke color for this project, so choose None for the Stroke color. It's represented in the color picker as the white block with the red diagonal line running through it (Figure 4.6).

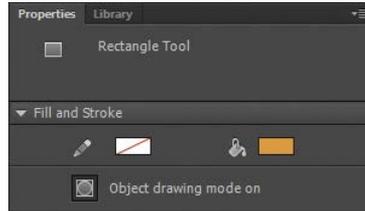


Figure 4.6 No stroke color as represented in the Properties panel

Decorating with the Paint Brush tool

Video 4.4
Decorating with
Art Brushes

Now for some freeform fun! The Paint Brush tool () is a new tool in Animate CC that lets you draw out strokes with vector art attached to them. This allows for massive creative possibilities, whether you're using built-in brushes from the Brush Library or using custom brushes you created with the Adobe Capture mobile app and shared with Animate CC via the CC Libraries panel.

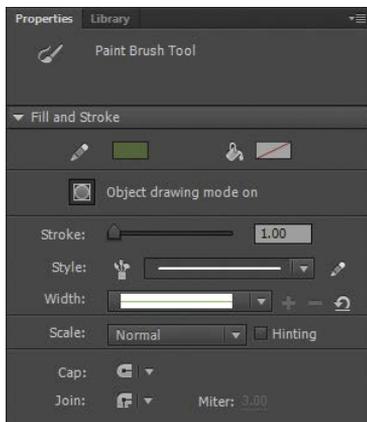


Figure 4.7 The Paint Brush tool Properties panel

- 1 To begin, choose the Paint Brush tool from the Tools panel. The Paint Brush tool Properties panel opens (Figure 4.7).
- 2 You'll be drawing these decorative assets in the BG layer, so be sure to enable Object Drawing Mode to prevent shape collisions.
- 3 Choose Window > Brush Library from the application menu to open the Brush Library.
- 4 Find one you'd like to use and select it.

In the Brush Library you'll see a number of categories, which contain a wide variety of prebuilt vector art brushes.

It should now show up in the Style area of your Properties panel. You can see in the next figure that I've chosen the brush named Chalk – Scribble from the Artistic > Chalk Charcoal category within the Brush Library (Figure 4.8).

Now it's time to get creative!

NOTE You can also access the Brush Library directly from the Properties panel by clicking its icon.

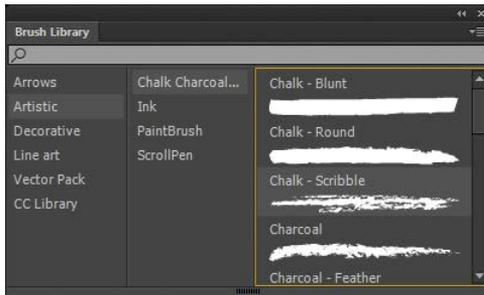


Figure 4.8 The Brush Library

- 5 Using the Paint Brush tool, click and drag around the Stage to generate free-form vector artwork. Don't make your art too busy...but don't hold back too much either. Make it interesting.
- 6 Select a Shape color. In **Figure 4.9**, I've chosen to use something a few steps darker than my background (#A35A12), but there's no reason you couldn't use a number of variations in color or even stroke width. You can control all these properties from the Properties panel with the Paint Brush tool selected.

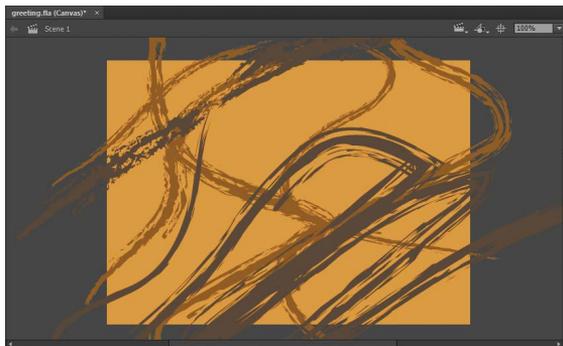


Figure 4.9 The shape color I've chosen

- 7 Once you're happy with the results, lock the BG layer and unlock the Frame layer for the next step in this project.

EDITING VECTOR ART BRUSHES

You can also edit existing brushes by clicking the Edit Stroke Style icon () from the Properties panel with the Paint Brush Tool selected. This allows even greater variation in how the vector art is mapped to your strokes.

Drawing the frame with variable-width strokes

 **Video 4.5** Making a Frame

Using variable-width strokes in your project is an easy way to impart a bit of controlled chaos to simple strokes. Animate CC will, in place of a linear stroke, impart a stroke whose width varies at certain points. The best way to grasp the concept is to use it. You'll create a nice frame for this project and apply a variable-width profile to the frame stroke.

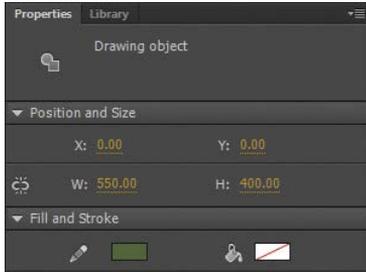


Figure 4.10 Drawing Object Properties

- 1 Select the Frame layer so that you can draw the object. You may want to ensure all other layers are locked before proceeding.
- 2 Choose the Rectangle or Rectangle Primitive tool and draw a vector object the exact size of your Stage: 550px by 400px. Use whatever color you like for the stroke (in **Figure 4.10**, I'm using #336600), and choose a transparent fill.
- 3 With the object selected, choose a profile from the drop-down in the Properties panel. If something doesn't look right, select the Width tool from the Tools panel and tweak existing width points or even create new ones.
- 4 For an additional creative flourish, apply a vector art brush to the variable-width stroke as well. In **Figure 4.11**, I'm using Artistic > Chalk Charcoal > Chalk – Round from the Brush Library.

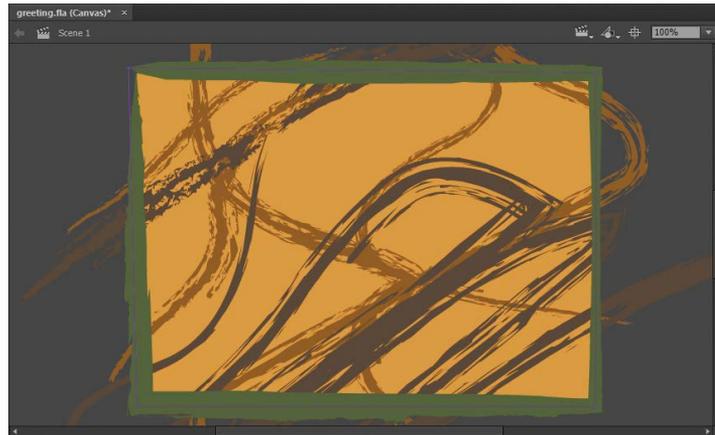


Figure 4.11 A little extra creative flourish added with a vector art brush

GROUPING THE LAYERS WITHIN A FOLDER

One of the best things you can do to keep things organized in your timeline is to group layers. You can accomplish this by creating a layer folder for specific sets of layers.

 **Video 4.6** Grouping Layers

- 1 Click the New Folder button below the layer stack on the bottom left of the timeline to create a folder to contain your static background elements.
- 2 Double-click the folder name and change it to **Background**.
- 3 Drag both the BG and Frame layers into the layer folder you just created (**Figure 4.12**).

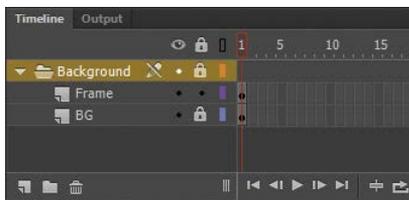


Figure 4.12 BG and Frame layers added to the new layer folder

With the folder created, you can now control the lock status, visibility, and out-line mode for all layers from the folder itself. You can also collapse and expand the folder by using the small triangle beside the folder icon.

Drawing the Flowerpot

Now that you've defined the static background elements for your greeting, you can begin work on the more complex, animated objects.

 **Video 4.7** Drawing the Flowerpot

Creating and modifying a vector rectangle shape

You'll begin by drawing a simple flowerpot.

- 1 Create a new layer and name it **Flower**.
- 2 With this layer selected, choose the Rectangle tool (). Don't select the Rectangle Primitive tool because in order for the shape to appear similar to the form of a flowerpot, you'll need to manipulate the shape.

Video 4.8 *Shaping with the Selection Tool*



Figure 4.13 The new square added on the Stage

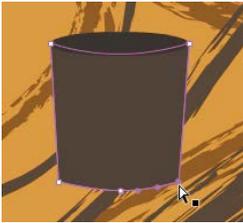


Figure 4.14 Choose a dark brown for the inside of the flowerpot.

- 3 Draw a square about 120px by 120px (**Figure 4.13**). It doesn't have to be exact, and the color choices don't matter at this point. You don't have to include any stroke for this square.
- 4 Choose the Selection tool ().
- 5 With nothing selected, hover the cursor along the top edge of your rectangle shape. Notice how the cursor changes to include a small arc. This indicates that you can bend the paths that define your shape by pulling and pushing the edges.
- 6 Pull down slightly on the top of the square to form a nice dip. This is the top of your flowerpot.
- 7 You need to create the inside of the flowerpot as well. Choose the Oval tool () and draw an oval that matches the arc and size of the flowerpot top. The color you use should be a dark brown (#4B2500) since this portion of the flowerpot will be in shadow (**Figure 4.14**). Don't apply any stroke to this object.
- 8 Using the Subselection tool (), select the bottom-left anchor point of your shape.
- 9 With the point selected, use the arrow keys on your keyboard to nudge the point over a few pixels, creating a taper in your flowerpot. Perform a similar action on the other side.
- 10 To finish the flowerpot shape, choose the Selection tool again and pull down the bottom of the shape slightly to give a little more perspective to the flowerpot, very similar to the modification you made to the top.

