

flash-cs3第十章PPT

# 10.2 ActionScript 语法规则

要使ActionScript语句能够正常运行，就必须按照正确的语法规则进行编写。

## 1. 字母的大小写

在ActionScript中除了关键字区分大小写外，其他的ActionScript语句大小写可以混用，但根据书写规范进行输入，可以使ActionScript语句更容易阅读。



## 2. 点语法

在ActionScript语句中，点“.”用于指示与对象相关的属性或方法。通过点语法可以引用类的属性或方法，例如：

```
//新建一个空对象，将其引用赋值给变量Company  
var Company:Object = {};  
//新增一个属性name，将字符串“企鹅”赋值给它  
Company.name = "企鹅";  
//输出“企鹅”  
trace (Company.name);
```

### 3. 括号和分号

在ActionScript中大括号“{}”、小括号“()”和分号“;”各有其作用。

- **大括号：**用于将代码分成不同的块。例如：

```
var a:int = 5;           //声明一个int型变量a并为其赋值5
if(a>0){                //如果a大于0
    trace ("正数");      //输出“正数”
} else {                 //否则
    trace ("负数");      //输出“负数”
}
```

- **小括号：**通常用于放置使用动作时的参数，在定义或调用函数时都要使用小括号。例如：

```
trace ("读者你好！"); //输出“读者你好！”
```

- **分号：**在ActionScript语句的结束处添加分号，表示该语句结束。虽然不添加分号也可以正常运行ActionScript语句，但使用分号可以使ActionScript语句更易于阅读。



## 4. 注释

在ActionScript语句的后面添加注释有助于用户理解动作脚本的含义，以及向其他开发人员提供信息。添加注释的方法是先输入两个斜杠“//”，然后输入注释的内容即可。注释以灰色显示，长度不受限制，也不会影响ActionScript语句的执行。

## 5. 标识符和关键字

现实生活中，所有事物都有自己的名字，从而与其他事物区别开。在程序设计中，也常常用一个记号对变量、方法和类等进行标识，这个记号就称为标识符。

标识符的命名必需符合一定的规范，在ActionScript语言中，标识符的第一个字符必须为字母、下划线“\_”或美元符号“\$”，后面的字符可以是数字、字母、下划线或美元符号。

ActionScript保留了一些单词用于特定用途，例如变量、函数或标签名称，它们不能用作标识符，我们称其为关键字。

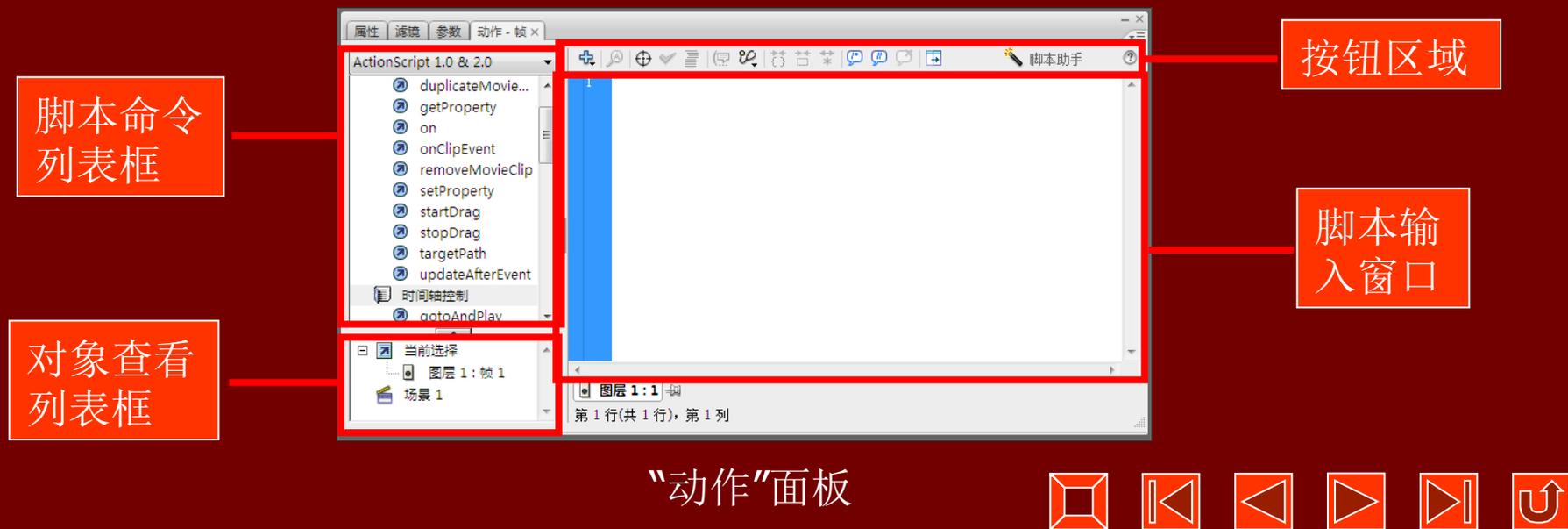
as	break	case	catch	false	class	const	continue
default	delete	do	else	extends	false	finally	for
function	if	implements	import	in	instanceof	interface	intemal
is	naitve	new	null	package	private	protected	pubic
return	super	switch	this	throw	to	true	try
typeof	use	var	void	while	with		

