ACTIONSCRIPT 3.0

- Pallav Gala

What's ActionScript?

- A programming language, developed by Macromedia Inc. (taken over by Adobe Systems)
- Used for websites and applications based on Adobe Flash Player and Adobe AIR run-time environments
- Compliant to ECMAScript, syntactically similar to JavaScript
- Executes in the ActionScript Virtual Machine (AVM)

A brief history…

- **ActionScript 1.0**
- Introduced in 2000
- Provided prototype-based programming and loose type system feature

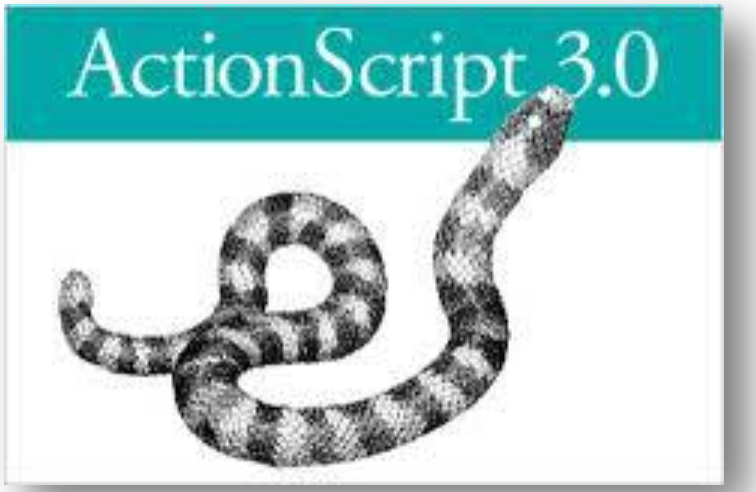
- **ActionScript 2.0**
- Introduced in 2003
- Provided class-based programming, compile time type checking and cross compilation to ActionScript 1.0 code

ActionScript 3.0

- Introduced in 2006 with release of Adobe Flash Player 9 and Adobe Flex 2.0
- Executes up to 10 times faster than the legacy ActionScript code on the new highly optimized ActionScript Virtual Machine(AVM2)
- Provide a true object oriented model for programmers
- Enables creation of applications with large data sets and object oriented reusable code

Features of ActionScript 3.0

- Since ActionScript 3.0 is fundamentally and architecturally different from AS 2.0, it provides many new features that increase performance and control over low-level objects
- Core Language Features
- API Features



Core Language Features

- Defines the fundamentals of the programming language such as data types, expressions, loops, etc.
- ActionScript 3.0 provides:
  - **Run-time exceptions** (reports more error conditions)
  - **Run-time types** (provides perform run-time type checking)
  - **ECMAScript for XML (E4X)** (AS3 implements E4X)
  - **Sealed classes** (includes the concept of sealed classes)
  - **Method closures** (to remember object instance, useful in event handling)

Flash API Features

- ActionScript 3.0 includes the following new APIs that give the user a better control over objects at a lower level
- **DOM3 event model** (provides a standard way of generating and handling events)
- **Display list API** (provides a set of classes for working with the visual primitives in Flash)
- **Working with text** (provides a flash.text package for all text-related APIs)

Let's Get Started

- With this basic idea of what ActionScript is, lets learn some basic programming concepts and then move on to some advance topics

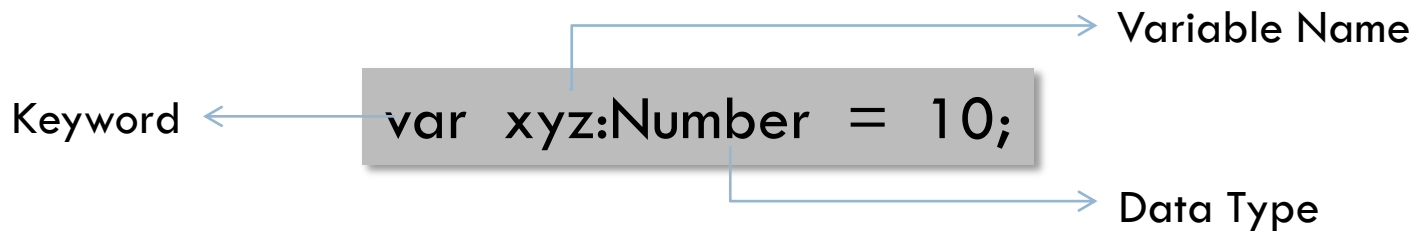
Basic Programming Concepts

ActionScript, like other programming languages provides basic features to users, such as:

- Variables
- Constants
- Data Types
- Operators
- Conditional Statements
- Looping Statements

Variables and constants

- A ***variable*** represent a piece of DATA to be used in the program
- Declaring and initializing a variable



- A ***constant*** represents a piece of DATA that can be assigned a value only once during the course of program execution

```
const xyz:Number = 10;
```

Data Types

Simple Data Types	**Complex Data Types**
Represent a single piece of data	Represent a set of values in a container
Examples: - **String:** A text, or array of characters - **Number:** Any numeric value - **Int:** Only integer values - **Boolean:** True or False value	Examples: - **MovieClip:** a movie clip symbol - **SimpleButton:** a button symbol - **Date:** information about a single moment in time - **TextField:** a dynamic or input text field

Operators

- Special functions that use one or more operands and return a specific value

Examples:

- **Arithmetic** : +, -, *, /, %
- **Logical** : &, !, ^, |, &&, ||
- **Relational** : >, <, >=, <=, ==, !=
- **Assignment** : =, +=, -=
- **Primary** : [], new, ., @, ::
- **Unary** : ++, --, typeof, void

Conditional and Looping Statements

Conditional	Looping
Used to execute certain statements based on the condition Example: - if..else - If..else if - switch	Used to execute a block of code repeatedly, a certain number of times Example: - for - for..in - while

Object Oriented Programming in ActionScript

- OOP is basically a way to organize our code with the help of objects
- ActionScript 1.0 and 2.0 provided support for OOP, but was limited
- ActionScript 1.0 used Function Objects, to create constructs similar to classes
- ActionScript 2.0 introduced the concept of classes by adding the keywords ‘class’ and ‘extends’
- ActionScript 3.0 extends over the previous versions and provides better functionalities to developers

Classes(I)

- The heart of all object-oriented code development
- An abstract representation of object
- ActionScript classes have three types of characteristics
  - Properties
  - Methods
  - Events

```
public class Hello {

   public var string:String = "Hello World";

   public function Hello() {
     trace(string);
   }
 }
```

Classes(II)

- Properties
  - Like variables, represent one piece of data
  - Example:

```
public var string:String = "Hello World";
```

Defines a property ‘string’ of the type ‘String’ and initializes it to “Hello World”

- Methods
  - Functions or behaviors that the object performs or exhibits
  - Example:

Defines a method Hello(), which displays the content of ‘string’.

This method is a constructor

```
public function Hello() {
    trace(string);
}
```

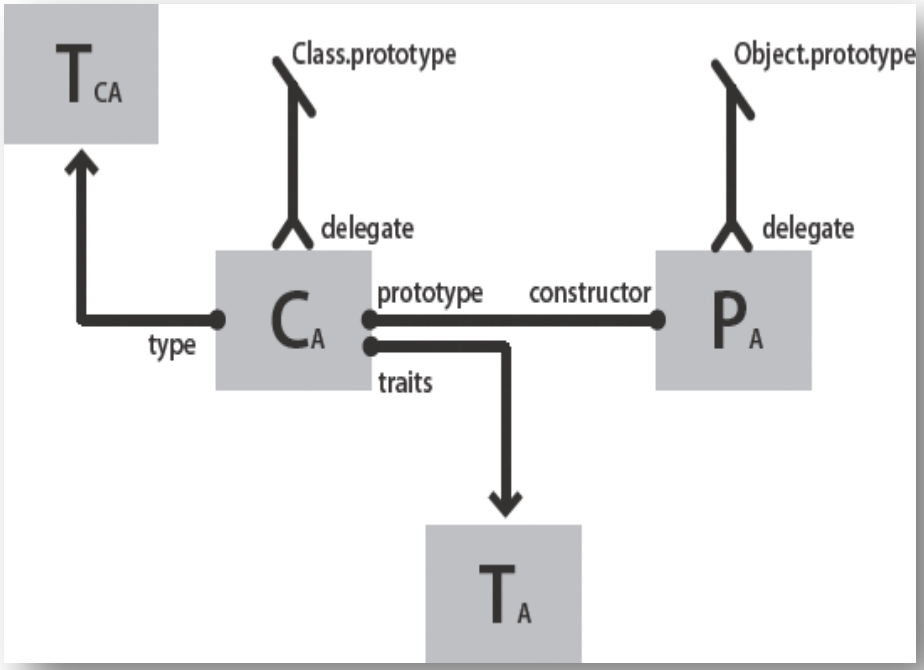
Classes(III)