Applying colors and gradients

Video 4.9 *Using Gradients*

Gradients in Animate CC always start as either centered radial **gradients** or horizontal linear gradients. To adjust these gradients, you use the Gradient Transform tool (), which you can access by clicking and holding the Free Transform tool icon () in the Tools panel.

The Gradient Transform tool lets you recenter, rotate, or stretch a gradient. It's very similar to the Free Transform tool, but it affects only how the gradient is displayed.

- 1 Use the Selection tool to select the flowerpot shape.

- 2 Open the Swatches panel and choose “Linear gradient” from the color type drop-down (Figure 4.15).
By default, you’ll see a gradient that runs from pure white to complete black.
- 3 Change this to run from a slightly lighter brown (#824000) to a slightly darker brown (#582C00). Gradients often look best when there’s a subtle change across the gradient.

As you adjust your linear gradient definition in the Color panel, your shape color will update to reflect the changes. If you want to preserve this color definition as a reusable swatch, click the Add to Swatches button to make it selectable from any Swatches panel within Animate CC.

If desired, you can use the Gradient Transform tool to adjust how the linear gradient is applied to the flowerpot shape. When an object with a gradient applied is selected using this tool, you can adjust the rotation, center point, width, and more.

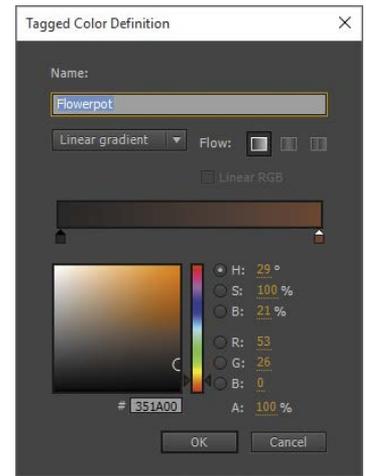


Figure 4.15 The settings you choose in the Tagged Color Definition dialog box should match the ones shown here.

ABOUT TAGGED SWATCHES

A recent addition to Animate CC is the concept of tagged swatches. This allows you to choose a swatch from the Swatches panel and transform it into a tagged swatch by clicking the Convert to a Tagged Swatch icon () below the swatches grid. Tagged swatches differ from other swatches in that a small white tag appears in the lower right of a tagged swatch.

When you have a number of shapes using the tagged swatch, you can easily update the swatch and the changes will automatically cascade across all objects using that tagged swatch.

Convert to a Graphic symbol

Symbols, at their most basic level, serve as containers for artwork. They come in three varieties, each with unique characteristics that allow them to be used for animation and interaction. Keep in mind that although Animate CC refers to “converting” elements to symbols, you’re really wrapping the existing elements into symbols. Using symbols will allow nearly any graphical element to become a button, be animated in specific ways, or even be manipulated by code.

★ ACA Objective 3.7

 **Video 4.10**
Creating a Graphic Symbol

Creating a symbol from an existing set of elements is fairly simple. Here's how.

- 1 Select everything on the Flower layer of the timeline and choose **Modify > Convert to Symbol** from the application menu.

Once the **Convert to Symbol** dialog box appears, you'll be given the option to type in a name for the symbol, select the symbol type, and specify a registration point alignment (**Figure 4.16**). There's also an area to access advanced options that are used primarily for linkage as a class within ActionScript.

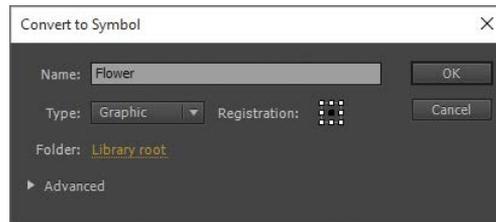


Figure 4.16 The **Convert to Symbol** dialog box

TIP

It's a good practice to always name your symbols with no spaces and no special characters, and beginning with an uppercase letter. This has to do with how Animate CC views symbols, and it will simplify the process as you explore more advanced topics.

- 2 Name the new symbol **Flower** and choose **Graphic** from the **Type** selection box. Click **OK**.

Once you click **OK**, the symbol is created and stored within the project Library. Additionally, an instance of that symbol remains on the Stage, exactly where the conversion occurred.

Types of symbols within Animate CC

You can to create three different types of symbols within a Animate CC project: **Movie Clip**, **Graphic**, and **Button**. In this project, you've created a **Graphic** symbol. The choice of which symbol type to use isn't always clear, but learning more about each of these symbol types can help you choose the one that will work within your interactive media application.

MOVIE CLIP SYMBOLS

A **Movie Clip** symbol has a timeline that behaves like—but is completely independent of—the main Stage and is the primary symbol for creating custom functionality within Flash. **Movie Clip** symbols can be controlled with **ActionScript** or **JavaScript** as well as through timeline manipulation, making them an extremely flexible tool for customizing an interface.

GRAPHIC SYMBOLS

Graphic symbols are fairly simple to create, and the operation of the symbol timeline is exactly the same as that of the main timeline. The main purpose of the Graphic symbol is to create reusable graphical elements and simple animations within an efficient package.

BUTTON SYMBOLS

Button symbols provide you with a quick and relatively simple way to create a standard interactive media button. These buttons can be created out of nearly any type of artwork, including other symbol instances, and can support a static or normal state, a rollover state, a down state, and a hit state. You construct the artwork for these states using keyframes, but rather than relying on frame numbers, the interface for construction of a button symbol uses Up, Over, Down, and Hit to identify where the keyframes for these states should be placed (**Figure 4.17**).

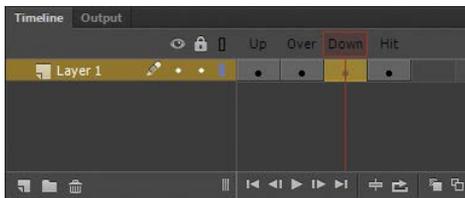


Figure 4.17 Creating button symbols in the timeline

The hit state for a Button symbol is not visible, but the artwork in the keyframe defines the clickable area of the Button symbol. If a Button symbol has no hit state, the current keyframe artwork is used to determine the clickable or live area of the button.

NOTE

If a Button symbol has only a hit state, it is an invisible button that will show up in light blue during author-time but will be invisible at runtime.

Editing the Flower Graphic Symbol

Symbols provide Animate CC projects with reusable elements that can be placed on the Stage as instances multiple times without re-creating artwork. This approach can drastically reduce file sizes and can also simplify editing a layout by minimizing the number of graphic elements required to produce an interactive media project.

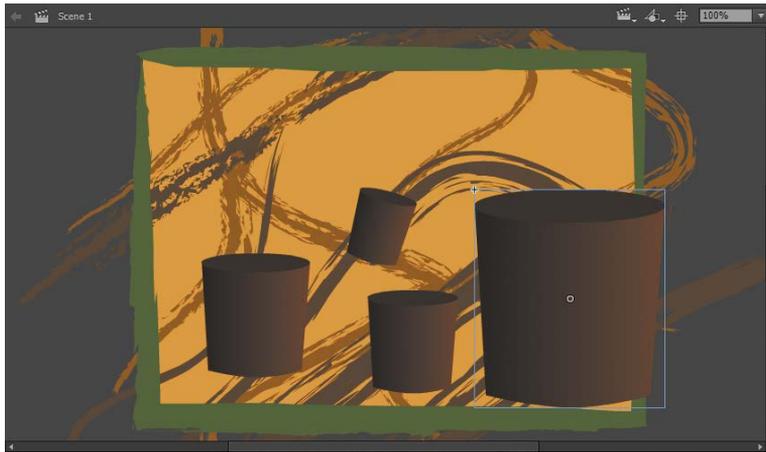
Explaining symbols and instances

Video 4.11
*Editing Symbols
and Symbol
Instances*

When a symbol is created, it resides in the project Library. The symbol acts as a blueprint to instruct Animate CC on how to construct an instance of that symbol.

Instances are representations of the symbol that exist on the Stage and timeline. When you drag a symbol from the Library to the Stage, you're creating an instance of that symbol. You can have multiple instances of a single symbol on the Stage (**Figure 4.18**). If you edit the symbol, all the instances will change as well.

Figure 4.18 You can have multiple instances of a symbol on the Stage.



This feature can be effective for making changes globally within a Animate CC project. However, it can also be frustrating if you want each instance to be slightly different.

Fortunately, symbol instances can be more than exact copies of the original symbol definition in the Library panel. You can modify the individual properties of each instance using the Properties panel or tools such as the Transformation tool. These property changes affect only the selected instance, not the original symbol definition.

Embedding additional symbols and elements within a symbol definition can also provide a means of customizing each instance using code. A common application is to include a dynamic text field within a symbol definition. With some simple programming, you can alter the content of this text field at the instance level. The nesting can get a bit complicated, but having only a single element in the Library can make it worth the trouble if you ever need to make global changes.

