

# FLEX



Oliver, Think, and Susie

# What is Flex?

- Open source application framework
- Actionscript
- MXML for UI

---

# Flex History

- Began as closed sourced Macromedia project to develop Flash apps, 2004
- Transfer to Adobe after acquiring Macromedia, 2005
- Open source, 2008
- Adobe donated Flex to the Apache Software Foundation in 2011 – continued support from Adobe

# Design Goals

- Cover a wide range of development projects
- Dynamic, interactive applications
- Easily connect frontend and backend development
- Applications translate to various browsers, OS, and devices
  - Solve JavaScript cross browser inconsistency

# MXML

- XML-based (eXtensible Markup Language)
- Developed by Macromedia, 2004
- Used for UI element layout
- Similar to HTML, but more dynamic and includes more visual elements (ex. tab navigators, accordions, and menus)
- Set up nonvisual components of an application (ex. accessing data sources)
- Rendered by Flash

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
               xmlns:s="library://ns.adobe.com/flex/spark"
               xmlns:mx="library://ns.adobe.com/flex/mx" minWidth="955" minHeight="600">
  <s:layout>
    <s:VerticalLayout/>
  </s:layout>
  <s:Panel backgroundColor="blue">
    <s:Form width="400" height="400" color="white">
      <s:Label text="Hello World!" fontWeight="bold" fontSize="24"/>
    </s:Form>
  </s:Panel>
</s:Application>
```

# ActionScript: Overview

- ActionScript is an object-oriented programming language mainly to target Adobe Flash Player and Adobe AIR
- Supported by Adobe
- It was derived from HyperTalk, a scripting language, by Macromedia
  - Not purely Object Oriented, can have stand-alone functions.
- The 3.0 version was a complete rewrite of ActionScript 2.0
- Is used for development of websites and desktop/mobile applications

# ActionScript: Language Features

- Strongly typed
- Has both compile-time type checking (static typing) and run-time type checking (dynamic typing)
- **Standard mode:** dynamic typing but in **Strict mode:** both dynamic and static typing. Flex uses ActionScript in strict mode by default.
  - Can declare untyped variable even when using strict mode.

```
public function dynamicTest (xParam: Object)
{
    if (xParam is String)
    {
        var myStr: String = xParam; // compiler error in strict mode
        trace("String: " + myStr);
    }
    else if (xParam is Number)
    {
        var myNum: Number = xParam; // compiler error in strict mode
        trace("Number: " + myNum);
    }
}
```

```
public function dynamicTest (xParam)
{
    if (xParam is String)
    {
        var myStr: String = xParam;
        trace("String: " + myStr);
    }
    else if (xParam is Number)
    {
        var myNum: Number = xParam;
        trace("Number: " + myNum);
    }
}
```

# ActionScript: Language Features

- Static scoping
- Unlike Java, blocks do not define new scopes; only function or global scope.

```
public function blockTest (testArray: Array)
{
    var numElements: int = testArray.length;
    if (numElements > 0)
    {
        var elemStr: String = "Element #";
        for (var i:int = 0; i < numElements; i++)
        {
            var valueStr: String = i + ": " + testArray[i];
            trace(elemStr + valueStr);
        }
        trace(elemStr, valueStr, i); // all still defined
    }
    trace(elemStr, valueStr, i); // all defined if numElements > 0
}
blockTest(["Earth", "Moon", "Sun"]);
```

valueStr is defined  
“locally” within the  
for loop but is still  
recognized globally  
within the function



# ActionScript: Language Features

- Memory Management
  - All values are stored as immutable objects
  - Quite similar to Java with “new” (heap-dynamic)
- Support For Abstraction
  - More flexible in standard mode, but abstraction always supported.
  - Provides “is” keyword that checks whether a variable is of a specified data type, an instance of a particular class or implements an interface.
  - “is” = “instanceof” in Java

# ActionScript: Language Features

- Sealed Classes vs. Dynamic Classes
  - “A sealed class possesses only the fixed set of properties and methods that were defined at compile time”
    - You cannot add properties or methods at run time
    - This enables stricter compile-time checking, resulting in more robust programs.
    - Improving memory by not requiring an internal hash table for each object instance.
    - Same as in Java
  - “A dynamic class defines an object that can be altered at run-time by adding or changing properties and methods.”

# ActionScript: Language Features

- Dynamic Class Example:

```
dynamic class Protean
{
    private var privateGreeting:String = "hi";
    public var publicGreeting:String = "hello";
    public function Protean()
    {
        trace("Protean instance created");
    }
}
```

```
var myProtean:Protean = new Protean();
myProtean.aString = "testing";
myProtean.aNumber = 3;
trace(myProtean.aString, myProtean.aNumber); // testing 3
```

# Sample Code

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <mx:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
3     xmlns:mx="library://ns.adobe.com/flex/mx" layout="vertical" minWidth="955" minHeight="600">
4     <fx:Declarations>
5         <!-- Place non-visual elements (e.g., services, value objects) here -->
6     </fx:Declarations>
7
8     <fx:Script>
9         <![CDATA[
10             import mx.controls.Alert;
11
12             private function onClickName():void{
13                 if (userName.text == "") {
14                     Alert.show("Please provide a name");
15                 } else {
16                     Alert.show("Hello, " + userName.text);
17                 }
18             }
19
20             private var die:Die = new Die(6);
21             private function onClickDie():void {
22                 dieImage.load("../assets/die" + die.roll() + ".gif");
23             }
24         ]]>
25     </fx:Script>
26
27     <mx:Panel title="Hello You!">
28         <mx:Form width="200">
29             <mx:Label text="Name:" />
30             <mx:TextInput id="userName" />
31             <mx:Button label="Press Me" click="onClickName()" />
32         </mx:Form>
33     </mx:Panel>
34
35     <mx:Panel title="Die">
36         <mx:Form>
37             <mx:Image id="dieImage" source="../assets/die1.gif" />
38             <mx:Button label="Roll Die" click="onClickDie()" />
39         </mx:Form>
40     </mx:Panel>
41 </mx:Application>
```

# Sample Code

The screenshot displays the Adobe Flash Builder 4.7 IDE interface. The main editor window shows the source code for a class named `Die` in the `helloYou` package. The code is as follows:

```
1 package
2 {
3     public class Die {
4         private var numSides:int;
5
6         public function Die(sides:int) {
7             this.numSides = sides;
8         }
9
10        public function getNumberOfSides():int {
11            return this.numSides;
12        }
13
14        public function roll():int {
15            return Math.floor(Math.random() * this.numSides + 1);
16        }
17    }
18 }
```

The Package Explorer on the left shows the project structure, including the `helloYou` package and its sub-packages (`src`, `Flex 4.6.0`, `assets`, `bin-debug`, `html-template`, and `libs`). The Outline view at the bottom left shows the class `Die` and its members: `numSides : int`, `Die(int)`, `getNumberOfSides() : int`, and `roll() : int`.

# Flex: Real World Applications

- Creating Rich Internet Applications
  - Mostly anything that uses Flash to run
    - Flex creates .swf files that are run using Flash in the browser.
  - Useful for animations, use of camera, games, etc...
- Examples:
  - Blueline's record function
  - MIT Scratch's final products are compiled as Flex files
  - According to Wikipedia, HBO Go and Yahoo Messenger use it too.

# Steve Jobs' "Thoughts on Flash"

- "Open"
- "Full Web"
- Reliability, security and performance
- Battery life
- Touch
- "The most important reason"

"New open standards created in the mobile era, such as HTML5, will win on mobile devices (and PCs too). Perhaps Adobe should focus more on creating great HTML5 tools for the future, and less on criticizing Apple for leaving the past behind."

# Sources

[Adobe Flex](#)

[About MXML](#)

<http://www.adobe.com/devnet/actionscript/learning/as3-fundamentals/data-types.html>

[Apache Flex](#)

[Justin Mclean - Intro to Flex](#)

[About ActionScript](#)