# 10.3 输入代码的两种方法

## 10.3.1 在关键帧中输入代码

在Flash CS3中，可以为时间轴、按钮元件或影片剪辑元件内的任意关键帧添加代码，播放动画时播放头移动到添加代码的关键帧即可执行该代码。要为关键帧添加代码，可先选中关键帧，然后选择“窗口”>“动作”菜单或按【F9】键，打开“动作”面板。

“动作”面板主要由三部分组成，其中，动作工具箱分类存放着ActionScript的大部分语句；脚本导航器中列出了当前选定对象的名称和位置等属性；“脚本”窗格用来供用户输入和编辑代码。

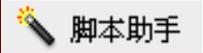


“动作”面板



在“动作”面板上方“按钮区域”中的按钮，有些需要输入代码后才会被激活，它们各自的作用如下。

- **“将新项目添加到脚本中”按钮** ：单击“将新项目添加到脚本中”按钮可在展开的下拉菜单中选择需要添加的ActionScript语句。 
- **“查找”按钮** ：单击“查找”按钮可查找指定的字符串并对指定的字符串进行替换 
- **“插入目标路径”按钮** ：单击“插入目标路径”按钮可在编辑语句时插入一个目标对象的路径。
- **“语法检查”按钮** ：单击“语法检查”按钮可检查当前语句的语法是否正确，并给出提示。
- **“自动套用格式”按钮** ：单击“自动套用格式”按钮可使当前语句按标准格式排列。
- **“显示代码提示”按钮** ：当鼠标光标定位到某某一位置时，单击“显示代码提示”按钮可显示所在语句的语法格式和相关的提示信息。
- **“调试选项”按钮** ：单击“调试选项”按钮可对当前语句进行调试。

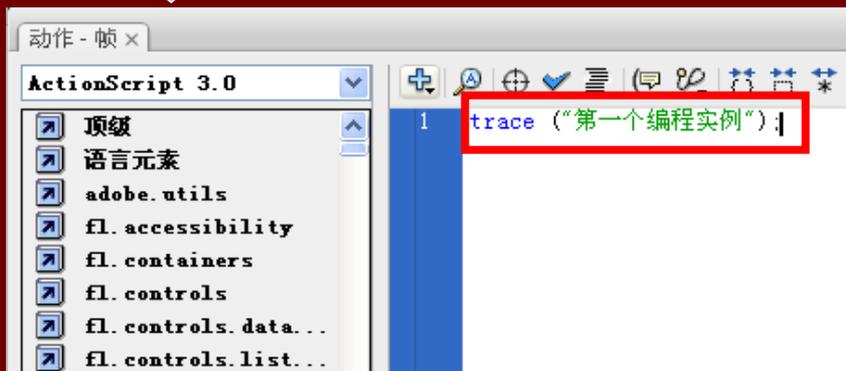
- “**折叠成对大括号**”按钮 ：单击“折叠成对大括号”按钮可将大括号中的语句折叠起来。
- “**折叠所选**”按钮 ：单击“折叠所选”按钮可将选中的语句折叠起来。
- “**展开全部**”按钮 ：单击“展开全部”按钮可将折叠的语句全部展开。
- “**脚本助手**”按钮 ：单击“脚本助手”按钮可开启或关闭“脚本助手”模式。当开启“脚本助手”模式时为对象添加语句，Flash会自动安排语句格式。