Differences between Movie Clip symbols and Graphic symbols

Movie Clip and Graphic symbols look very similar within the Animate CC authoring environment, but big differences between them exist. The first has to do with the synchronization between the symbol timeline and the main movie timeline.

A Graphic symbol is linked to the main timeline, whereas a Movie Clip symbol is an independent object whose timeline runs independently of the main timeline. If the main timeline is changing frames, so is the Graphic symbol. If the main timeline is paused, the Graphic symbol (which is linked frame by frame) is paused too. The Movie Clip timeline isn't bound to the main timeline and runs completely independently.

Another difference involves the use of ActionScript and JavaScript. Using code, you can access and control Movie Clip instances through the instance name assigned in the Properties panel (Figure 4.19). The Graphic symbol has no option to assign an instance name and is not accessible via code as an object.

A performance difference exists as well. Graphic symbols require less processing. Even with this advantage, many interactive media developers make nearly everything Movie Clips, just to increase the flexibility of the object, though animators often prefer Graphic symbols for the ability to specify the presented frame being shown at any time via the Properties panel.

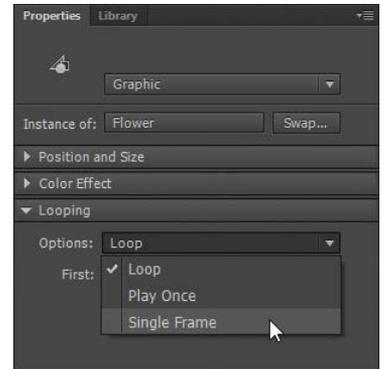


Figure 4.19 The Properties panel

Editing the flower symbol

You can edit symbols by double-clicking either any instance of the symbol on the Stage or the symbol within the Library panel.

- 1 Double-click the symbol instance you created to edit it in place (Figure 4.20).

When the symbol opens, notice that a breadcrumb trail showing the hierarchy of the object shows up in the top-left corner of the Stage. You can leave the editing mode by clicking the scene name in this breadcrumb trail.

 **Video 4.12**
Preparing the Symbol

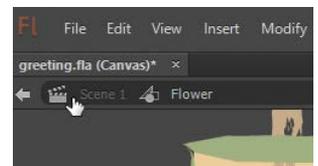


Figure 4.20 Double-click to edit in place.

Now that you're operating within the symbol, notice that the symbol itself has its own timeline. In an Animate document, each symbol, no matter the type, has its own timeline. This allows for a great amount of creative possibilities.

The next thing you'll do is to manage this internal timeline a bit better. Because you already have a number of objects within this symbol, you can place each of them on its own layer.

NOTE *It doesn't matter if you double-click the symbol within the Library panel or use the edit-in-place option of double-clicking an instance on the Stage. You're still editing the base symbol—not the individual instance. This means that any changes you make will be made to all instances of that symbol. This doesn't matter much in this case, since you'll be using only one instance of this symbol.*

- 2 Select all the objects on the Stage either by dragging a selection rectangle around them with the Selection tool, or by giving the Stage focus and using the Select All shortcut, Cmd+A (Mac OS) or Ctrl+A (Windows).
- 3 With all of your elements selected, right-click them and choose Distribute to Layers from the drop-down menu that appears (**Figure 4.21**).

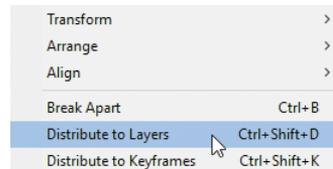


Figure 4.21 Choosing Distribute to Layers

This takes each object and creates a new layer for it and then empties the previous layer of all selected content. The original layer is empty, as indicated by a blank keyframe.

- 4 Delete the layer by clicking the Delete Layer icon () beneath the layer stack in the timeline.

Both the flowerpot shape and the darker, inner pot shape should now be on separate layers.

- 5 Rename main pot shape **pot**, and rename the layer representing the inner, backside of the flowerpot **potback**. Lock these layers before moving on.

Drawing a flower stalk

For the stalk of the flower, you'll create a simple shape with the Brush tool so that it isn't completely straight and appears a bit more natural. The Brush tool is a free-form tool that draws shapes composed of fill alone.

 **Video 4.13**
Drawing the Stalk

- 1 Create a new layer and name it **Stalk**.
- 2 Place this layer between the pot and potback layers by clicking and dragging the new layer into position (**Figure 4.22**).

Now the order from top to bottom should be pot, Stalk, and potback.

- 3 Select the Brush tool () and look at the Properties panel.
- 4 Choose a nice green color (#32CC32) for your fill and check the Brush Shape area of the panel to see the brush size and shape options. Select a round brush and bump the size way up for a thick stalk.
- 5 Draw a stalk beginning behind the pot and extending up. If you like, you can draw a straight shape by holding down the Shift key as you draw, though it may look more natural if you freehand it. The stalk, when complete, should measure about 230px in height (**Figure 4.23**).

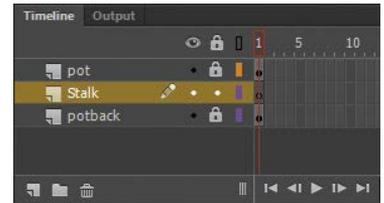


Figure 4.22 Place the Stalk layer in the middle of the layer stack.

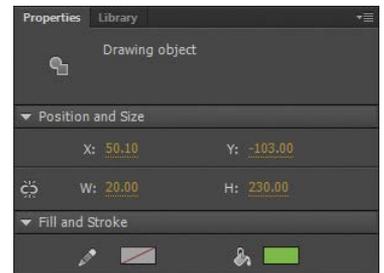


Figure 4.23 Set stalk parameters in the Drawing object Properties

Animating the Flower Stalk with Inverse Kinematics

The Bone tool has been absent from Animate CC ever since the application was rewritten for 64-bit systems with the introduction of Animate CC CC. Today, however, the Bone tool, the Bind tool, **armature** layers, and everything else in the Animate CC inverse kinematics system are back.

★ *ACA Objective 4.5*

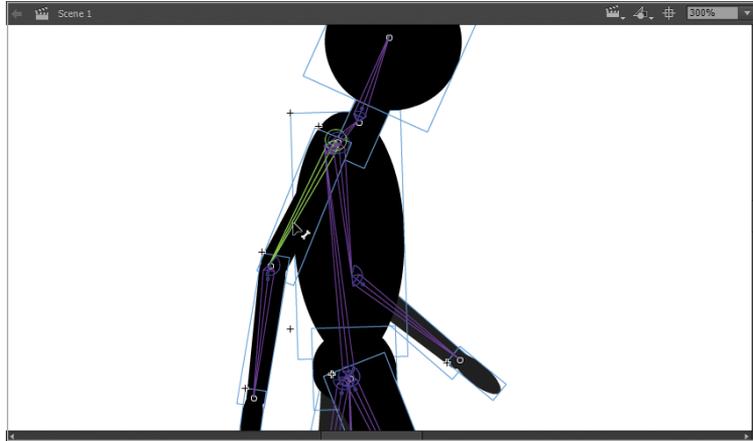
About inverse kinematics in Animate CC

Inverse kinematics allow you to animate objects in Animate CC using a series of bones chained into linear or branched armatures in parent-child relationships. This means that when one bone moves during an animation, that connected bones will

also move in relation to it. Inverse kinematics is a unique way of animating content to create natural motion.

To animate using inverse kinematics, specify the start and end positions of bones on the timeline. These positions are known as “poses” within Animate CC. The armature will be automatically adjusted to tween positions of the bones in the armature between poses (**Figure 4.24**).

Figure 4.24 Animating using reverse kinematics



You can apply inverse kinematics within Animate CC in one of two ways: by drawing bones across a single shape or through the construction of an armature composed of multiple Movie Clip symbol instances. In this project, you’ll create a shape-based armature from the flower stalk you just created.

Animating the flower stalk using an armature layer

Video 4.14 Using the Bone Tool

Your next step is to animate the stalk growing up and out of the flowerpot. To do this, you’ll take advantage of the armature you’ve already created. You don’t need to specify any sort of tween when working with armatures; Animate CC will handle all this automatically once different poses are defined.

- 1 First, you’ll need to extend your frame span to frame 65. To do so, move the playhead to frame 65, select the frame, and choose Insert > Timeline > Frame from the application menu. These frames will provide enough time to nicely animate the growth of your stalk. Bring the playhead back to frame 1 when finished.

- 2 Select the Bone tool from the Tools panel. You'll use this tool to define the bones within your armature (Figure 4.25).



Figure 4.25 Defining the bones

- 3 To define a bone, click and drag a short way. Draw subsequent bones across the shape by clicking the end point of the previous bone and dragging out a new segment. Draw 3–5 bone segments across the stalk.

Notice how the layer color is now yellow and the icon for the layers is a small figure running? These changes indicate that this is now an armature layer (Figure 4.26).

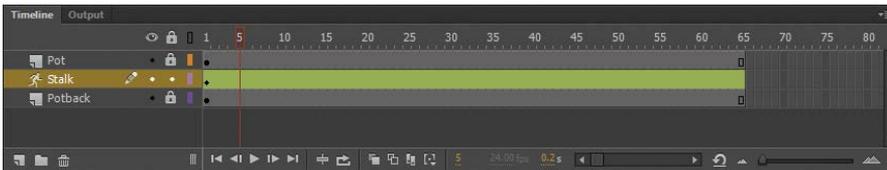


Figure 4.26 The yellow layer color indicated that Stalk is now an armature layer.

