



Flash AS3 Jumpstart - zero to Hero

Level: Fast track to Expert
Duration: 5 Days
Time: 10 AM - 5 PM
Cost: £997 ex VAT

Overview

- Day 1 – Introducing Actionscript
- Day 2 – Orientation to OO Development
- Day 3 – Working with an Entire Project
- Day 4 – OO, Events and Design Patterns
- Day 5 – Building and Implementing Components

Description

Adobe Flash Actionscript Jumpstart provides experienced Flash designers with the knowledge and hands-on practice they need to create dynamically generated event-driven animation and interactive games with Flash. And then later we show you how to build even more flexible and dynamic design based Flash content.

This Flash Package is great for people who want to learn ActionScript and a whole lot more. This class is a combination of the Flash Actionscript course and Flash Advanced Actionscript course.

This course provides experienced Adobe Flash designers with the knowledge and hands-on practice they need to create dynamically generated event-driven animation and interactive games with Adobe Flash. The course teaches fundamental programming techniques and begins by introducing core concepts including instance names, variables, functions, properties, and methods; then proceeds through conditions, loops, event handling, and animating with Adobe Flash ActionScript.

Also - this course focuses on teaching students more advanced ActionScript topics that will remove a reliance on Timeline-based visual tools. Students will learn to generate dynamic design and navigation elements that cannot be created without ActionScript.

Who is this course for?

This course is for Adobe Flash designers who have some experience with Adobe Flash design, Object-oriented programming and would like to learn basic to advanced ActionScript. This class is also fantastic for those who are wanting to take the Adobe Certified Expert (ACE) exams

Assumed Knowledge

To take this course, you should have completed the Adobe Flash ACA Jumpstart course, or have equivalent experience. The desire to learn how to program Adobe Flash content

using Adobe Flash ActionScript is a must.

What you will learn

By the end of the training course you will have

- * knowledge of Variables and Properties,
- * Functions and Methods,
- * Text Format Class,
- * Attaching Movie Clip Objects at Runtime,
- * Event Handling,
- * Arrays,
- * Objects, and built-in classes,
- * Loops and Conditions,
- * ActionScript Classes,
- * ActionScript Animation,
- * Object Drawing Model,
- * Blend Modes in Flash,
- * Loading XML,
- * Advanced Flash Text and CSS,
- * Loading Graphics Dynamically,
- * Animation with ActionScript,
- * Advanced Styling with CSS,
- * Components in Flash,
- * Attach Movie Clips Dynamically.

Outline

h3. Day 1 – Introducing Actionscript

Introducing Actionscript

- a brief history of flash
- Actionscript 123
- building a flash file
- exploring the document settings
- adding code on the timeline
- layers, organization and good structure

The Actionscript Syntax

- adding comments
- variables and assigning memory
- data types and assigning values, String, Number, int, uint and Boolean
- complex data types and instantiation and Date
- trace – your best friend in flash development
- breakpoints, debugging and the downloads

Controlling our Visual Design with our Code

- the importance of instance names

- feeling your way in the dark
- strong typing and defining aliases in code
- using getChildByName to access the 'named' objects
- casting - giving flash a hint
- modifying a movieclip's properties, x, y, alpha, scaleX, scaleY and rotation
- saving time (but not effort) with with

All the World's a Stage

- the width and height of the stage
- some useful properties, frameRate, fullScreenWidth, fullScreenHeight, displayState and wmodeGPU

Using the Stage Dimensions

- randomizing properties, x ($\text{Math.random()} * \text{this.stage.stageWidth}$) etc
- its no fun with just 1 – the for loop and numChildren
- that's great, but what if we want 142

Tidying up the Code

- functions to do stuff
- call the functions
- passing an object to a function

If you can do it in Design...

- we've seen alphas, what about a tint, importing extra libraries filters such as bevel, drop shadow and blur
- introducing arrays, some simple shorthand and more formal instantiation

Adding Mouse Interaction

- asking questions, an if and an is
- the buttonMode property
- adding event listeners (the trickiest of all)
- event handling functions and the game of catch
- back to trace
- event.target

Controlling the Flow

- stopping the movie
- making it start again, and designing a button to do it
- changing the mouse pointer
- adding an audible click using the code

Making Interaction More Interactive

- mouse down and mouse up
- enabling drag and drop
- limiting the drag area
- watching out for classic problems

- bringing our item to the top
- adding and removing a drop shadow

Scripted Animation Using the Tween Class

- importing the necessary
- on drop.... plummet
- its ok, but where's gravity gone? – easing to the rescue

h3. Day 2 – An Orientation to Object Oriented Development

Drawing is so passe

- when we draw we create an object, can we do it in code?
- our first class
- extending code to save time and effort
- the constructor function
- shapes and the drawing api
- this is all a bit conceptual, let's put one on the stage
- making our shape randomly position itself
- applying a gradient fill with code
- applying a tint in a random colour

Perhaps this is a bit Extreme

- Design <-> Develop workflow
- draw in the library, control behaviour in code
- our second class
- public and private instance variables
- reintroducing our randomness
- add many from the stage and see the code take charge

So if we can Inherit Behaviour

- the concept of inheritance
- a simple UML diagram
- sneaking something in the gap
- writing a 'drag and drop' movieclip class
- applying this to our class using inheritance
- testing the behaviour

