ACTION SCRIPT 3

LEVEL

BASICS ☑

INTERMEDIATE

ADVANCED

25 TIPS TO GET YOU STARTED

FREE eBOOK

TIP 1

Variable Types

ActionScript 3 supports a wide range of variable types including some which were not present in previous versions of ActionScript. Basic types for AS3 include:

Primitive:

* Number
* String
* Boolean
* int
* uint
* null
* undefined

Complex:

* Object
* Array
* Date
* Function
* RegExp
* XML
* XMLList
* Error

There's a new * type that is used to represent any data type. This should be used instead of ommitting typing information for your variables.
For example, var anything:*;

TIP 2

ActionScript 3 now has a new collection of "display objects" which includes those objects that can be seen on the screen or added to the "display list." AS3 display objects include

* AVM1Movie
* Bitmap
* Loader
* MorphShape (cannot create them via ActionSscript)
* MovieClip
* Shape
* SimpleButton
* Sprite
* StaticText (cannot create them via ActionSscript)
* TextField
* Video

TIP 3

Declare types for all variables, parameters, and return values.

* It is considered best practice

* It will help the compiler give you more helpful error messages

* It also increases runtime performance because the virtual machine will know the types you're working with ahead of time.

Next wave
Multimedia

TIP 4

The default access specifier for declarations is now internal instead of public, meaning that the definition is visible only to the package containing the definition, not to all code.

TIP 5

Use package declarations to put a class definition into a package. The package keyword is new to ActionScript 3.0.

TIP 6

Dereferencing a null or undefined reference will throw an exception

TIP 7

Flash Player API has been reorganized into packages, for example; MovieClip is now `flash.display.MovieClip` and `getTimer` and `setInterval` have been moved to the `flash.utils package.`

Next wave
Multimedia

TIP 8

Use the new Timer class instead of setInterval/setTimeout.

TIP 9

Visual elements must extend DisplayObject, and you can define them like any other class. Visual elements are now created dynamically with new and added to the display list using addChild or addChildAt. Visual entities, including TextField, can be instantiated like any other object and simply added to a display list using addChild or addChildAt.

TIP 10

The root object of a SWF file can now be an instance of a custom class of your choice. In ActionScript 2.0, the root object of a SWF file was always of class MovieClip. In ActionScript 3.0, it may be any subclass of Sprite. When it's loaded, the SWF file will instantiate that class to serve as its root object.

TIP 11

Multiple Arguments in trace() statement,
for example; trace(value1, value2, value3);

Next wave
Multimedia

TIP 12

Changing the frame-rate of your movie dynamically

```
stage.frameRate = 25;
```

TIP 13

Accessing FlashVars,

```
root.loaderInfo.parameters.myVar;
```

TIP 14

With Transparent SWF file (that is, wmode = transparent), special characters cannot be inputted in the Flash text field.

TIP 15

Use TextField.appendText() which is faster and more efficient, for example;

```
var myText:TextField = new TextField();
myText.text = "Hello";
myText.appendText(" world");
```

Next wave Multimedia

TIP 16

Speed up the search in an array by using, Array.indexOf() or Array.lastIndexOf()

TIP 17

Close Net Connections - You can abort loading requests or process made by the player. For example, if you started loading a 50MB swf file into the Flash player but wanted to stop it when the user requested different content or time out

```
var loader:Loader = new Loader();
var request:URLRequest = new URLRequest("assets.
swf");
loader.load(request);
addChild(loader);

// abort loading if not loaded in 5 seconds
var abortID:uint = setTimeout(abortLoader, 5000);

loader.contentLoaderInfo.addEventListener(Event.COM-
PLETE, abortAbort);

function abortLoader(){
    try {
        loader.close();
    }catch(error:Error) {}
}
function abortAbort(event:Event){
    clearTimeout(abortID);
}
```

Next wave Multimedia

TIP 18

Scale and Alpha Ranges
ActionScript 2.0 I ActionScript 3.0
_xscale: 0 – 100 I scaleX: 0 - 1
_yscale: 0 – 100 I scaleY: 0 - 1
_alpha: 0 – 100 I alpha: 0 – 1

TIP 19

ActionScript 3 supports regular expressions. The implementation is much similar to JavaScript.

For example;

```
var temp1:RegExp = new RegExp("\\w+", "i");
var temp2:RegExp = /\w+/i;
```

RegExp methods include:

* RegExp.exec()
* RegExp.test()

String methods that work with regular expressions include:

* String.match()
* String.replace()
* String.search()

TIP 20

ActionScript 3 now allows you to detect when the mouse has left the flash movie using the stage's mouseLeave event. for example, `stage.addEventListener(Event.MOUSE_LEAVE, yourFunction);`

TIP 21

Preventing cache in Flash Player,
Add the following at the end of the call:

```
'?ignoreCache='+new Date().getTime();
```

For example, if we want load assets.xml:

```
'http://www.yourdomainname.com/assets.xml?ignore-
Cache ='+new Date().getTime();
```

TIP 22

When you have a dynamic text field inside a Movie clip that you want to use as a button and you set,

```
my_mc.buttonMode = true;
```

So the hand cursor appears when you hover over it, make sure you set the following attributes to the text field:

```
my_mc.mouseChildren = false;
my_text.selectable = false;
```

This will disable the mouse for the children of my_mc. You could also use

```
my_text.mouseEnabled = false;
my_text.selectable = false;
```

TIP 23

`hitTest()` in Actionscript 2.0 is classified as `hitTestObject()` and `hitTestPoint()`

TIP 24

Capturing keyboard input
Display objects that inherit their interaction model from the InteractiveObject class can respond to keyboard events by using event listeners. For example, you can place an event listener on the Stage to listen for and respond to keyboard input. In the following code, an event listener captures a key press, and the key name and key code properties are displayed:

```
function reportKeyDown(event:KeyboardEvent):void
{
    trace("Key Pressed: " + String.fromCharCode(event.
charCode) + " (character code: " + event.charCode + ")");
}
stage.addEventListener(KeyboardEvent.KEY_DOWN,
reportKeyDown);
```

TIP 25

Masking the display object
To indicate that a display object will be the mask for another display object, set the mask object as the mask property of the display object to be masked:
// Make the object maskSprite be a mask for the object mySprite.

```
mySprite.mask = maskSprite;
```