Because you're using an armature, you need to use the Selection tool only to modify your pose. Try it out! Notice how everything naturally moves in relation to any portion of the stalk you click and drag? This is the power of inverse kinematics.

Once an armature is assembled, it has a default pose. When you modify a property on any particular frame, it's called a *pose*. Poses can be copied for easy modification and reuse. You want your ending frame to contain a pose with the stalk fully extended.

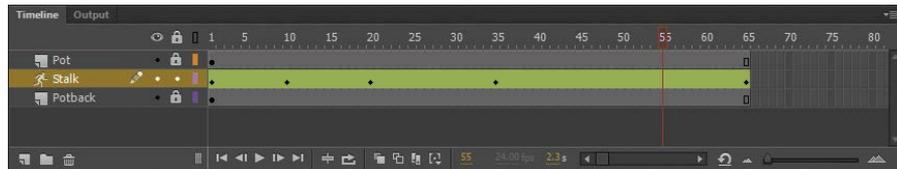
- 4 Select the small black diamond on frame 1 and right-click it.

 **Video 4.15**
Managing Armature Poses

Video 4.16
Animating the Stalk

- 5 Choose Copy Pose from the context menu and move the playhead to frame 65.
- 6 Select frame 65 and right-click again; this time choose Paste Pose.
The duplicated pose is shown as a small black diamond. Now you need to define your additional poses across the frame span.
- 7 Moving back to frame 1, modify the pose so that all segments of your stalk are collapsed down into the flower pot.
- 8 Move the playhead forward about 10 frames and create a new pose by guiding the growth of the stalk up and out of the flower pot.
- 9 Jump ahead about 10–15 more layers. Create as many poses and modifications to these poses as necessary to animate the snake-like motion of a plant growing up from the soil (**Figure 4.27**).

Figure 4.27 Keyframes set at 10-15 frame intervals, poses created



TIP You can also use inverse kinematics to construct an armature out of multiple Movie Clip instances. This is useful when building an armature for a human form or something mechanical in nature. Shape-based armatures are more suited to plants with long swaying stalks or creatures like snakes.

Designing and Animating the Flower Head

Now you'll create the head of the flower, which will bloom as the stalk grows. This process is pretty simple—you'll animate the flower head to follow the growth of the stalk and scale accordingly as the stalk rises until you have a fully matured flower.

Drawing the flower head

Video 4.17
Drawing the Flower Head

To draw the head of your flower, you'll use the Oval or Oval Primitive tool. But you'll need to do a little setup before you start drawing.

- 1 Create a new layer directly above the pot layer and name it **Flower**.
- 2 Lock all the other layers and move the playhead to frame 45 (**Figure 4.28**). This is where your flower will first appear.

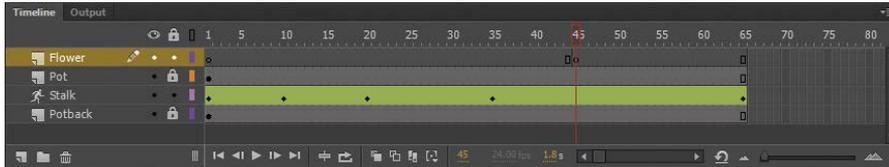


Figure 4.28 Playhead at frame 45

- 3 Choose Insert > Timeline > Blank Keyframe from the application menu to create a blank keyframe.
Now you'll use this new keyframe to contain the vector content that forms your flower head.
- 4 Choose the Oval Primitive tool and draw a small circle by holding Shift as you click and drag. The resulting circle shape should be about 40px to 44px in diameter.
- 5 Choose the same color for both the fill and stroke. I'm using #FF9900 in this example.
- 6 To roughen the stroke a bit, click the Brush Library icon with the object still selected and browse through the various brushes.
In this example I've selected Artistic > Chalk Charcoal > Charcoal - Tapered as my vector brush and adjusted the stroke width to a value of 8. This is the center of your flower.
- 7 To draw the petals, once again use the Oval Primitive tool and draw an oval that is roughly 25px wide and 65px tall. Change the color of the fill to something floral, like #FFFF99, and disable the stroke.
- 8 Duplicate this petal by making seven copies, resulting in eight complete petals.
- 9 Using the Free Transform tool, arrange each petal around the center circle shape created previously (**Figure 4.29**).

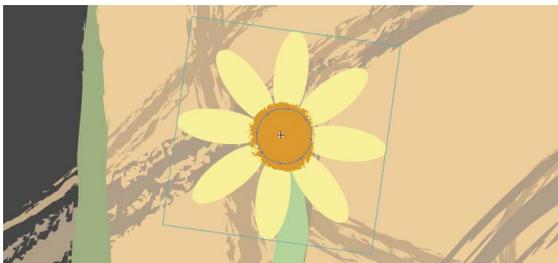


Figure 4.29 Petals arranged

TIP

If you need to modify the individual petals within the group, just double-click to enter the group for adjustments.

Video 4.18
Animating the Flower Head

- 10 Select all the petals and choose **Modify > Group** from the application menu.
All of the petals are now grouped together, making them easier to manage.
- 11 Right-click the new group and choose **Arrange > Send to Back** so that the center shape appears in front.

Animating the flower head along the stalk

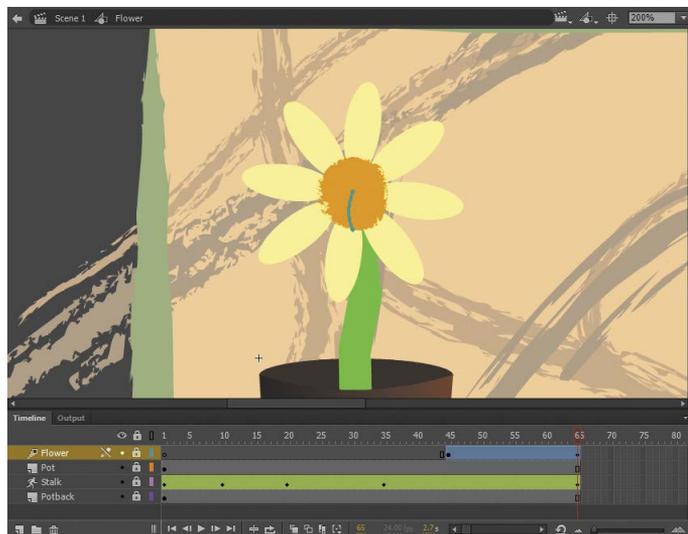
Even though the flower head is made entirely out of shapes, you can't use a shape tween to animate it because it is too complex. Instead, you'll create a Movie Clip symbol and apply a motion tween to the instance that results.

- 1 Select all the parts of the flower head—the petal group and the center portion—and choose **Modify > Convert to Symbol** from the application menu.
- 2 In the dialog box that appears, name the symbol **Bud**. For the type, choose **Movie Clip**.

The Movie Clip Symbol itself will now be in your project Library, and an instance of that symbol is added to the Stage.

Now that you have your completed flower head as a Movie Clip symbol instance on the Stage, all you need to do is tween it along with the growth of your stalk to complete the flower animation (**Figure 4.30**). To do this, you'll adjust the scale and position of the instance with a motion tween.

Figure 4.30 Completed flower head on the stage



- 3 Select any frame or span of frames on the Flower layer between frames 45 and 65, right-click the frame(s), and choose Create Motion Tween from the menu.

The background of the entire frame span should turn blue, and the layer icon should change to a small box in motion, indicating the motion tween has been created.

Now all you need to do is adjust the scale and position of the Movie Clip instance on both keyframes.

- 4 On frame 65, the instance should be at full size and positioned at the top center of the fully extended stalk. On frame 45, scale it down using the Free Transform tool or the Properties panel so that it measures about the same width as the stalk.
- 5 Position it at the top of the stalk wherever it is at this point in the animation. That should be it, although you may need to tweak the motion guide or even add an additional keyframe to help direct the tween.

NOTE

The motion guide is an overlay line that appears on the Stage when using a motion tween. The motion guide can be manipulated with the Selection tool to modify the tween path.

DETECTING PROBLEMS WITH TWEENS

Things may not always go as planned when tweening, and errors can occur. The most common tweening errors occur when you try to apply a tween on an incompatible element type.

An example is attempting to apply a motion tween to an object that doesn't support motion tweens. Animate CC tries to help you out by asking if you want to convert the object to a symbol to allow the tween. Although this works, you should have your instances properly converted before you apply the motion tween.

Another common error involves attempting to apply a shape tween to a pair of keyframes that don't contain shapes, in which case the option won't even be selectable. You can resolve this issue by using the Modify > Break Apart command to get the artwork to a basic vector level that a shape tween can be applied to.

Create a Personalized Text Message

★ ACA Objective 4.3

The final visual element to apply to your animated greeting is the message that the viewer sees after the animation has concluded. You might normally encounter problems working with text with HTML5 Canvas, since you can choose only Dynamic Text; you must be sure that the font used is a common font that everyone has installed. The old workaround is to break the text apart into shapes after it is created or to render the text elements to image data and then include those elements in place of editable text.

