1. **编码规范**
2. 变量

局部变量必须以小写开头。

|  |
| --- |
| **local** pkt = {}  pkt["msg"] = "createPlayer"  pkt["player"] = **net**.\_player  **self**.sendEvent(pkt) |

成员变量必须以\_+小写开头。

|  |
| --- |
| **self**.\_db = **None** -- 数据库操作类  **self**.\_logic = **player\_logic**.PlayerLogic.\_instance -- 玩家逻辑 |

1. 常量/枚举

常量/枚举必须全部使用大写，并且用\_分割单词。

|  |
| --- |
| --- 月份最大值  **MONTH\_MAXCOUNT** = 12  --- 能力最大值  **ABI\_MAXCOUNT** = 10  --- 等级最大值  **LEVEL\_MAXCOUNT** = 100  --- 男性  **SEX\_MALE** = 1  --- 女性  **SEX\_FEMALE** = 2 |

枚举用类型开头。

|  |
| --- |
| --- 男性  **SEX\_MALE** = 1  --- 女性  **SEX\_FEMALE** = 2 |

所有数字和字符串都应该定义成一个常量。

|  |
| --- |
| --- 月份最大值  **MONTH\_MAXCOUNT** = 12 |

1. 函数

函数名用小写开头。

|  |
| --- |
| --- 创建初始数据  -- **@param** net 网络连接  -- **@param** name 名字  -- **@param** model 模型  -- **@param** month 出生月份  -- **@return** 成功返回None，否则返回唯一字符错误码  **function** createData(self, net, name, model, month) |

1. 类

类名必须以大写开头。

|  |
| --- |
| --- 玩家逻辑  -- @class table  -- @name PlayerLogic  -- **@field** \_name 名字  -- **@field** \_sex 性别  **PlayerLogic** =  {  \_name = **nil**,  \_sex = **nil**,  } |

1. 缩写词

缩写词最多只使用首字母大写。

|  |
| --- |
| --- 创佳你html文档  **function** createHtml(self) |

1. 测试值

检测时尽量不要省略右值。应为python中0和None是不同的数据

|  |
| --- |
| -- id有可能为0  **if** **playerId** == **nil** **then**  **return** "PLAYERLOGIC\_REMOVE\_NTLOGIN"  **end** |

1. 留空

操作符，关键字，冒号分号之类的前后注意使用空格留空。

|  |
| --- |
| -- 检查是否可以叠加  **if** **t**.\_stackable == **false** **then**  **return** "ITEMLOGIC\_SPLIT\_UNSTACKABLE"  **end** |

逻辑及不同组的变量之间用换行分隔开。

|  |
| --- |
| -- 检查是否可以叠加  **if** **t**.\_stackable == **false** **then**  **return** "ITEMLOGIC\_SPLIT\_UNSTACKABLE"  **end**  -- 检查物品数  **if** **srcCount** <= 0 **or** **dstCount** <= 0 **then**  **return** "ITEMLOGIC\_SPLIT\_COUNTINVALID"  **end**  **if** srcCount + dstCount != srcItem[item\_define.ITEMDATA\_COUNT] **then**  **return** "ITEMLOGIC\_SPLIT\_COUNTINVALID"  **end** |

1. **注释风格**
2. 文件头

|  |
| --- |
| --- 游戏服务器  -- @author Dashi Bai  -- @date 2011-06-21  -- @note 额外说明  -- @bug 要修改的BUG  -- @todo 还需要做的事情 |

note，bug，todo标记是非必须的

1. 类

|  |
| --- |
| --- 玩家逻辑  -- @class table  -- @name GameServer  -- **@field** \_players 玩家表  -- **@field** \_monsters 怪物表  GameServer =  {  \_players = {}  \_monsters = {}  } |

field为public的成员变量,格式为-- @field 变量名 说明

note，bug，todo标记是非必须的

1. 枚举/常量

|  |
| --- |
| --- 月份最大值  **MONTH\_MAXCOUNT** = 12;  --- 能力最大值  **ABI\_MAXCOUNT** = 10;  --- 等级最大值  **LEVEL\_MAXCOUNT** = 100;  --- 男性  **SEX\_MALE** = 1;  --- 女性  **SEX\_FEMALE** = 2; |

1. 变量

|  |
| --- |
| --- 实例  **\_instance** = **nil**  -- 子控件  **\_children** = {} |

如果\_instance需要暴露到外面，则我们标记为3个减号；而\_children这个变量，我们不准备暴露，则使用2个减号

1. 函数

|  |
| --- |
| --- 创建玩家数据  -- **@param** player 玩家信息  -- **@param** sex 性别  -- **@param** month 出生月份  -- **@param** model 模型  -- **@return** 返回新创建的玩家  **function** createPlayerData(player, sex, month, model) |

param标记表示变量参数

return 标记表示返回的数据

note，bug，todo标记是非必须的

1. 代码片段

|  |
| --- |
| --- 播放滚动动画  -- @name play  -- **@param** ci 当前要播放的下标  -- **@param** ni 下一个要播放的下标  -- **@return** 返回当前播放的动画  **function** scroller:play(ci, ni)  **if** #tweens == 0 **then** #self.sort\_dict > 1 **end**  -- 取得要做的动画的两个控件  **local** node1 = self.sort\_dict[ci].mc\_node  **local** node2 = self.sort\_dict[ni].mc\_node    -- 只显示要做的动画  **for** k, v **in** ipairs(self.sort\_dict) **do**  v.mc\_node:setVisible(**true**)  **end**    -- 开始做动画  tweens[#tweens + 1] = mgr:TweenFromTo()  **end**  **return** tweens  **end** |

如果不是原子操作，则必须对代码进行详细的注释。务必写清楚各块的功能。注意，由于不需要暴露，所以用一个#号