在“动作”面板中可以通过下列任意一种方法添加  
ActionScript语句。

- 展开“动作”面板的脚本命令列表框，双击需要的语句进行添加。
- 展开“动作”面板的“脚本命令列表框”，将需要的语句拖到“脚本输入窗口”。
- 直接在“脚本输入窗口”中输入语句。
- 单击“动作”面板“按钮区域”中的“将新项目添加到脚本中”按钮，从展开的下拉菜单中选择要添加的动作语句。

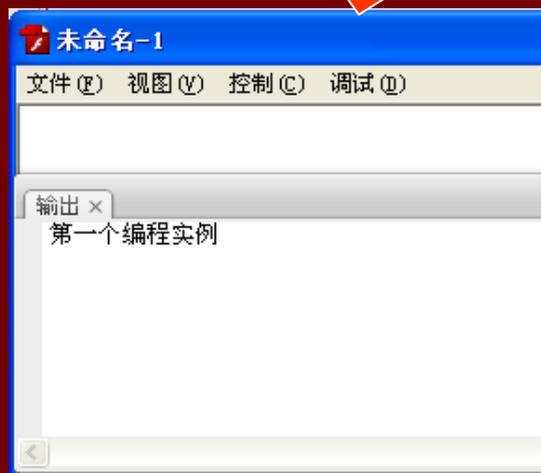
下面以一个简单的实例介绍在在关键帧中输入代码的具体操作。

1.新建一个Flash（ActionScript 3.0）文档，单击选中“图层1”的第1帧，然后按【F9】键开“动作”面板，并在右侧的“脚本输入窗口”中输入代码。



在“动作”面板中输入代码

2.按快捷键【Ctrl+Enter】测试影片，会在“输出”面板中显示“第一个编程实例”



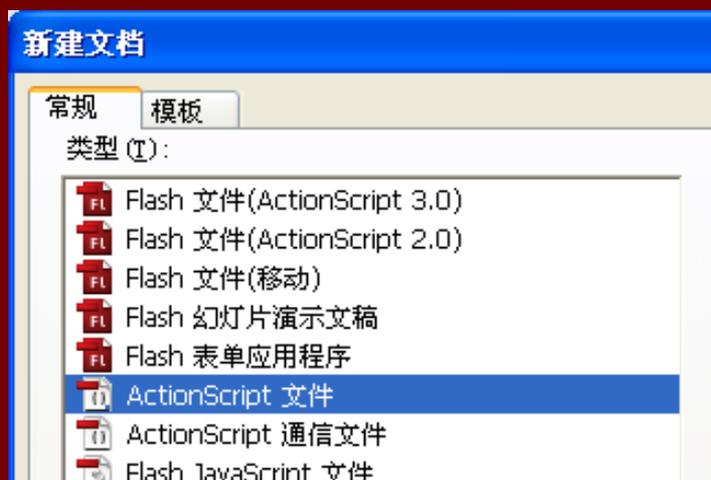
“输出”面板中的文字



## 10.3.2 创建单独的ActionScript文件

ActionScript 3.0可以将外部的ActionScript类文件和fla文件绑定，编译时将SWF文件看成是这个类的一个实例。

当构建较大的应用程序或包含重要的ActionScript代码时，最好在单独创建的ActionScript源文件（扩展名为as的文本文件）中组织代码，从而方便程序的维护。



1. 新建一个Flash (ActionScript 3.0) 文档，并将其保存为“first fla”。然后创建一个ActionScript文件。

创建ActionScript文件

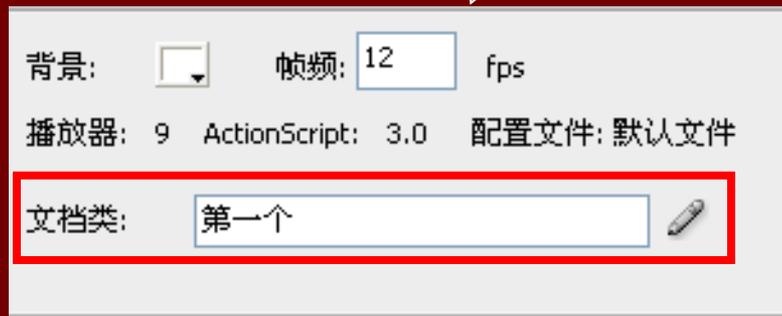


2. 将新建的ActionScript文件存为“第一个.as”，并和“first.fla”文件放在同一目录，然后在“第一个.as”文档中输入代码。

```
1 package{
2     import flash.display.MovieClip;
3     public class 第一个 extends MovieClip{
4         public function 第一个(){
5             trace ("第一个编程实例");
6         }
7     }
8 }
```