This is no longer necessary. Now Animate CC includes an option to render the text as outlines when publishing for HTML5 Canvas. So feel free to use any bizarre typeface you like in this greeting.

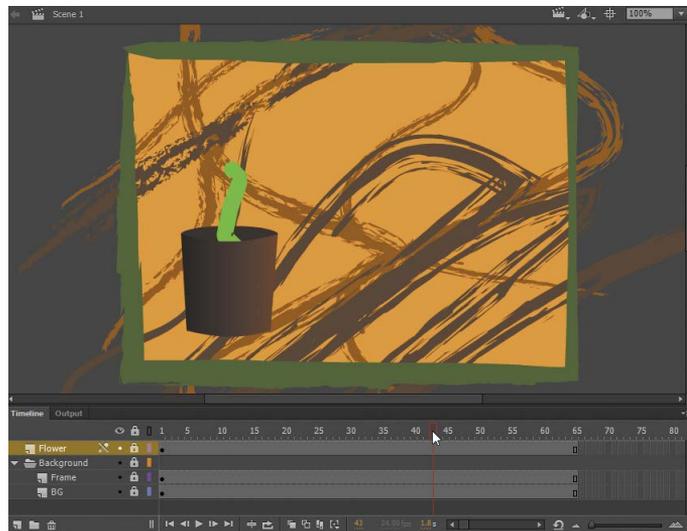
Building the text block

Video 4.19
Creating a Text
Greeting

In this exercise you'll build a text message that you'll include on the main timeline.

- 1 If you're still within the Flower Graphic symbol instance, click the Scene 1 identifier in the location breadcrumbs above the Stage to return.
- 2 Extend the main timeline to 65 frames just like the Flower animation by moving the playhead to frame 65, selecting this frame, and choosing Insert > Timeline > Frame from the application menu (Figure 4.31).

Figure 4.31 Timeline extended



With the number of frames on the main timeline now matching those within the Graphic symbol instance, scrubbing the playhead will reveal one of the strengths of Graphic symbols. Notice that for every frame, the animation is completely synchronous. If you compare this with a Movie Clip symbol instance, you'll see only the first frame of that animation unless you test or publish. This is one reason Graphic symbols are great for animation.

NOTE When you select the instance and check the Properties panel, you'll notice the *Looping* section. This section provides even greater control when using Graphic symbols; you can specify the starting frame, how many times the animation loops, and other options.

- 3 Create a new layer above the Flower layer and rename it **Message**.

This layer will hold your text message content. The message won't display until the animation is complete, so you need to create an empty keyframe on frame 65 and create your message there.

- 4 To do so, choose **Insert > Timeline > Keyframe** from the application menu.

Notice that it created a blank keyframe even though you didn't choose that option. That's because there was no content in the previous keyframe, so inserting a keyframe simply duplicates the content of the previous keyframe—which, in this case, contained nothing.

- 5 Select frame 65 on the Message layer, lock every other layer, and then choose the Text tool.

- 6 Access the Properties panel and notice in the Text Type drop-down that only Dynamic Text is selectable because you're targeting HTML5 Canvas (**Figure 4.32**). Choose a font that looks nice to you and set the size value to roughly 60 pt.

- 7 For color, click the color swatch and then use the Eyedropper tool that appears to sample the color from the flower petals on Stage. You can, of course, color as you desire.

- 8 With your text properties in place and the Text tool selected, click and drag a text block in an empty area of the Stage. Feel free to move the Flower instance to the side if need be.

- 9 Type in a message. I'm using the generic greeting "A Flower for You!" but as with many choices in this project, it's really up to you.

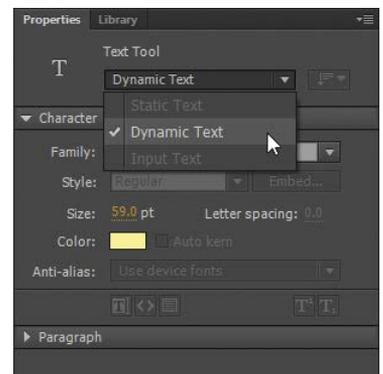


Figure 4.32 Choose Dynamic Text from the Text Type drop-down.

The text should display in the exact font, color, and size you have chosen (Figure 4.33).

Figure 4.33 Text displayed



- 10 Because the text is editable, select the text block to modify any of the properties within the Properties panel. You can also adjust the size of the text block itself. Double-click anywhere in the text block to edit the text and change your message, if needed.

Applying Text Effects

Video 4.20 Text Effects in HTML5 Canvas

All text types accept filters, which can be applied from the bottom of the Properties panel. You access these filters by selecting the text and then clicking the Add Filters icon in the Filters area.

Available filters include Drop Shadow, Blur, Glow, Bevel, Gradient Glow, Gradient Bevel, and Adjust Color.

Available filters for HTML5 Canvas are restricted to Drop Shadow, Blur, Glow, and Adjust Color.

Next you'll add a Drop Shadow to your text.

- 1 Select the text block and then click the Add Filters icon in the Properties panel.
- 2 Choose Drop Shadow and make the following adjustments: change Blur X and Blur Y to a value of 6, keep Strength at 100%, and choose #000000 for a color (Figure 4.34).

When you're using HTML5 Canvas as a publish target, many filters will work just fine on text. However, if you choose to render text to outlines in your publish settings, the filter will not be applied because a filter cannot be applied to simple shapes but only text blocks or Movie Clip content. As a workaround, you can always wrap the actual text within a Movie Clip symbol and then apply the filter to that instance on the Stage. You'll do this now.

- 3 With the text block selected, click the Options icon in the Filters area of the Properties panel. Choose the "Copy all filters" option and then remove the Drop Shadow by selecting it and clicking the Delete Filter icon.
- 4 Choose Modify > Convert to Symbol from the application menu. When the Convert to Symbol dialog box opens, create a Movie Clip symbol named **Message**.
- 5 Select the instance on the Stage and access the Properties panel. Movie Clip instances have a Filters area that's identical to what you've seen with text blocks.
- 6 Click the Options icon and choose "Paste filters." Watch as the Drop Shadow previously attributed to the text block becomes active on the Movie Clip instance.

Now, no matter how you choose to publish your text, it will render with the Drop Shadow applied in HTML5 Canvas (**Figure 4.35**).

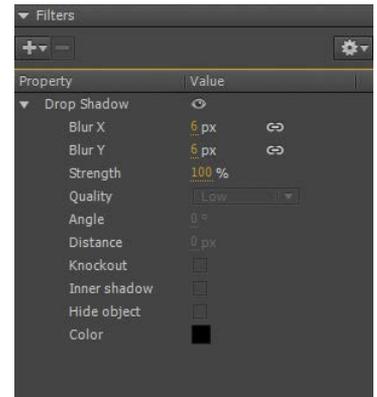


Figure 4.34 Choosing Drop Shadow settings



Figure 4.35 Drop Shadow applied

Using JavaScript Code to Stop the Playhead

★ ACA Objective 2.4

If you test the project at this point by choosing Control > Test from the application menu, you'll see how well everything has come together. But the animation loops continuously and the actual text message blips by too quickly to digest as it exists on only one frame out of 65. You need to stop that playhead so the viewer can read the message.

Working with code in Animate CC

Whether you're working with ActionScript or **JavaScript** in Animate CC, the initial workflow is basically identical. Code can be placed on the timeline on specific frames to instruct the project to perform instructions in various ways. The two main panels you'll deal with when writing simple code are the Actions panel and the Code Snippets panel. You access both from the Window application menu item.

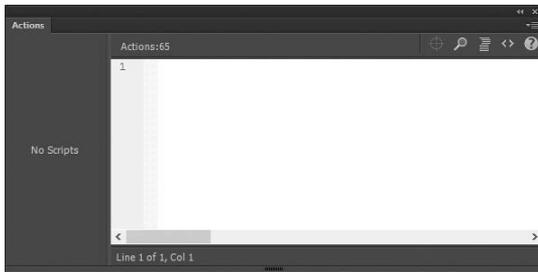


Figure 4.36 The Actions panel

ACTIONS PANEL

The Actions panel is a simple code editor where you write all your frame scripts (Figure 4.36). It doesn't matter what sort of code you're writing in the Actions panel—both JavaScript and ActionScript are completely supported.

Of course, you can use ActionScript only in an ActionScript-based project and JavaScript only in an HTML5-based project. Frame scripts are the pieces of code you bind to certain frames along the timeline.

Additionally, you have a sidebar, which makes it easy to jump around to different pieces of code located on different objects and frames.

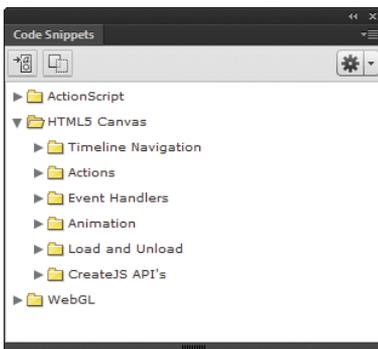


Figure 4.37 The Code Snippets panel

CODE SNIPPETS PANEL

Working in the Code Snippets panel (Figure 4.37) is a great way for those who are new to working with code to learn how to do some fundamental things within Animate CC projects. The snippets included are divided into three main categories: ActionScript, HTML5 Canvas, and WebGL. These correspond to whichever target platform you're working with.

