

## AS3.0 百条实用小知识

【鼠标坐标】

mouseX mouseY

【检查变量类型并返回布尔值】

is

【检查变量类型并返回类型】

typeof

【检查对象类型并返回该对象】

as

【是数字但不是有效数字问题】

```
var quantity:Number = 15 - "rabbits";
```

```
trace(typeof quantity); //显示: "number" , 但它是 NaN (not a number)
```

```
trace(quantity is Number); //true
```

```
trace(quantity != NaN); //false
```

```
//使用 isNaN()函数检测:
```

```
isNaN(quantity); //true
```

```
//检测变量是否内含有效数字:
```

```
!isNaN(quantity); //false
```

【Timer 类注意事项】

不要认为 Timer 可以极其准确;使用 Timer 时间间隔不要低于 10 毫秒.

【检测播放器版本】

Flash.system.Capabilities.version

对于 8.5 版以前的任何 Flash Player 版本,这种方法都不适用.

**【判断客户端系统】**

Flash.system.Capabilities.os

**【检测播放器类型】**

Flash.system.Capabilities.playerType

可能的值有:

"StandAlone", 用于独立的 Flash Player

"External", 用于外部的 Flash Player 或处于测试模式下

"PlugIn", 用于 Flash Player 浏览器插件

"ActiveX", 用于 Microsoft Internet Explorer 使用的 Flash Player ActiveX 控件

**【检测系统语言】**

Flash.system.Capabilities.language

**【判断用户是否启用了 IME(输入法编辑器)】**

Flash.system.IME.enabled

**【检测屏幕的分辨率】**

Flash.system.Capabilities.screenResolutionX

flash.system.Capabilities.screenResolutionY

**【把弹出窗口居中的算法】**

$X = (\text{舞台宽}/2) - (\text{窗口宽}/2)$

$Y = (\text{舞台高}/2) - (\text{窗口高}/2)$

**【控制影片配合 Player 的方式,包括缩放问题】**

stage.scaleMode

可供选择值:Flash.display.StageScaleMode

【舞台的对齐方式】

stage.align

可供选择值:Flash.display.StageAlign

【隐藏 Flash Player 的右键菜单】

stage.showDefaultContextMenu = false;

【检测系统是否具有音频功能】

Flash.system.Capabilities.hasAudio

【检测播放器是在具有 MP3 解码器的系统上运行,还是在没有 MP3 解码器的系统上运行】

Flash.system.Capabilities.hasMP3

【检测播放器能 (true) 还是不能 (false) 播放流式视频】

Flash.system.Capabilities.hasStreamingVideo

【检测播放器是在支持 (true) 嵌入视频的系统上运行,还是不支持 (false) 嵌入视频的系统上运行】

Flash.system.Capabilities.hasEmbeddedVideo

【检测播放器能 (true) 还是不能 (false) 对视频流(如来自 Web 摄像头的视频流)进行编码】

Flash.system.Capabilities.hasVideoEncoder

【显示 Flash Player 中的"安全设置"面板】

flash.system.Security.showSettings();

可供选择项:flash.system.SecurityPanel

【数字的不同进制之间的转换】

parseInt(str:String, radix:uint = 0):Number 返回十进制的数,参数 radix 表示要分析的数字的基数.如果省略 radix,就默认为 10,除非字符串的开头是"0x","0X"或"0":

```
trace(parseInt("0x12")); //设定 radix 为 16,输出:18
```

```
trace(parseInt("017")); //设定 radix 为 8,输出:15
```

或者使用 Number,uint 和 int 对象的 toString(radix)方法.

【使用 Math.round()对一个数取整,四舍五入】

```
Math.round()
```

```
trace(Math.round(204.499)); //输出:204
```

```
trace(Math.round(401.5)); //输出:402
```

【使用 Math.floor()对一个数向下取整,就是只要整数部分而不理会小数分】

```
trace(Math.floor(204.99)); //输出:204
```

【使用 Math.ceil()对一个数向上取整,只要小数部分不为零,整数部分就加 1】

```
trace(Math.ceil(401.01)); //输出:402
```

【产生一个随机数】

使用 Math.random()产生一个伪随机数 n,其中  $0 \leq n < 1$

【把数字取至最近的小数点位,即指定精确度】

1. 决定你要取的数字的小数点位数:例如,如果你想把 90.337 取成 90.34,就表示你要取到两位小数点位,也就是说你想取至最近的 0.01;

2. 让输入值除以步骤 1 所选的数字(此例为 0.01);

3. 使用 Math.round()把步骤 2 所计得的值取成最近的整数;

4. 把步骤 3 所得的结果乘以步骤 2 用于除法的那个值.