- **Events**
- Mechanisms that determine which statements need to executed and when
- Events are things that occur and need to be responded to
- For example, a mouse click by user or a key typed in by user
- Following shows the ActionScript code to handle an event

```
function eventResponse(eventObject:EventType):void
{
// Actions performed in response to the event
 }
eventSource.addEventListener(EventType.EVENT_NAME, eventResponse);
```

Class Object

- For every class, ActionScript creates a special class object that allows sharing of both behavior and state
- Though it may not have practical implications for many applications
- The block diagram below represents the structure of a class object, of a class A



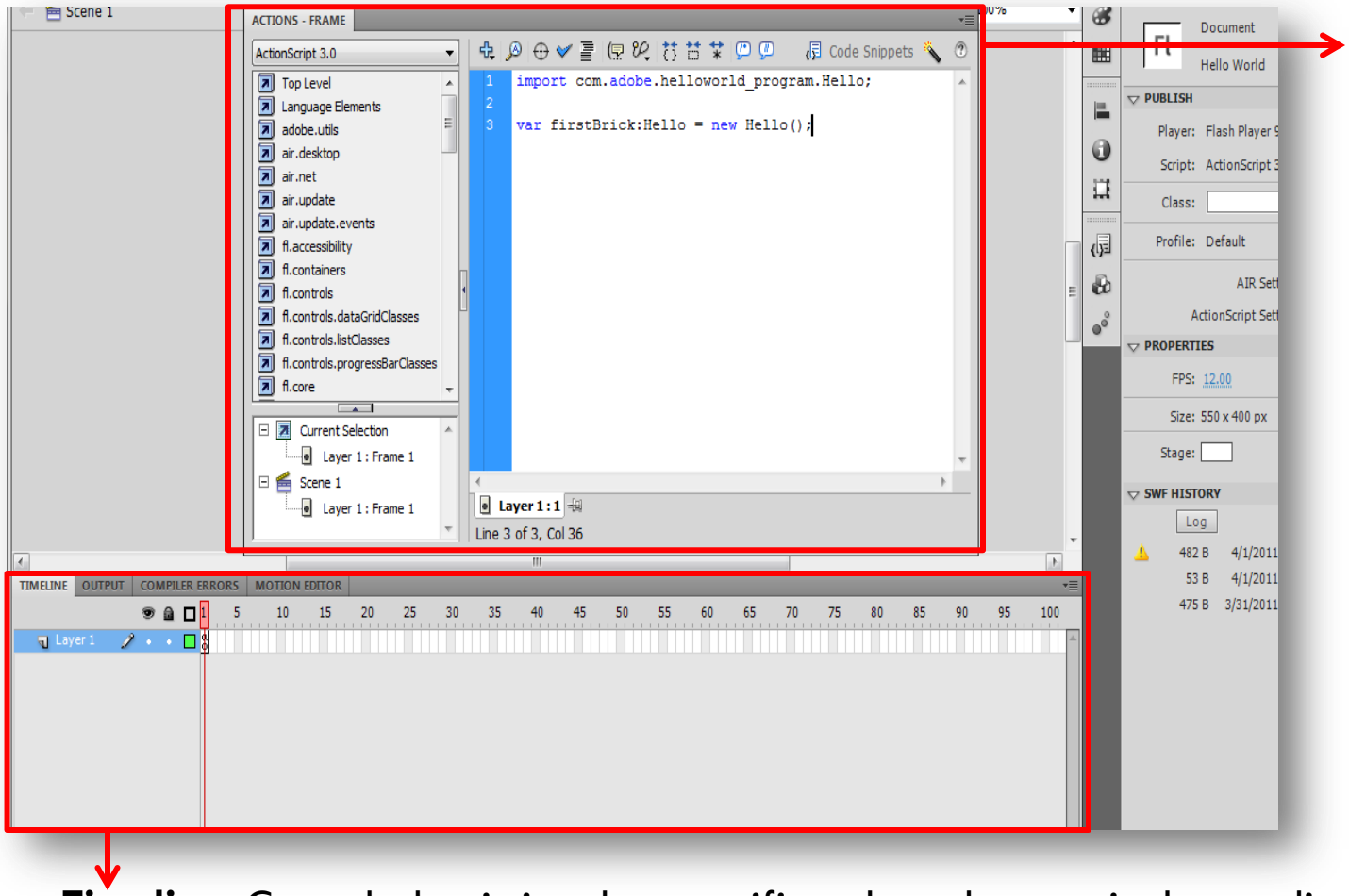
CA-Contains references to other objects
TA-Stores the instance properties
PA-Represents the class object it was attached through constructor
TCA-Represents static properties defined by the class

My First Program: ‘Hello World’