Within these three main categories are a number of additional, functional categories. Every target platform represented has snippets to manipulate the playhead: either stopping the timeline from playing or jumping around to different frames. Take a look at the possibilities that snippets provide.

Using JavaScript to stop your animation

Now you'll stop that playhead and give your text more screen time with a little JavaScript. The first thing you need to do is decide where to place the code. You'll use frame 65 because that's the frame you need to stop the playhead upon—but which layer?

It's best practice in Animate CC to create a layer called Actions where all your frame scripts will be placed. Not only does this make it easy to find any of your code, but it also keeps the code separate from your visual assets that exist in other layers.

- 1 Create a new layer and name it **Actions**.
- 2 Insert a blank keyframe on frame 65 by choosing Insert > Timeline > Blank Keyframe from the application menu.

The only other thing you need to do is write some code within frame 65 on your Actions layer to stop the playhead.

- 3 With frame 65 of the Actions layer selected, open the Actions panel and place the cursor at line 1 (**Figure 4.38**). Type the following code within the panel:

```
this.stop();
```



Video 4.21
Stopping the Playhead

Figure 4.38 Code entered in the Actions panel

It's important to understand what this does. In the code, `this` refers to the present Stage—the main Stage. The instruction `stop()` is a method that tells the playhead on that Stage to stop moving immediately and stay there, effectively stopping the

animation. That's it! You could certainly use the Code Snippets panel to do this, but it is so simple that typing it in is the more direct route.

Notice on frame 65 that the keyframe has changed. In place of the empty circle, you now have a symbol that lets you know there is code on that keyframe.

NOTE *The syntax for stopping an animation in ActionScript is exactly the same as the syntax used here, in JavaScript. CreateJS was written to be as close to ActionScript as possible. Some major differences exist, however. Using JavaScript, frame numbers begin at 0 instead of 1 as they do when using ActionScript.*

ANIMATING WITH CODE

All of the animation techniques discussed so far in this book involve working with the timeline. This is the easiest way for a new interactive media creator to get something to move on the screen. Using ActionScript or JavaScript code to animate your creations can be challenging, but it provides a level of control and interactivity that can't be achieved using the timeline alone.

Carefully analyze what you're trying to accomplish, and determine the best route to getting the results required for the project. Once you get past learning the syntax for whichever platform you're targeting, your animation will be much more flexible and easier to edit once created.

Publishing Your Project for HTML5 Canvas

★ ACA Objective 5.4

▶ **Video 4.22**
Publishing for
HTML5 Canvas

Publishing for HTML5 Canvas is a bit more straightforward than publishing for Flash Player, simply because you have fewer publish settings to set. When publishing for HTML5 Canvas, in place of a single SWF file, you'll have a number of different files that must all be kept together for the project to run. The number of files you have depends on the file types used in the project and your publish settings. The primary file among these is a JavaScript file (with the .js extension).

Publish Settings

You can access the Publish Settings for HTML5 Canvas in the same manner you do when targeting Flash Player. With nothing in your project selected, the Properties panel displays document properties.

- 1 Under the Publish area of the panel, click the Publish Settings button to open the Publish Settings dialog box (Figure 4.39).
- 2 In the left-hand column in the Publish Settings dialog box, you'll see a column where you can select JavaScript/HTML under the Publish heading. Make sure this check box is selected. From here, you're able to choose a number of settings:
 - The first section allows customization of the output file, asking you to specify the filename and directory for the output. You also have the options to loop the timeline or overwrite any existing HTML.
 - Next are directories for your asset exports. If you use any images or sounds in your project, this is where they'll be published to. If you're including images, you can also export them all to a single spritesheet using these settings. There is also a place to specify the directory for the published CreateJS files if you're not using the content delivery network (CDN).
 - With the next option you can define JavaScript namespaces. This is useful when you're using a number of additional JavaScript libraries as part of your project to avoid conflicts.

HTML5 Canvas also has an Advanced section, which provides the following options:

- **Hosted Libraries:** Enabling this option means HTML5 Canvas will grab the CreateJS libraries from a remote CDN.
- **Include Hidden Layers:** This includes any layers hidden using layer control.

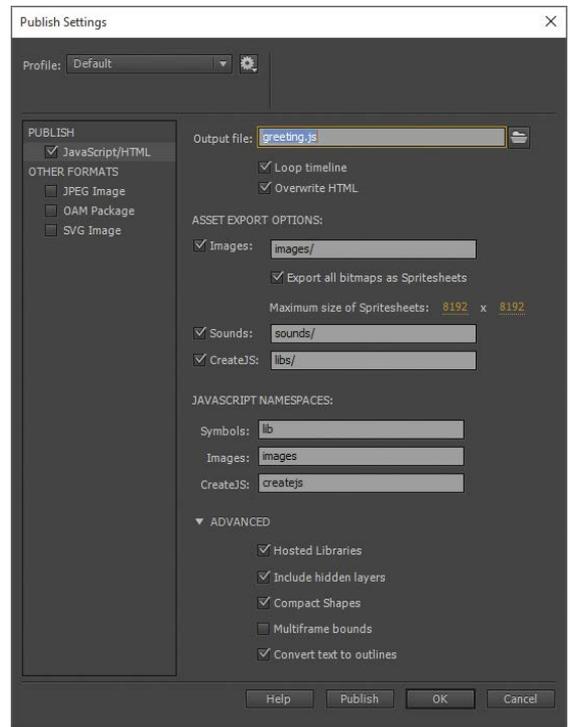


Figure 4.39 The Publish Settings dialog box

- **Compact Shapes:** Select this option if you want to make shapes as optimized as possible.
 - **Multiframe bounds:** This includes frame bounds properties to objects.
 - **Convert text to outlines:** Selecting this option means HTML5 Canvas will render dynamic text as shapes to preserve fonts.
- 3 When you're ready, click Publish.

WORKING WITH WEBGL

The secondary native HTML5 document type involves the use of WebGL (Web Graphics Library) technology. Interestingly enough, WebGL also makes use of the HTML5 canvas element, but in a different way. WebGL content is actually GPU-accelerated, meaning that it makes direct use of the hardware graphical processing unit (GPU) rather than sharing rendering tasks with everything else running on the central processing unit (CPU). This makes for a much more effective and powerful rendering target.

Although WebGL can be rendered in an accelerated way, it's more difficult to work with than HTML5 Canvas documents. Adobe has created a specialized WebGL runtime to make things easier, but it's still not as robust and direct in use as CreateJS. If you're considering WebGL, be sure that your project will benefit from being GPU-accelerated!

 **Video 4.23**
Converting to
WebGL

Challenge!

Right now, your message text just pops in. But it could transition into the message in a subtler way. Find a way to tween in the text as a fade, scale it in, or use some other mechanism for performing an advanced reveal. The next chapter will give you good ideas for accomplishing this.

Conclusion

So there you have it! You used a large variety of tools and techniques to build an animated greeting that can be run in your web browser without the need for Flash Player to be installed. That means your greeting will also run on mobile devices like Android phones and iPads. Of course, even if you do have Flash Player installed on your desktop, this project doesn't rely on or use it.

While working on this project, you've had a look at how easily some of the tools and concepts we already covered when targeting Flash Player are adapted to new publish targets. You've also explored new tools and techniques such as vector art brushes, the Paint Brush tool, variable-width strokes, the Width tool, armature layers, the Bones tool, Graphic and Movie Clip symbols and instances, text effects, and HTML5 Canvas and WebGL publish targets.

