# Flex学习笔记：

# 一、ActionScript

**1、命名空间的添加：**

添加初始化函数：

initialize="windowedapplication1\_initializeHandler(event)"

事件响应方法：

creationComplete="windowedapplication1\_creationCompleteHandler(event)"

**2、基本类型定义：**

**var** abc:int = 100;

**var** hello : String = **'hello world'**;

**var** b : Boolean = **true**;

**trace**(b);

**var** i : Number = 0.1;*//精度浮点型*

**trace**( i + i + i );*//浮点型相加可能得到的不是自己想要的*

**var** max : Number = 10000000000000;

**var** min : Number = 0.00000000000001;

**trace**(max + min);*//两个相差过大的浮点型不能相加减，否则会忽略掉小的*

**var** string : String;*//null*

**var** str1 : String = **""**;*//''*

**var** str2 : String = **new** String();

**3、引用类型的定义：**

**引用类型与Java一样，传递的是指针。**

**（1）数组**

**var** arr : Array;

arr = **new** Array();*//不建议使用*

arr = [];

数组与js类似

**（2）Object**

与js类似

不可直接trace(Object)，可用for..each遍历

**4、常量的声明**

**（1）基本类型常量**

**const** c : int = 10;*//必须定义的时候赋值，如果不赋值则为null，以后也不能赋值*

**（2）引用类型常量**

**const** o : Object = {};*//引用类型变量只能保证地址不变，不能保证地址存放内容不变*

o.age = 10;

**5、运算符、表达式**

**一、运算符**

**（1）delete 用于删除属性**

**var** person : Object ={};

person.age = 20;

person.name = **'xiaoli'**;

**trace**(person.age);*//控制台输出20*

**trace**(**'delete时'**+**delete** person.age); *//控制台输出true,有删除成功true*

**trace**(person.age);*//控制台输出undefined*

**（2）in 验证运算符，判断是否存在**

**var** person : Object ={};

person.age = 20;

person.name = **'xiaoli'**;

**trace**(**'age' in** person);*//控制台输出true*

**delete** person.age;

**trace**(**'age' in** person);*//控制台输出false*

**（3）typeof** **跟js一样**

**（4）is用来判断是否是某种类型**

**trace** (9 **is** Number); *// 控制台输出true*

**trace** (9 **is** Object); *// 控制台输出true*

**trace** (9 **is** String); *//控制台输出false*

**var** o : int;

**trace**(o **is** Number); *// 控制台输出true*

**（5）as用于多态的转变，成功返回原值，失败返回null**

**var** i : int = 10;

**var** o : Object = i **as** Object;

**var** n : Number = o **as** Number;

**trace**(i);*//控制台输出10*

**trace**(o);*//控制台输出10*

**trace**(n);*//控制台输出10*

**var** a : Array = i **as** Array;

**trace**(a);*//控制台输出null*

**二、表达式**

**（1）switch case**

Switch后面的可以是对象

**（2）for…in遍历集合里key-value的key**

**var** person : Object = {};

person.age = 20;

person.gemder = **'男'**;

person.name = **'张三'**;

**for**(**var** key : Object **in** person){

**trace**(key);

}

*//控制台输出：*

*//Age*

*//Gemder*

*//name*

**（3）for each… in遍历集合里key-value的value**

**var** person : Object = {};

person.age = 20;

person.gemder = **'男'**;

person.name = **'张三'**;

**for each**(**var** item : Object **in** person){

**trace**(item);

}

*//控制台输出：*

*//20*

*//'男'*

*//'张三'*

**（4）break，continue，标签**

loop0:

**for**(**var** i: int = 0;i < 10;i++ ){

loop1:

**for**(**var** j: int = 0;j<10;j++){

loop2:

**for**(**var** k:int = 0;k<10;k++){

**trace**(**'i='**,i,**'j='**,j,**'k='**,k);

**if**(k == 3){

**continue** loop1;

}**else** **if**(j == 2){

**break** loop0;

}

}

}

}

可以设定一个标签，用**break**和**continue**来对循环进行相应的操作

**6、函数**

**函数定义有两种方式：**

**参数的特性：**

**①可以在定义函数的时候给参数赋值，这是默认值，如果调用函数没有给函数参数的时候，就使用默认值。注意：带默认值的参数只能放后面，否则会提示错误。**

**②不定参数，列表里写（...param）不用指定类型，函数体里用for each**...**in遍历参数值**

**（1）普通函数：**

**function 函数名**(**参数名** : **参数类型**):**返回类型**{**函数体**}

**（2）匿名函数：**

**var** fun : Function;

fun **= function** (**参数名** : **参数类型**):**返回类型**{**函数体**}

**函数里可以用arguments对象arguments.length得到调用的时候输入参数个数，默认的不算**

**可以用arguments[下标]得到对应参数的值。**

**7、封装**

**①类分两种：加dynamic关键字的是动态类，可以动态增加或删除属性；不加为密封类，既不可改的普通类。**

**②用get和set关键字设置方法只读、只写**

**（1）set**

**public** **function** **set** gender(value : String):**void**{

\_gender = value;

}

**（2）get**

**public** **function** **get** gender():**void**{

**return** \_gender;

}

**8、继承**

**①as3不支持重载**

**②as重写 在权限修饰符前加一个override，方法名、参数、返回值都必须一样**

**③接口**

**（1）使用interface关键字定义**

**（2）定义在跟类名同名的.as文件中**

**（3）定义的方法不需要方法体**

**（4）只能定义方法，不能定义属性**

**（5）多继承 :implements接口1，接口2，...**

**9、多态**

**①向下转型：子类 = 父类 as 子类**

**②向上转型：父类 = 子类**

**10、类的管理，权限修饰符**

**①类成员控制：**

**1）internal包内访问，默认的**

**2）public 公共访问**

**3）private 当前类可以访问**

**4）protect继承关系可以访问，通常用来被子类重写**

**②类访问控制**

**1）internal包内访问**

**2）public 公共访问**

**3）final不能被继承的类**

# 二、flex开发

**1、MXML 组件定位**

**界面描述 样式表**

**渲染器**

**①功能包括： 数据描述 皮肤**

**数据绑定**

**事件绑定**

**可视标签：所有的类都继承自UIComponent**

**非可视标签： 所有的非UIComponent类**

**包括自己定义的不是继承自UIComponent的类**

**SCRIPT：脚本**

**STYLE：**s|Label{fontSize:20px;fontWeight:bold;color:#ff0000}**②mxml** s|XXX代表标签命名空间歧义使声明哪个空间

**分为三个部分 核心标签： int**

**Number**

**核心支持的数据类型 XML**

**Object**

**等等**