Controlling the Whole Movie

- if we can control each library item, can we control the movie?
- getting the code off the timeline
- the document class – a massive movieclip
- loading and playing an mp3 bed
- controlling the volume of the audio
- event listeners and the id3 meta data in the mp3

Allowing Our Objects to Move

- can we give an object motion?
- instance variables and introducing setters/getters
- waiting until our object is on the stage
- the 'enter frame' event
- handling the event in a protected function
- using the Point class to help with the trigonometry
- checking we're still on the stage
- apply the behaviour to some library items as the 'base class'

Keeping Track of Each Other

- instance versus class variables
- maintaining a list in a static variable
- iterating through the array on enter frame
- performing a hit test
- handling the collision
- notifying the parent container
- handling this notification

Watching Some Video

- building a simple interface
- toggling the play/pause button
- displaying the flv filename
- allowing our movie to change the filename
- getting the video to work

Watching Ourselves

- using the camera
- taking a snapshot
- bitmaps and bitmapdata
- using the UILoader class to display the snapshot

h3. Day 3 – Working with an Entire Project

Working in a Project

- creating a flash project
- creating and setting our default document
- coming up with a design
- building our scene
- working in layers
- buttons, movieclips and navigation
- splitting out our pages into modules
- loading them in with the UILoader
- adding our document class and working in folders

Building a Preloader

- building a preloader animation

- adding a scene for it
- the bandwidth profiler gives it all away
- 'export for actionscript' and the compiler settings
- writing the 'PreloadedDocument' base class
- implementing the base class with inheritance
- testing the movie and simulating the download

Tracking the Progress

- creating the 'ProgressDisplay' movieclip
- discussing the design
- building the components and naming the instances
- adding it in, and naming it !
- writing the code to control it
- testing the movie and simulating the download

Building Our Pages as Separate Movies

- working with multiple swfs
- adding the movies to the project publish list
- designing the scenes
- implementing our preloader and progress
- testing the movies individually

Implementing the Navigation

- listening for mouse events
- event.target and the switch statement
- getChildByName and the UILoader
- loading the correct page
- testing the movie and simulating the download

Dynamically Loading Text

- The Home document class
- changing our static text to be dynamic
- getChildByName and setting the text programmatically
- font embedding
- extracting our text into an external file
- adding some html formatting to the text
- loading the text from the file
- tracking the load complete event
- displaying the text in the dynamic text box
- exploring the html formatting
- revisiting our font embedding
- adding scrollable functionality to the text box

Going Even Further

- we want to build a list of products
- building the list in another fla

- implementing the preloader
- using a scroll pane to load the content
- testing the project and simulating the download

- Bonus Topic – Loading XML Data
- loading some xml, after all its only text
- breaking it down and grabbing what we want
- looping through and tracing out the data
- homework – try and build the products list dynamically

h3. Day 4 – OO, Events and Design Patterns

Straight in at the Deep End

- a revision of OO and the document class
- design patterns and introducing mvc
- planning our strategy
- the model and the singleton pattern
- handling the logic in the controller
- custom events and inheritance
- the problem with events
- building an event bus
- commands to do the work and interfaces
- linking the event to the command
- triggering our first command
- utilizing someone else's code – the flex sdk
- loading our data from an xml file
- strong typing and valueobjects
- responders, asynctokens and the model
- telling the view our data's ready

Dynamically Creating Content

- first, build a visual representation of the component
- program the class and alias the visual children
- expose the properties using setters and getters
- think about accessibility
- linking to the model
- for each iteration
- adding and positioning children
- the scroll pane as a useful tool
- rollover interaction and the mouse enabled property

Custom Events that Carry Data

- previewing a product
- dispatch from a view, handle in the controller
- store the 'selected' item in the model
- notifying the view

- constructing the component
- writing the class to tie it all together

A Favourites List

- movieclips as buttons and a handy base class
- events and commands to add to the favourites
- using the List control and the DataProvider class
- checking we're not adding the same product twice
- removing a single item from the list
- removing all the items from the list
- storing the favourites for next time
- loading the favourites next time in

Using FlashVars

- passing data from the html into the movie
- allow the xml filename to be passed in
- breakpoints and debugging
- making our movie load a different xml file

Making it Look Good

- nesting our movieclips allows us to apply effects, but not hinder functionality
- adding a little loading animation

h3. Day 5 – Building and Implementing Components

Building an Audio Player

- the visual components
- an event architecture
- loading and playing the sound
- id3 tag info and displaying it in a text box
- controlling the volume
- muting the sound
- BONUS – analyzing the sound and displaying the frequency spectrum
- using compiler meta data to add a component definition
- using the component definition in the designer
- passing in the filename using the component inspector
- using our player in the products application

Toggling to Full Screen

- building a toggle button
- listening to the stage
- changing the display state
- making it work in the browser with html
- using our full screen toggle in the products application
- discussing the stage scale modes

Downloading Images From Our Application

- a custom event
- linking to a custom command
- the FileReference class
- event listeners and notification of the download
- notifying the user with a modal popup

Exploring the Idea of Popups

- a compare favourites panel
- iterating through the favourites data
- adding instances of a 'compare' movieclip
- using the scroll pane horizontally
- adding and removing the popup
- applying a transition with the transition manager