Index

3D Rotation tool, 71, 112, 113
3D Translation tool, 71, 112

A

accessibility, 209

Actions layer, 97, 140, 162

Actions panel, 96, 97, 140, 162

ActionScript 3.0

 Animate CC and, 17

 audio playback with, 166

 classes in, 162–163, 182–187

 comments used in, 168–169

 completed code example, 166–167

 document creation, 48–49

 errors encountered in, 168

 events in, 162, 165–166

 functions in, 161–162, 164

 fundamentals of using, 159–162

 JavaScript vs., 138–139

 panels for working with, 96–97

 publish settings, 40–44

 reference document, 211

 symbols combined with, 159–160

 testing projects created with, 168–169

 variables in, 160–161, 163, 191

 versions of ActionScript and, 138

 video embedding with, 131

ActionScript 4.0, 139

Add Anchor Point tool, 28

Adobe AIR, 13, 140–144

 AIR Settings dialog box, 170, 202–203

 Android apps and, 174

 application installer, 171

 blog related to, 211

 desktop document setup, 141–142

 explanation of, 141

 iOS apps and, 174

 mobile document setup, 175–176

 updating, 143

Adobe Animate CC. *See* Animate CC

Adobe Certified Associate (ACA) exam, 5

Adobe Flash Player. *See* Flash Player

Adobe Flash Professional CC, 3, 4–5, 6

Adobe Learn Books, 4

Adobe Media Encoder, 128, 129, 156

Adobe Media Server, 133

Advanced publish settings, 42–43

AI files, 52

AIR, Adobe. *See* Adobe AIR

AIR Application Installer, 171

AIR Settings dialog box, 170, 202–203

Align panel, 59

alignment

 of text, 63, 119

 of vector shapes, 59

Alpha slider, 117

Android apps, 174

Animate CC, 3

 code features, 96–97

 exporting video from, 127–128

 Flash Professional CC as, 4–5

 interface features, 6–7, 9–13

 resources, 211

 symbol types, 80–81

 video playback and, 130–134

 workspace management, 6–9

animated greeting card project, 69–101

 animation processes, 85–88, 90–91

 background elements layout, 73–77

 Challenge! exercise, 101

 drawing the images, 77–81, 85, 88–90

 editing the graphic symbol, 83–84

 HTML5 Canvas document setup, 70–71

 JavaScript code used in, 96–98

 publishing for HTML5 Canvas, 98–100

 Stage and timeline configuration, 72

 text message creation, 92–95

animations

 code used for, 98

 creating, 32–35

 easing applied to, 38, 155

 frame-by-frame, 35

 inverse kinematics for, 85–86, 88

 JavaScript for stopping, 97–98

 mask, 107–109

 modifying and improving, 36–38

 motion tweens used in, 90–91

 name, 118–120

 previewing, 39

 publishing, 39–44, 98–100, 169–171

 raw video from, 127–130

Animator workspace, 8
anti-aliasing options, 63–64
apps
 Android vs. iOS, 174–175
 See also mobile quiz app project
armature creation, 86–88
Array object, 190, 196
Array.push() method, 193
Array.splice() method, 196
audio, 155–159
 file types, 156
 importing, 156–157
 quality options, 42
 sync settings, 158–159
 triggering with ActionScript, 166
 using on timeline, 157–159

B

backgrounds
 drawing, 73–74
 importing, 105, 143
 transparent, 144
beta testing, 210
bitmap fill, 145
bitmap images
 adjusting on the Stage, 53
 breaking apart, 54–55, 110
 importing, 51–53, 145
 managing, 54–58
 tracing, 56
Black and White icon, 12
blank keyframes, 32, 89
blend modes, 118
block comments, 169
Blur filter, 115–117
Bone tool, 87
breadcrumb trail, 25
Break Apart command, 55, 64, 110
Brush Library, 28, 74–75
Brush tool, 27, 85
brushes, editing, 75
bugs/errors, 207
Button symbols, 81, 179
buttons
 adding interactivity to, 197–199
 process of designing, 179–181

C

Captive Runtime, 174
circular icon, 25
class keyword, 184
classes in ActionScript, 182–187
 creating new, 183, 185–186
 defining, 183–184, 186–187
 explanation of, 182
 importing, 162–163
 integrating, 191–195
 saving, 184–185
Classic motion guides, 122–127
 animating descriptors with, 122–125
 variable-width strokes as, 126–127
Classic tweens, 124, 127
Classic workspace, 8
clock design, 144–155
 body of clock, 144–145
 digital display, 147–148
 face of clock, 146–147, 149–150
 hands of clock, 150–152
 pendulum, 152–155
 See also virtual clock project
code
 animating with, 98
 panels for working with, 96–97
 using JavaScript, 97–98
Code Snippets panel, 96–97
Color panel, 32
colors
 applying to shapes, 78–79
 tools for modifying, 31–32
comments, ActionScript, 168–169
Compact Shapes option, 100
compilation errors, 168
Compress movie option, 42
content delivery network (CDN), 99
content evaluation, 207–208
Convert Anchor Point tool, 29
Convert to Symbol dialog box, 80
converting
 animations to raw video, 127–130
 elements to symbols, 79–80, 90, 111
 text to outlines, 100
copying/pasting keyframes, 34
CreateJS, 70, 98
creativity, 5–6
custom workspaces, 9

D

- `dataReady()` method, 192
- Date class, 163
- Debug workspace, 8
- Delete Anchor Point tool, 28
- Delete Layer icon, 84
- demo reel, 103
- deployment testing, 207
- deprecation, ActionScript, 138
- Designer workspace, 8
- desktop installations, 169–171
- Developer workspace, 8
- digital rights management (DRM), 128, 134
- `dispatchEvent()` method, 193
- Document Properties dialog box, 19
- documents
 - ActionScript 3.0, 17, 40–44
 - creating for projects, 16–23, 48–50
 - setting properties for, 18–19, 20
- drawing
 - backgrounds, 73–74
 - merge vs. object, 25
 - shapes, 25–26, 58–59, 85
 - vector assets, 23–29
- DRM (digital rights management), 128, 134
- drop shadows, 94–95
- dynamic text, 62, 92, 147–148, 177

E

- easing controls, 38, 155
- ECMAScript specification, 138–139
- Edit Guides option, 50
- Edit Stroke Style icon, 75
- effects
 - fade-in, 117–118
 - sound, 159
 - text, 94–95
- embedded video files, 131–132
- errors
 - ActionScript, 168
 - testing for, 168, 207
- Essentials workspace, 8
- evaluation process, 206–210
 - accessibility evaluation, 209
 - content evaluation, 207–208
 - initial or preliminary evaluation, 207
 - technical evaluation, 206–207
 - usability evaluation, 208–210
- event listeners, 165

- events, ActionScript, 162, 165–166
- expanding-width text, 61
- Export Video dialog box, 127
- exporting
 - images, 64–67
 - video, 127–128
- Eyedropper tool, 31

F

- F4V file format, 131
- fade-in effect, 117–118
- failure, 6
- file types
 - audio, 156
 - FLA, 17
 - image, 64–66
 - video, 130–131
- Fill Color swatch, 12
- fills
 - bitmap, 145
 - locking, 27
 - selecting, 29
- filters
 - accessing, 115
 - Blur, 115–117
 - text, 94–95
- fixed-width text, 61
- FLA files, 17, 20
- Flash Player, 3, 5, 39, 130
- Flash Professional CC, 3, 4–5, 6
- FLV file format, 130–131
- FLVPlayback component, 131
- focus groups, 210
- folder icon, 22
- folders, layer, 77
- Font Embedding dialog box, 63
- fonts
 - embedding, 62–63
 - selecting, 119, 148
- for each loops, 190
- for loops, 193
- frame rate, 38
- frame-by-frame animation, 35
- frames
 - explanation of, 13, 32
 - inserting a span of, 33
 - labels for, 160
 - See also* keyframes
- Free Transform tool, 36–37, 78, 89
- functions, ActionScript, 161–162, 164

G

- `gameOver()` method, 196–197
- GIF files, 65
- Google Play, 174
- GPU-acceleration, 100
- Gradient Transform tool, 78, 79
- gradients
 - shapes and, 78–79
 - strokes and, 126
- Graphic symbols, 81, 83, 93
- greeting card. *See* animated greeting card project
- grid, 50
- Group command, 147
- grouping layers, 77
- guides, 50

H

- H.264 video format, 130, 131
- Hand tool, 12
- Hardware acceleration option, 43
- HDS streaming protocol, 134
- headers, app, 176–177
- headshot movie clip, 109–118
 - creating, 109–111
 - fade-in effect, 117–118
 - filter applied to, 115–117
 - rotating, 112–115
- hidden layers, 42, 99
- hints, shape, 35, 109
- HLS streaming protocol, 133
- Hosted Libraries option, 99
- HTML sample file, 43–44
- HTML5 Canvas
 - document creation/setup, 70–72
 - Dynamic Text used in, 71, 92
 - filters available in, 94, 95
 - publishing projects for, 98–100
- HTML5 video element, 130
- HTTP Dynamic Streaming (HDS), 134
- HTTP Live Streaming (HLS), 133

I

- icons
 - desktop, 170
 - mobile, 203
- Illustrator files, 52–53
- image generation, 64–67
- `import` statement, 162

- Import Video command, 132
- importing
 - audio files, 156–157
 - background images, 105, 143
 - bitmap images, 51–53, 145
 - video as a guide, 132
- informal testing, 209
- inheritance, 184
- initial evaluation, 207
- Ink Bottle tool, 31, 111
- input text, 62
- instances
 - class, 182
 - symbol, 82
- interactive buttons, 197–199
- interactive media content
 - evaluation process, 206–210
 - revisions process, 210
- interface
 - features overview, 6–7, 9–13
 - See also* workspaces
- internal attributes, 182
- inverse kinematics, 85–86, 88
- iOS apps, 174, 175

J

- JavaScript
 - ActionScript vs., 138–139
 - panels for working with, 96–97
 - Publish Settings related to, 99
 - stopping animations using, 97–98
- JavaScript libraries, 70
- JPEG Deblocking option, 42
- JPEG images
 - exporting stills as, 64–65
 - quality settings for, 41–42
- JSON files, 173, 187–191
 - building, 187–188
 - code example, 188–189
 - loading, 189–191
- `jsonLoaded()` method, 190

K

- keyboard shortcuts, 13
- keyframes
 - adjusting, 37
 - audio attached to, 158
 - blank, 32, 89
 - copying/pasting, 34

explained, 13, 32
inserting, 33–34, 37
labeling, 160
See also frames

L

labels, frame, 160
Land, Edwin, 6
Lasso tool, 30
Layer Options menu, 23
Layer Properties dialog box, 33
layers, 13

- Actions, 97, 140, 162
- armature, 86–88
- deleting, 84
- grouping, 77
- hidden, 42, 99
- icons for managing, 22
- mask, 57–58
- outline mode for, 54
- setting properties for, 22–23
- stacking order of, 26
- visibility of, 54