- ActionScript code can be compiled and run in any Flash Authoring environment
- I used Adobe Flash Professional CS5 to run my code
- Flash Professional CS5 provides two tools for writing ActionScript code:

- **Action Window**: Allows to write ActionScript code attached to frames on a timeline
- **Script window**: Dedicated text editor for working with ActionScript (.as) code files

Flash Professional CS5 User Interface



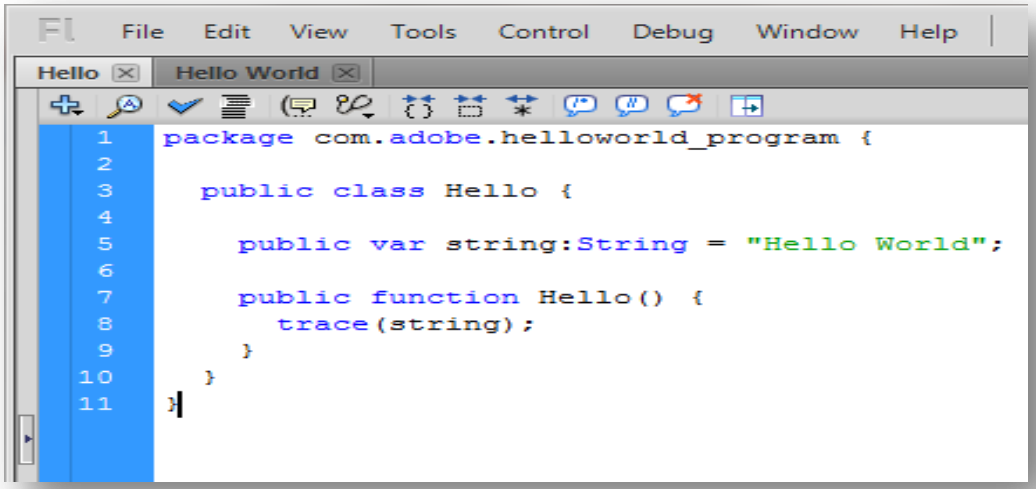
Action Window:
It is text box where we can write our ActionScript code

Timeline: Controls the timing that specifies when elements in the application appear
Since this is a simple program, it contains only 1 layer and 1 frame

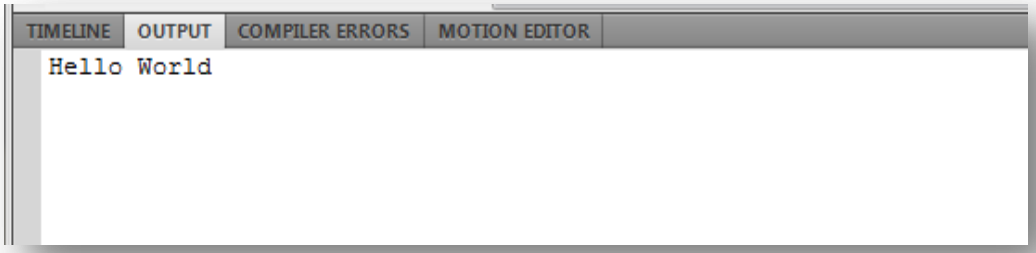
Hello World

- Figure on the right shows the definition of the class 'Hello'
- A property 'string' of type String is defined and using a method(constructor) it is displayed on the Output Window
- On compiling the code, we get the following output on the 'OUTPUT' window

Hello.as file



OUPTUT



OOP Concepts

- ActionScript 3.0 has more functionality, compared to the previous versions, to support the OOP concepts such as:

- **Inheritance** ('final' and 'override' keywords added)
- **Encapsulation** ('protected' and internal' keywords added)
- **Polymorphism**
- **Interfaces** ('IEventDispatcher' interface added)
- **Design Patterns** ('Factory', 'MVC' and other patterns included)

Advanced Topics

Core ActionScript Classes(I)

- AS3 provides many in-built top-level classes that facilitate the development of software applications
- Working with dates and times:

- Classes: **'Date'** and **'Timer'** (in flash.utils package)
- Returns the current date and time, sets date and time, etc. based on the number of parameters passed
- For example:

```
// Sets 'current' to today's date
var current:Date = new Date();
```

Core ActionScript Classes(II)

- Working with strings:
  - Classes: '**String**'
  - Methods of this class can be used for string manipulations such as compare, copy, concatenate, etc.
  - For example:

```
//Capitalizes the first letter of a single word
private function capitalizeFirstLetter(word:String):String{
//Code
}
```

