1. **前言**

1. 创建通俗易懂的编码规范，因为在软件开发的生命周期中，大部分时间都是在维护。这样，易于理解的代码片段变得很重要，因为不总是最初的开发者去维护代码。通俗地说就是，让开发者能快速理解别人的代码；

2.为了避免产生一些奇怪的bug和防止内存泄漏或者提高性能必须要做一些约束；

1. **编码环境要统一**  
   1.统一使用相同的IDE，建议都使用FlashBulder

2.使用同样版本的引擎库

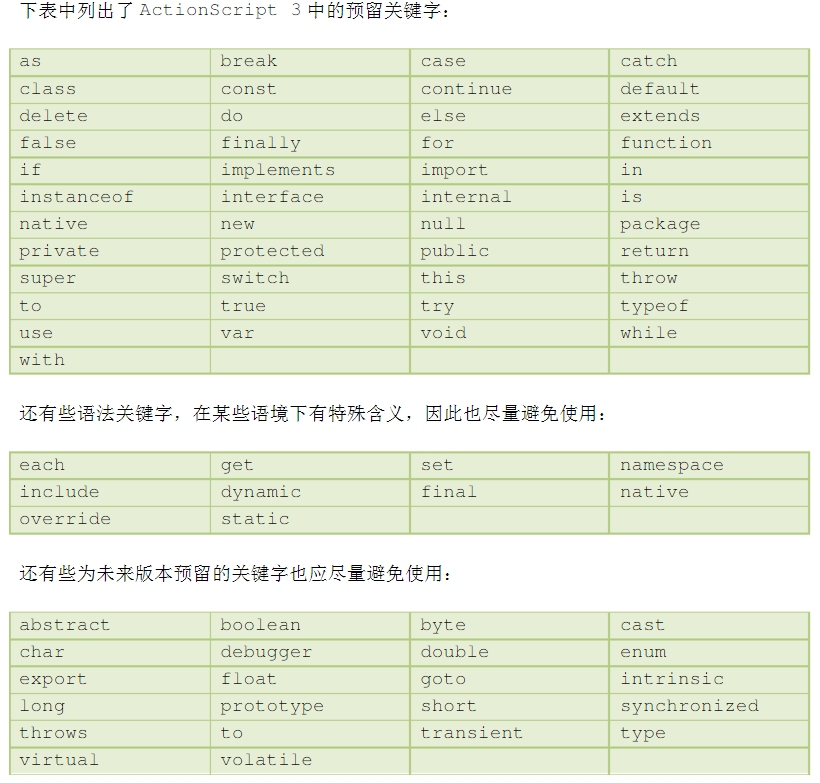
1. **命名规范**
2. 包的命名必须使用lowerCamelCase（首字母小写驼峰式命名），第一个单词的开头字母小写，其他单词的开头字母大写
3. 类名推荐使用名词，也可以使用形容词。总是使用UpperCamelCase（开头字母大写的驼峰式命名）形式
4. 接口的命名规则和类的是一样的，只是开头加上一个大写字母I

4. 方法名必须使用动词开头，并使用lowerCamelCase（开头字母小写的驼峰式命名）形式，如果方法被一个事件调用，结尾加上Handler

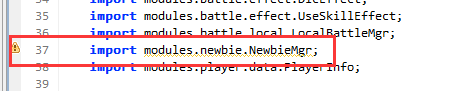
1. 变量名使用lowerCamelCase（首字母小写驼峰式命名）形式命名，并且使用有意义的名称，如果变量使用getter和setter方法读/写值，开头应加上下划线（\_）
2. 私有变量名应该以下划线（\_）,函数中局部变量名不必在前面加下划线（\_）
3. Boolean型变量应以can、is和has开头，不应该使用否定式名称
4. 常量全部使用大写字母，单词间用下划线分割（\_）
5. Ui组件变量命名应该加上UI组件缩写
6. 虽然我们使用as3编码，但是最终还是会编译成JavaScript，常用的方法名、变量名，尽量控制它的长度，禁止过长，可以减少js文件大小
7. 预留关键字禁止使用







1. 所有文本文件的字符编码必须为UTF-8，特别是手动新增的js、html文件等
2. 每一行只有一个声明，只执行一条语句，后面要加分号( ; )
3. 导入的类没有使用到，一定要移除，FlashBulder组织导入快捷键（Shift+Ctrl+O）



1. 别人最先写的类，定义好了这个格式，你就要遵循这个格式，否则你可以全部改成你的格式

如UIFactory类的页面类型对应的资源列表，都是一行对应一个资源列表的，又掺杂多行的格式



1. **内存泄漏注意事项**

**一.所有用EventMgr监听的事件，都要在destroy方法添加移除事件**

如：

EventMgr.getInstance().on("XXX",this,this.onHandler);

override public function destroy(destroyChild:Boolean = true):void {

super.destroy();//这行代码一定要加，否则基类释放不干净

EventMgr.getInstance().off("XXX",this,this.onHandler);

}

**二.所有用到非对象本身（或者不是对象的子组件）监听的事件，都要移除事件**

如在某个页面内，使用stage监听事件：

Laya.stage.on(Event.MOUSE\_DOWN,this,this.onXXX);

**三.所有用到Handler的地方都要注意了**

所有地方都统一用Handler.create()创建，替换new Handler()的方式创建

handler直接赋值为null，是不会回收的（如 handler=null;）

Handler释放一定要调用recover()方法，或者clear()方法，建议使用recover()，清理并回收到对象池

Handler泄漏的方式如下：

1.重复创建覆盖，只需在init执行一次监听即可，禁止多次覆盖创建替换

btnClose.clickHandler = Handler.create(this,onClose,null,false);

btnClose.clickHandler = Handler.create(this,onClose,null,false);

2.Handler.create(caller:\*, method:Function, args:Array=null, once:Boolean=true)

当once=false，即不是执行一次就释放的，都要自己手动释放

如：

Handler.create(this, this.onXXX,null,false);

Laya.loader.load(res, Handler.create(this, onResLoaded,null,false));

3.对于列表里面的render监听  
itemList.renderHandler = Handler.create(this,this.onRender01,null,false);  
一定要在初始化完成，不可重复监听

4.销毁组件一定要考虑组件自身销毁的情况，要重写组件的destroy的方法进行处理

**三.定时器**

Laya的底层做了处理，在当组件释放时，该组件的监听的定时器也会释放掉，不用手动释放也可以，但也要注意一些事项

**1.统一使用Laya.timer监听，不要使用Laya.stage.timer监听**

**2.使用非对象本身监听的定时器，一定要手动释放**

1. **性能优化**

**1.** 禁止使用forEach的方式，建议使用第二种方式

