

使用Wax时要注意的地方（<https://github.com/probablycorey/wax/wiki/Overview>）。

虽然Wax能让OC和Lua协调工作，但Wax也有一些功能上的限制，并有相应的解决方案。下

面将介绍一些要注意的地方：

1、实例化一个OC对象时，无需调用alloc，因为Wax已经帮我们处理了。同时也没必要调

用retain或者release，据说Lua已经实现了类似Java的垃圾回收机制，关于这点，有待

文献查阅或亲测验证。

如：`local view = UIView initWithFrame(CGRect(0, 0, 100, 100))`

2、要在Lua中创建OC类时，只要调用函数`waxClass{CLASS_NAME, PARENT_CLASS}`即可。

创建时，原则上要求一个文件最多只能创建一个类，因为Wax会把文件中的所有函数当做

该类的方法。

如：`waxClass{"MyClass", NSObject}`

等价于`waxClass({"MyClass", NSObject})`

当然，要想添加协议(Protocols)的话，如下即可：

```
waxClass{"MyClass", NSObject, protocols = {"UITableViewDelegate", "UI
UITableViewDataSource"}}
```

说明：第一个参数表示要创建的类名，第二个参数表示要继承的父类，第三个参数表

示要实现的协议列表。

3、在Lua中调用OC的方法是，不能用点、方括号等方法调用，而是用冒号。

如：`UIApplication:sharedApplication()`

说明：在Lua中的点能够动态创建成员变量（而在OC代码中完全不知道这些变量），

这些成员在对象的生命周期中是有效的。

4、在Lua中，没有t样的his或self这关键字，但时Wax为我们提出了一个解决方案，就是

方法中的首个参数（必须）为self，也就是说，在Lua中定义的类中的所有函数，第一个

参数必须为self，从第二个参数起才是我们所希望传递的参数。

如：`waxClass{"MyClass", NSObject}`

```
function storeWords(self, words)

    self.words = words

    self.super:words(self)

end
```

说明：如果想要调用父类的方法，Wax为我们准备了super的属性，如上例所示。

5、在OC方法中，往往需要传递多个参数，在Wax中会用下划线(\_)来替代OC中的冒号。

如：在OC中：`[UIAlertView initWithTitle:@"title" message:@"message" deleg`  
`ate:nil];`

在Wax中：`UIAlertView:initWithTitle_message_delegate("title", "messag`  
`e", nil)`

6、如果OC中的一个类中的变量声明了Properties，那样在Wax中是无法直接赋值或获取

值的，而要调用getters/settes的方法。

如：在OC中：`someView.frame`

在Wax中：`someView:frame()`  
`someView:setFrame(someFrame)`

7、如果我们不想OC对象被自动转换，则可以使用toobjc方法。

如：`local string = "a string"`  
`string:capitalizedString()`

这个时候，string是Luastring，所以调用capitalizedString()方法时会报错。  
。

如果如下这样做的话，则会将Luastring转换成NSString，那就可使用NSStrin  
g中的方法了：

`toobjc(string):capitalizedString()`

8、枚举类型是被硬编码在APP\_ROOT/wax/stdlib/enums.lua中的。也就是说，我们要想

新增一个枚举类型，就要在enums.lua中注册枚举，具体注册方法，见enums.lua。不过

，在这里新增的枚举要生效，需要使用compile-stdlib.sh重新编码。当然，也可以在自

定义的lua文件中声明，只要在合适的位置引入lua文件即可。

9、在OC中，会经常传递selector，在Wax中，把selector中的串当作字符串即可。

如：在OC中：`[self scheduleOnce:@selector(makeTransition:) delay:1];`  
在Wax中：`self:scheduleOnce_delay("makeTransition", 1)`

10、OC中的结构体在Wax中是不被支持的。结构体在Wax中同样是硬编码的，在APP\_ROOT  
/wax/wax-scripts/structs.lua中。看下这个文件，应该就能知道结构的声明该怎么做

。可参照第8点。11、最后补充一点，在Lua中的单行注释方法是在两条短横线("--")

，多行注释的方法是以"--[[“开头，以”]]”结尾。