- Working with arrays:
  - Classes: '**Array**'
  - Methods of this class can be used for sorting an array, searching in an array, etc.
  - For example:

```
//Initializes 'object1' to the start of an array
this.object1 = new Array();
```

Core ActionScript Classes(III)

- Using regular expressions:
  - Data Type: **'RegExp'**
  - A regular expression describes a pattern that is used to find and manipulate matching text in strings
  - For example:

```
//Defines the 'pattern', consisting of characters
//A, B, C, followed by any number of digits
var pattern:RegExp = /ABC\d*/;
```

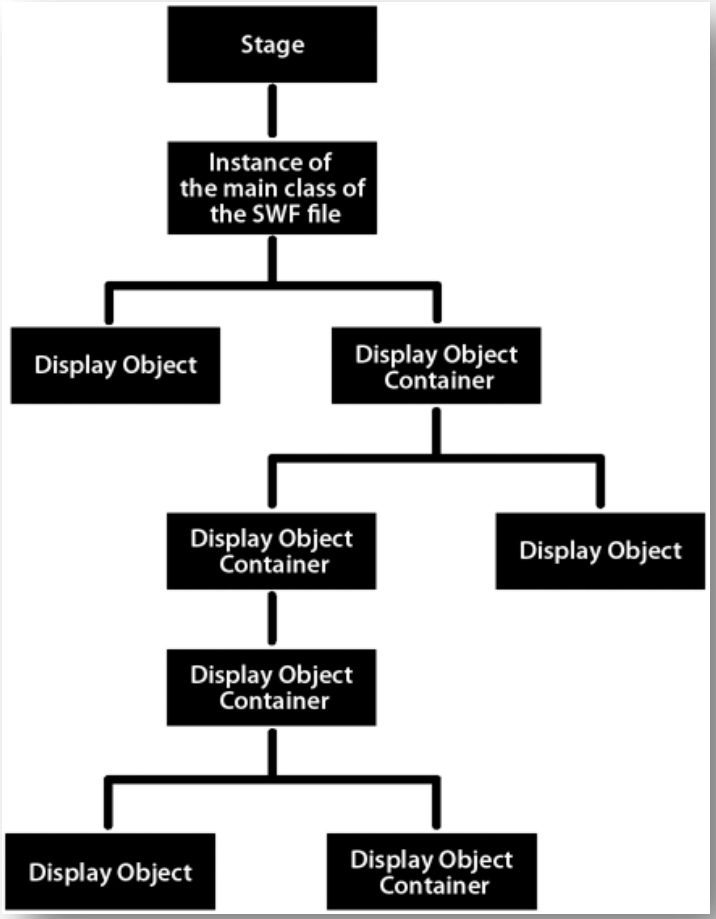
- Working with XML:
  - AS3 provides a group of classes compliant with E4X specification which provide powerful functionality to work with XML data
  - For example:

```
//Defines an XML object 'obj'
public var obj:XML;
```

Display Programming(I)

- ActionScript 3.0 provides support for all the visual elements in an application by using display objects
- Display objects can be worked with to manipulate the elements on the stage, for e.g.: add, move, remove, etc.
- An application has a hierarchy of all the display objects used, known as 'display list'