Library panel, 57
Line tool, 27, 123, 150
linear gradients, 78, 79
Linkage ID, 52, 157
Linux systems, 141
`loadJSON()` method, 190
Local playback security option, 43
Lock Guides option, 50
lock icon, 22

M

Magic Wand tool, 30
Manage Adobe AIR SDK dialog box, 143
masks

- animating, 107–109
- drawing shape, 106–107
- hiding content with, 57–58

meme image project, 47–67

- black vector shape creation, 58–59
- Challenge! exercise, 67
- document creation/setup, 48–50
- importing bitmap images, 51–53
- managing bitmap images, 54–58
- still image generation, 64–67
- text elements, 59–64

Merge Drawing mode, 25
methods, ActionScript, 182
mobile quiz app project, 173–203

- Adobe Air document creation, 174–176
- button interactivity, 197–199
- Challenge! exercise, 203
- code example, 199–202
- interface element design, 176–181
- logic and flow description, 195–197
- publishing for distribution, 202–203
- question bank creation, 187–195
- writing the classes for, 182–187

Motion Editor, 114, 115, 116–117
motion guides, 122–125
motion presets, 120
motion tweens, 90–91, 122
mouse interaction, 165–166
`MouseEvent` class, 163, 197
Movie Clip symbols, 80

- ActionScript and, 160
- animated greeting card project, 90–91
- Graphic symbols distinguished from, 83
- promotional video project, 111, 121
- virtual clock project, 149

`MovieClip` class, 163, 182
MP3 file format, 156
MP4 file format, 130, 131
Multiframe bounds option, 100

N

name animation, 118–120

- motion presets for, 120
- movie clip creation, 119

New Document dialog box, 71, 142
New Layer icon, 21, 72
number variable, 160

O

Object Drawing mode, 25, 73
objects, class, 182
onion skinning, 37
Options area, 12
outline text, 92, 100
Output File option, 41
Oval Primitive tool, 24, 89, 146
Oval tool, 24, 78, 146, 179

P

- packages, 182
- page icon, 22
- Paint Brush tool, 28, 74–75
- Paint Bucket tool, 31
- panels
 - overview on managing, 7–8
 - See also specific panels*
- Paste in Place option, 34
- pasteboard, 10
- Pen tool, 28
- Pencil Mode icon, 27
- Pencil tool, 27
- permissions, 203
- Permit debugging option, 43
- perspective motion, 115–117
- Photoshop files, 52–53
- placeholder text, 148
- `play()` method, 166, 182
- playback security, 43
- PNG files, 66
- Polygon tool, 30
- PolyStar tool, 29, 106
- poses, armature, 86, 87–88
- preliminary technical evaluation, 207
- Preset Browser panel, 129
- presets
 - motion, 120
 - workspace, 8
- previewing animations, 39
- primitive tools, 24
- private attributes, 182
- Profile settings, 40–41
- progressive download, 132
- Project Library, 51
- projectors, 44
- promotional video project, 103–135
 - 3D rotating headshot design, 109–118
 - Challenge! exercise, 135
 - Classic motion guide layer, 122–125
 - descriptor text movie clip, 121
 - document creation/setup, 104
 - dynamic background creation, 106–109
 - importing backgrounds for, 105
 - name animation process, 118–120
 - rendering/converting to raw video, 127–130
- properties
 - class, 182
 - document, 18–19, 20
 - layer, 22–23
- Properties inspector, 38

- Properties panel, 11, 55, 72, 83, 209
- Protect from import option, 43
- protected attributes, 182
- PSD files, 52
- public attributes, 182
- Publish Settings dialog box, 40–44, 99
- publishing animations, 39–44, 98–100, 169–171
 - ActionScript settings for, 40–44
 - for Adobe Flash Player, 39, 44
 - for app distribution, 202–203
 - for desktop installation, 169–171
 - HTML5 Canvas settings for, 98–100
 - projectors created for, 44

Q

- quiz app. *See* mobile quiz app project

R

- radial gradients, 78
- raw video files, 127–130
- Real Time Messaging Protocol (RTMP), 133
- Rectangle Primitive tool, 24, 73, 144
- Rectangle tool, 24, 58, 73, 77, 144
- rendering animations to video, 127–130
- resources for Animate CC, 211
- reverse-domain naming scheme, 182
- revisions process, 210
- rotating shapes, 36
- Rotation tool, 12
 - See also* 3D Rotation tool
- RTMP streaming protocol, 133
- rulers, 49
- runtime errors, 168

S

- saving classes, 184–185
- scaling shapes, 36
- Script settings, 41
- Script time limit option, 43
- security, playback, 43
- Selection tool, 29–30, 53, 78, 145, 147
- `setTimeout` class, 197
- shape hints, 35, 109
- shape tweens, 32, 35, 91
- shapes
 - drawing, 25–26, 58–59, 77–78, 85
 - masks created from, 106–107
 - modifying the color of, 31–32
 - working with vector, 29–32

- sharing library items, 57
- signing code, 170, 202
- simulated 3D, 113
- single-line comments, 169
- size report, 42
- skewing shapes, 36
- Small Screen workspace, 8
- smart palettes, 11
- snapping options, 20–21
- sound effects, 159
- `SoundMixer.stopAllSounds()` command, 166
- spelling check, 208
- spritesheets, 16
- stacking order, 26
- Stage, 10
 - adjusting image layouts on, 53
 - configuring timeline and, 72
 - setting properties for, 48, 49
- Stage3D frameworks, 113
- static attributes, 182
- static text, 62, 147, 177
- still image generation, 64–67
- `stop()` method, 97, 182
- streaming video, 132–134
 - playback using, 132–133
 - types of, 133–134
- string variable, 161
- Stroke Color swatch, 12
- strokes, 29
 - drawing variable-width, 76, 126–127
 - editing brush style for, 75
- subject matter experts (SMEs), 208
- Subselection tool, 28, 30, 53, 78
- SVG files, 66–67
- Swap Colors icon, 12
- Swap Symbol dialog box, 125
- swatches, tagged, 79
- Swatches panel, 31, 79
- SWF files, 39, 44
- symbols
 - ActionScript and, 159–160
 - converting elements to, 79–80, 90, 111
 - duplicating, 180
 - editing, 83–84
 - instances and, 82
 - movie clip vs. graphic, 83
 - types of, 80–81
- Sync drop-down menu, 158–159
- synchronizing audio, 158–159

T

- Tagged Color Definition dialog box, 79
- tagged swatches, 79
- Target settings, 41
- technical evaluation, 206–207
- Test Movie option, 39
- testing
 - ActionScript projects, 168–169
 - beta, 210
 - deployment, 207
 - informal, 209
 - See also* evaluation process
- text, 59–64, 92–95
 - alignment of, 63
 - animation of, 118–120
 - anti-aliasing options, 63–64
 - converting to outlines, 100
 - creating elements of, 60–61, 92–94
 - dynamic, 62, 92, 147–148, 177
 - effects applied to, 94–95
 - font embedding for, 62–63
 - mobile app interface, 176–178
 - Movie Clip symbol, 121
 - static, 62, 147, 177
 - types of, 62
- text blocks
 - adjusting, 61
 - breaking apart, 64
 - building, 92–94
 - expanding- vs. fixed-width, 61
- Text tool, 60, 71, 119, 147, 176
- `TextField` class, 163
- timeline, 21
 - audio placed on, 157–159
 - configuring the stage and, 72
 - frame labels on, 160
 - internal to symbols, 84
- Timeline panel, 12–13
- `Timer` class, 163, 164
- tools
 - commonly accessed, 12
 - primitive, 24
 - selection, 29–30
 - vector creation, 27–29
 - See also specific tools*
- Tools panel, 11
- top-level classes, 163
- Trace Bitmap command, 56
- trace statements, 42, 169
- transcoding video, 129

transform point, 37
transparent backgrounds, 144
trash can icon, 22
twens
 Classic, 124, 127
 motion, 90–91
 problems with, 91
 shape, 32, 35, 91

U

Update Library Items dialog box, 56
usability evaluation, 208–210

V

vanishing point, 112
var keyword, 160, 161
variables, ActionScript, 160–161, 163, 191
variable-width strokes, 76, 126–127
vector animation project, 15–45
 Challenge! exercise, 45
 creating the animation, 32–35
 document creation, 16–23
 drawing vector assets, 23–29
 modifying the animation, 36–38
 publishing the animation, 39–44
 working with vector shapes, 29–32
vector images, 23–29, 51
video
 embedded, 131–132
 exporting, 127–128
 file types, 130–131
 progressive download, 132
 protecting with DRM, 134
 recorded for this book, 6
 streaming, 132–134
 transcoding, 129
virtual clock project, 137–171
 ActionScript for controlling, 162–167
 Adobe AIR document creation, 140–144
 audio added to, 155–159
 basic element creation, 144–148
 Challenge! exercise, 171
 movie clip asset design, 149–155
 publishing for desktop installation, 169–171
 testing ActionScript code, 168–169
void data type, 192

W

web browsers
 embedding videos for, 130
 Flash Player plug-in for, 3, 5, 39
WebGL technology, 100
Welcome Screen, 18
Width tool, 76
workspaces, 6–9
 creating custom, 9
 panel management, 7–8
 using preset, 8

X

XFL files, 20

Y

YouTube, 129, 130

Z

Zoom tool, 12
zoom-in-2D preset, 120
Z-order, 22, 26