Making the Leap to ActionScript
Phillip Kerman

annotated presentation and links will be available at:
`www.phillipkerman.com/fc02/`

<u>Overview</u>:

Walking across the bridge from competent Flash user to ActionScript programmer takes more than a simple desire. You'll be entering a world unlike any you've seen before. And even though the language spoken (ActionScript) is unfamiliar and the rules seem overly strict, your basic goal—to gain control over Flash—is clear. This workshop introduces ways take your programming tasks, narrow them down to specific goals, and translate them to ActionScript. After all, programming is nothing more writing instructions. It's just that learning how to think like a programmer and write instructions in a new language can be difficult when you start.

<u>Outline</u>:

Instructions

Scripts are nothing more than instructions. Although you'll need to translate those instructions to ActionScript, you always begin with an objective.

Events to trigger scripts

After clarifying the script, you can specify *when* that script should execute. Events include user interactions like clicking a button as well as timed events like "every 2 seconds".

Expressions vs. Statements

Think of expressions as phrases where statements are complete sentences. Statements "do" things where expressions simply have a value. For example, "slow as molasses" is an expression but "you walk as slow as molasses" is a statement.

Begin writing expressions and statements with pseudo-code. This way you'll ensure the script makes sense before translating it to real code.

Dynamic expressions are more powerful (than explicit ones) as they adapt to changes. For example, "two meters longer than your car" is always enough room to park no matter how large your car.

Flash specific helpers

In order to let Flash help you with its "auto-complete" feature, learn the default suffixes for objects in Flash (like "_mc" for movie clip). Other features such as "code-hints" and the reference panel are also quite powerful.

Outline (continued):

Variables

Variables provide a safe—yet temporary—way to store data for later use. (The new Local Shared Objects let you store variables on the user's computer for later retrieval.) Every variable has a name and a value. Like children you can have as many variables as you wish—you just have to name them and take care of them.

Properties

Changing properties is the most common way to cause a visual change onscreen. Not only can you change properties while authoring (say, of clips like **_alpha**, or **_x**, or **_y**) but—by using script—you can affect properties during runtime. Realize that every clip instance (and now buttons and text fields) maintain their own set of properties.

Here are examples of setting the **_alpha** property of a clip instance called **clip** using a statement (**clip._alpha=10**) and referring to a property within an expression (**clip._alpha=*clip._alpha+10***)—just the part right of the "=" is an expression.

Addressing

Even once movie clips are nested, you'll often need address a specific clips by instance name. You can either use an address that's relative or absolute. Neither one is "better" though relative addresses can automatically adapt to changes in the movie hierarchy. An example of an relative address is "the person next door". In comparison, "the person in the house at 123 Main St." is absolute.

Dot syntax

Addressing clips, variables, and properties all use dot syntax in the form: *address.object.property* where "address" goes from "general to specific" (europe.germany.stuttgart); "object" is the clip you're addressing; and "property" is the variable or property you're referencing. For example: europe.germany.stuttgart.mayor.age refers to the "age" property of the "mayor" object in Stuttgart Germany. (As this is just an expression, nothing is changing.)

Where to go from here:

if-statements
for-loops
built-in functions (particularly getTimer())
core objects (array, math, string)
homemade functions
movie objects (color, sound, textField, textFormat)
homemade objects/classes/properties.
(You'll get time to practice these topics in my workshop next week.)

My books (`www.phillipkerman.com/books`):

Sams Teach Yourself Macromedia Flash MX in 24 Hours (Sams 2002)
ActionScripting in Flash MX (NewRiders 2002)