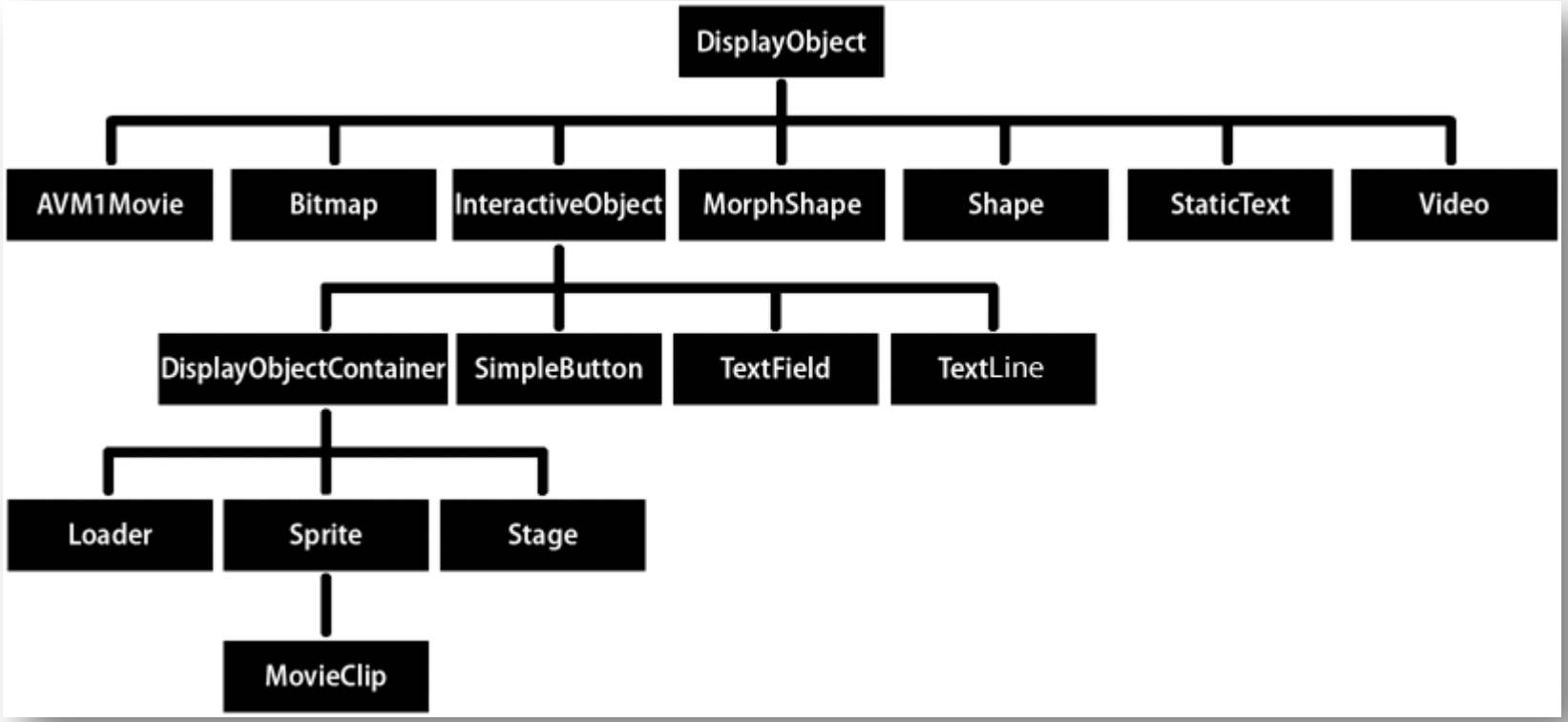
Display List



Display Programming(II)

- AS3 provides a set of core classes to support every visual element that can appear in a Flash Application

Subclass relationships of core display object classes



Working with Movie Clips

- ActionScript 3.0 provides a core 'MovieClip' class, which extends the features of the display object, i.e. includes properties and methods for controlling its timeline
- 'MovieClip' class includes methods such as play(), gotoandPlay(), gotoandStop(), stop(), etc. to handle video clips

```
//Defines and plays a video clip
var video:MovieClip = new MovieClip(event.currentTarget);
video.play();
```

Additional Features

- ActionScript 3.0 provides many more functionalities that are not covered here
- In addition to the once discussed, AS3 provides support for:

- Text Fields
- User Interactions
- Sound Files
- File system
- SQL Databases

and many more…

Another Example:

- In this example, most of the features discussed before are implemented
- Here, a shape is drawn on the stage using the following function

```
43          private function drawAsset(AssetClass:Class, posX:uint, posY:uint):MovieClip {
44              var mc:MovieClip = new AssetClass();
45              mc.transform.colorTransform = getRandomColor();
46              mc.rotation = Math.random() * 360;
47              mc.addEventListener(MouseEvent.CLICK,rotate);
48              mc.addEventListener(MouseEvent.MOUSE_DOWN,startDragAsset);
49              mc.addEventListener(MouseEvent.MOUSE_UP,stopDragAsset);
50              mc.stop();
51              mc.x = posX;
52              mc.y = posY;
53              addChild(mc);
54              return mc;
55          }
```

- User can select the shape using a dropdown list and add it to the stage using the button ‘Add’

```
29          btn1.label = "Add";
30          btn1.addEventListener(MouseEvent.CLICK, addAssetToStage);
```

```
72      private function addAssetToStage(me:MouseEvent):void {
73          var AssetClass:Class = getDefinitionByName(cb.selectedItem.data) as Class;
74          if(AssetClass) {
75              drawAsset(AssetClass, 200, 75);
76          }
```