在ActionScript文件中输入代码

3. 切换回“first.fla”文档，在“属性”面板的“文档类”编辑框中输入文字“第一个”。



在“文档类”编辑框中输入文件名

4. 对“first.fla”文档进行保存，并按快捷键【Ctrl+Enter】测试影片，会看到一个空白的Flash Player窗口，和一个“输出”面板，面板中显示“第一个编程实例”。



# 10.4 ActionScript 3.0的基本元素

## 10.4.1 数据类型

ActionScript 3.0中的数据可分为简单数据类型和复杂数据类型。简单数据类型表示单条信息，如单个数字或单个文本序列。在ActionScript 3.0中常用的简单数据类型主要有布尔值（**Boolean**）、数字（**int**、**uint**、**Number**）和字符串（**String**）三种：

- **Boolean**: 用来表示真假的数据类型，它只有**true**（真）和**false**（假）两个值。
- **int**、**uint**、**Number**: 它们都是表示数字值的数据类型，其中**int**和**uint**表示整数型数值（**uint**为不带负号的整数）；**Number**表示浮点数，浮点数在计算机中用来近似表示任意某个数值。
- **String**: 表示Unicode字符集中的符号，即通常说的文本型数据，例如一个名称或书中某一章的文字。

ActionScript 3.0中定义的大部分数据类型都可以被描述为复杂数据类型，因为它们表示组合在一起的一组值。常用的复杂数据类型有：

- **MovieClip:** 影片剪辑元件。
- **TextField:** 动态文本字段或输入文本字段。
- **SimpleButton:** 按钮元件。
- **Date:** 日期和时间。



## 10.4.2 变量

在ActionScript中变量用于存储信息，它可以在保持原有名称的情况下使其包含的值随特定的条件改变。

### 1. 声明变量的语法

一个变量包含变量名、数据类型和值3个部分。可以在声明变量时为变量赋值，变量的值可以是字符串、数字、数组、对象、xml和日期，也可以是自己创建的自定义类。在ActionScript 3.0中声明变量的格式如下。

```
var 变量名: 数据类型;  
var 变量名: 数据类型 = 值;
```

**var**是一个关键字，用来声明变量。变量的数据类型写在冒号后。若要为变量赋值，则值的数据类型必须和变量的数据类型一致；若不为变量赋值，则变量将根据数据类型赋予默认值。例如：

```
var M:int;  
var N:int = 100;
```

其中“**var M:int**”声明了一个**int**型变量但没有赋值，所以会使用默认值；“**var N:int = 100**”声明了一个**int**型变量，并为其赋值“**100**”。

## 2. 变量的命名规则

变量的命名规则不仅是为了让编写的代码符合语法，更重要的是增强代码的可读性，标准清晰的命名是优秀代码的必备条件。在为变量命名时应遵循以下规则。

- 变量名必须符合标识符命名原则。
- 在一个Flash文件中变量名必须是唯一的。
- 变量名不能是关键字或ActionScript文本。这里的ActionScript文本是指直接出现在代码中的值，如true、false、null或undefined等。
- 变量名区分大小写。如果一个变量名由多个单词构成，第一个单词为小写，第二个单词首字母大写，如anInt。
- 变量不能是ActionScript语法中的任何元素，例如类名称。

### 3. 变量的默认值

若声明了一个变量，却没有为其赋值，则系统会自动为该变量赋予一个默认值，该默认值取决于变量的数据类型。

数值类型	默认值
Boolean	false
Int	0
Number	NaN
Object	null
String	null
Uint	0
未声明（与类型注释*等效）	undefined
其他所有类（包括用户自定义的类）	null

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/726004123031010232>