例如,要把 90.337 取成两个小数点位数,可以使用:

```
trace(Math.round(90.337/0.01)*0.01); //输出:90.34
```

【把数字取成一个整数的最接近倍数值】

例 1,这样会把 92.5 取成 5 的最近倍数值:

```
trace(Math.round(92.5/5)*5); //输出:95
```

例 2,这样会把 92.5 取成 10 的最近倍数值:

```
trace(Math.round(92.5/10)*10); //输出:90
```

【在指定数值范围内获得随机数】

```
//可取范围:[min, max]
```

```
private function randRange(min:Number, max:Number):Number {  
  
var randomNum:Number = Math.floor(Math.random() * (max - min + 1)) +  
min;  
  
return randomNum;  
  
}
```

【弧度(radian)与度数(degree)之间的转换】

从弧度转为度数:  $\text{degrees} = \text{radians} * 180 / \text{Math.PI}$

从度数转为弧度:  $\text{radians} = \text{degrees} * \text{Math.PI} / 180$

【计算两点之间的距离】

勾股定理:  $c^2 = a^2 + b^2$

假设有两个影片剪辑 mc1 和 mc2,则它们两点间的距离 c 为:

```
var c:Number = Math.sqrt(Math.pow(mc1.x - mc2.x, 2) + Math.pow(mc1.y -  
mc2.y, 2));
```

【模拟圆周运动】

已知圆心 o(x0, y0),半径 r 和弧度 angle,求圆上任意一点 P(x, y)的坐标:

```
x = x0 + (Math.cos(angle) * r);
```

```
y = y0 + (Math.sin(angle) * r);
```

注意:舞台的  $x$  轴正方向水平向右, $y$  轴正方向垂直向下.

#### 【模拟椭圆运动】

已知圆心  $o(x_0, y_0)$ ,长轴  $a$ ,短轴  $b$  以及弧度  $angle$ ,求圆上任意一点  $P(x, y)$ 的坐标:

$$x = x_0 + (\text{Math.cos}(angle) * a);$$
$$y = y_0 + (\text{Math.sin}(angle) * b);$$

#### 【华氏温度和摄氏温度之间的转换】

$$\text{华氏温度} = \text{摄氏度数} * 9 / 5 + 32$$
$$\text{摄氏温度} = (\text{华氏度数} - 32) * 5 / 9$$

#### 【公斤与磅之间的转换】

$$\text{公斤} = \text{磅} * 2.2$$
$$\text{磅} = \text{公斤} / 2.2$$

#### 【对用户输入的文字进行样式化】

使用 `defaultTextFormat` 属性,样式会施加至用户键入输入框的文字身上:

```
var formatter:TextFormat = new TextFormat();
```

```
formatter.color = 0x0000FF; //把文字变成蓝色
```

```
field.defaultTextFormat = formatter;
```

#### 【对现有文字的一部分进行样式化】

```
TextFormat.setTextFormat(format:TextFormat,    beginIndex:int    =    -1,  
endIndex:int = -1):void
```

#### 【设定文字框的字体】

例子:

```
HTML: field.htmlText = "<font face='Arial'>Formatted text";
```

```
TextFormat: formatter.font = "Arial";
```

CSS: P{ font-family: Arial; }

也可以使用以逗号隔开的字体清单: `formatter.font = "Arial, Verdana, Helvetica";`

注意:

字体和字体群组不同.字体群组有三种: `_sans`, `_serif` 以及 `_typewriter`.

`_sans` 群组一般就是指 Arial 或 Helvetica 这种字体;

`_serif` 群组一般就是指 Times 或 Times New Roman 这种字体;

`_typewriter` 群组一般就是指 Courier 或 Courier New 这种字体.

### 【嵌入字体】

使用[Embed]后设标签.[Embed]后设标签应该出现在 ActionScript 文件中,处于类宣告之外.你可以内嵌 TrueType 字体或系统字体.内嵌 TrueType 字体时的语法:

```
[Embed(source="pathToTtfFile", fontName="FontName",  
mimeType="application/x-font-truetype")]
```

`pathToTtfFile`: ttf 文件的路径,TrueType 字体的路径可以是相对的,也可以是绝对的;

`FontName`: 字体名称;

内嵌系统字体的语法:

```
[Embed(systemFont="Times New Roman", fontName="Times New Roman",  
mimeType="application/x-font-truetype")]
```

`fontName`: 以相同名称作为实际系统字体名称.

注意: 在使用嵌入字体时,要把 TextField 的 `embedFonts` 属性设为 `true`,这样 TextField 只能用内嵌字体了.如果你试着替 `embedFonts` 设为 `true` 的 TextField 使用设备字体,什么都不会显示.如果 `embedFonts` 设为 `true`,就无法指定以逗号相隔的字体清单.

### 【建立可以旋转的文字】

使用内嵌字体.当你旋转文字框时,设备字体就会消失.

### 【显示 Unicode 文字】

1. 从外部来源载入 Unicode 文字;
2. 假如你的编辑器支持 Unicode(如 Flex Builder),则可以直接在 ActionScript 程序中使用该字符;
3. 使用 Unicode 转义字符,ActionScript 里所有 Unicode 转义字符都以 u 开头,后面再跟四位十六进制数字.

注意: 如果你想取得 Unicode 字符,在 Windows 下使用: 开始>所有程序>附件>系统工具>字符映射表.

#### 【把 Flash Player 的焦点带给文字框】

```
stage.focus = field;
```

把焦点移除:

```
stage.focus = null;
```

注意: 当.swf 文件首次载入至网页浏览器时,并没有焦点.因此,以程序的方式把焦点指定给 Flash 应用程序的一个元素前,必须先把焦点移至 Flash Player.

#### 【以 ActionScript 选取文字】

使用 `TextField.setSelection(beginIndex:int, endIndex:int):void`

为了可以正常选取文字,文字框必须具有焦点:

```
stage.focus = field; //把焦点设给文字框
```

```
field.text = "This is example text"; //设定文字
```

```
field.setSelection(0, 4); //把"This"这个单词以高亮显示
```

使用只读的 `selectionBeginIndex` 和 `selectionEndIndex` 属性访问所选取的字符范围的索引.