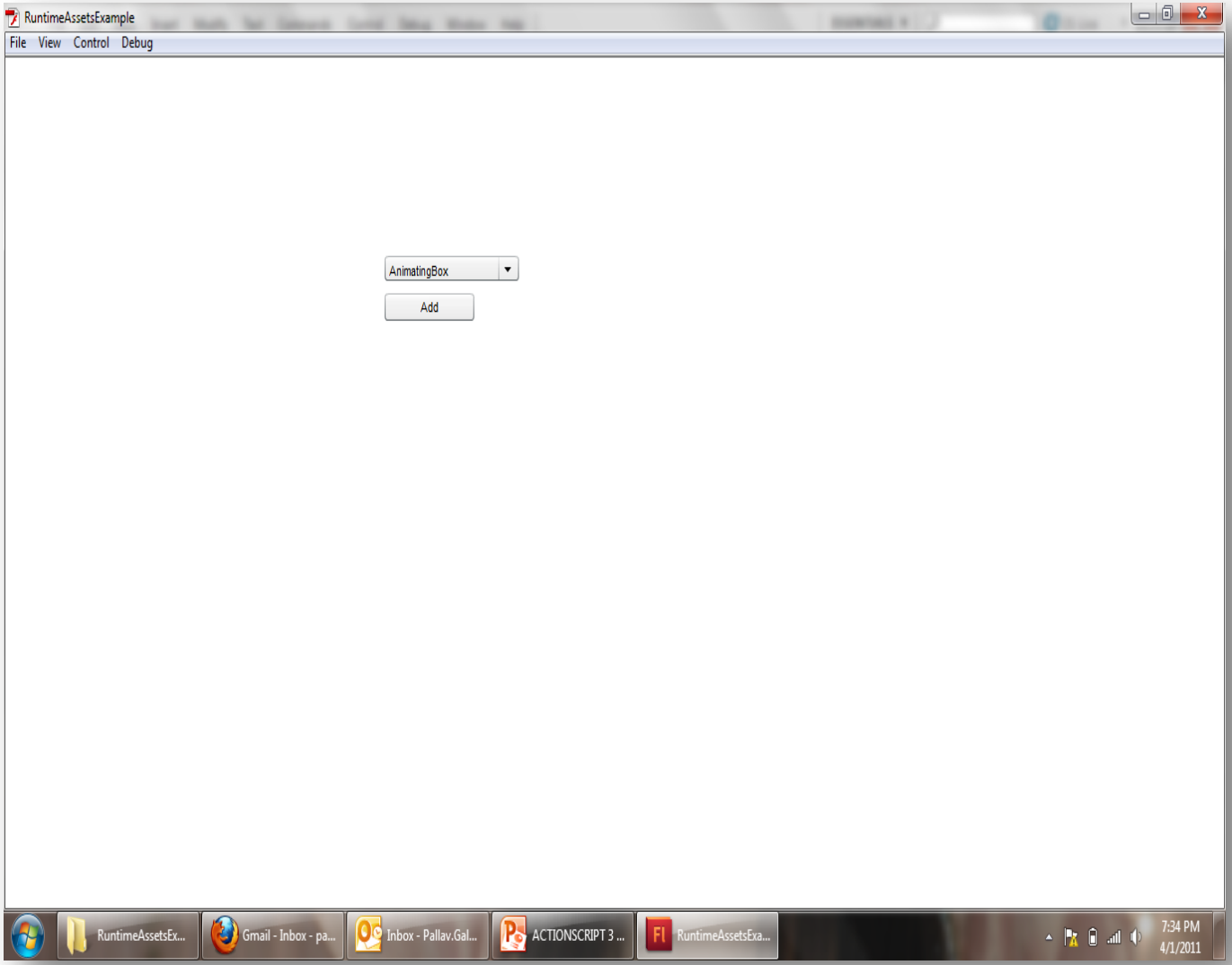
- After the shape is added to the stage, clicking on the shape give a rotating feature to it

```
56          private function rotate(event:MouseEvent):void {
57              var target:MovieClip = MovieClip(event.currentTarget);
58              target.play();
59          }
```

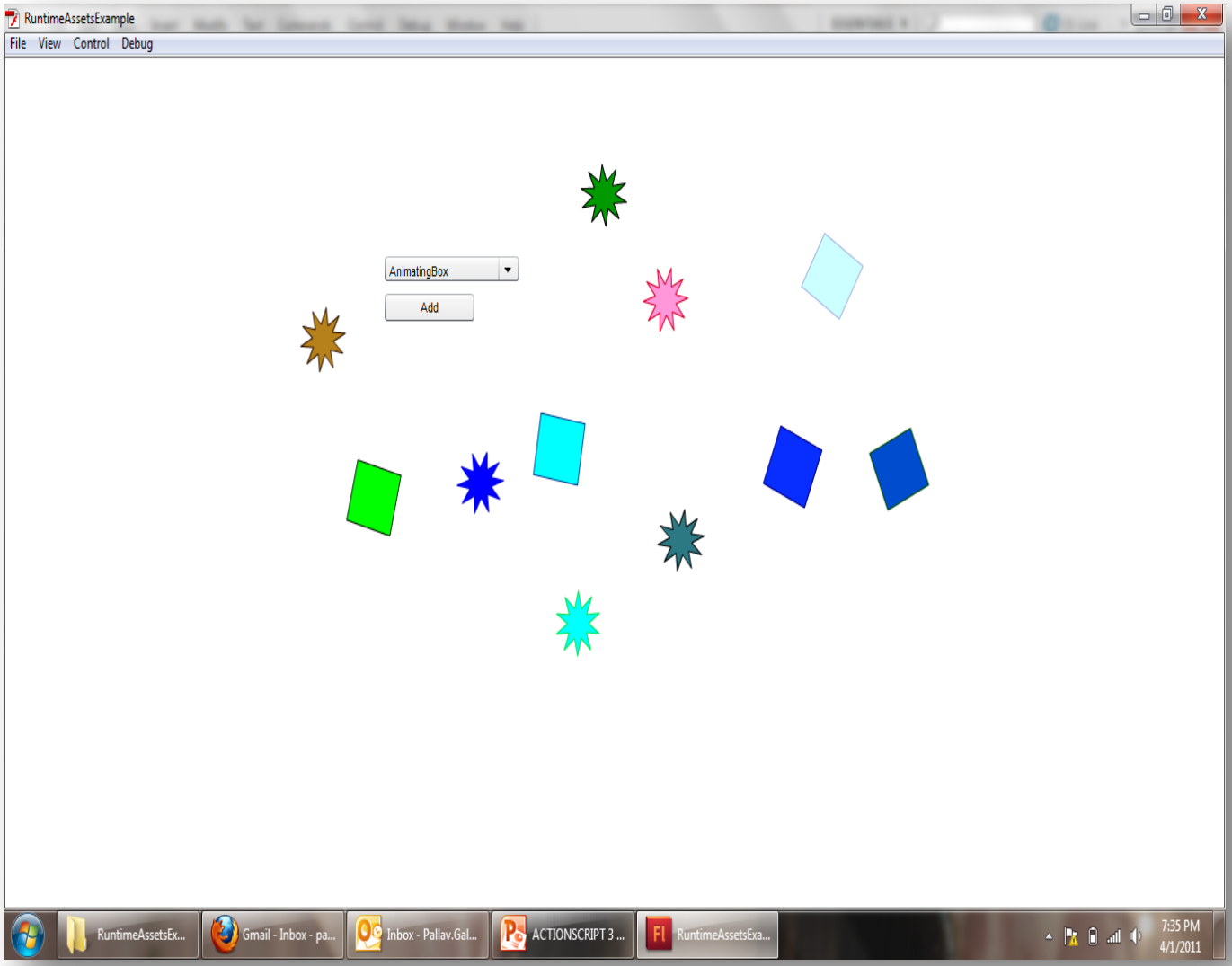
- User can also drag a shape to anywhere on the screen

```
63          private function startDragAsset(event:MouseEvent):void {
64              var target:MovieClip = MovieClip(event.currentTarget);
65              target.startDrag();
66          }
```

Screenshot of the stage before running the code



Screenshot of the stage after running the code



Resources

- **Websites:**

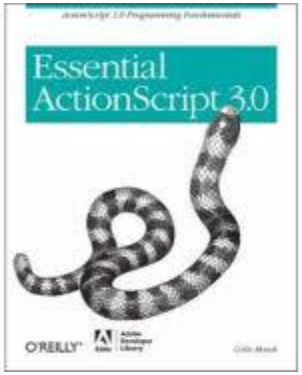
http://www.adobe.com/devnet/actionscript/articles/as3_tour.html

http://www.adobe.com/devnet/actionscript/articles/actionscript3_overview.html

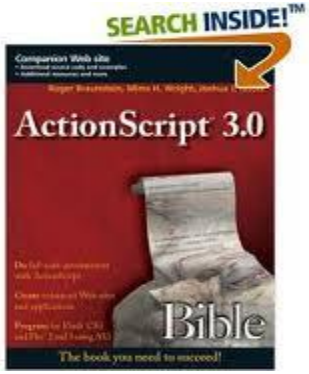
http://www.flashandmath.com/basic/index.html

- **Books:**

Essential ActionScript 3.0



ActionScript 3.0 Bible



QUESTIONS?