

# DM接口文档2021.2.6

---

文本加密解密接口

写界面配置

商会相关

宝宝相关

队伍

技能 目标 环境遍历

背包仓库锦囊

NPC任务寻路

通用

玩家

# 文本加密解密接口

```
1 -- 编号: N0.1
2 -- 返回加密后的文本,分发
3 -- aType = 1 时,aText是明文,aMiMa是密码
4 -- aType = 2 时,aText是明文,aMiMa是指定要绑定的脚本用户名
5 -- aType = 3 时,aText是明文,aMiMa不使用,内部使用脚本登录的用户名+密码加密,
   修改密码后,下次登录后,密文无法解密
6 EncryptLuaText(aType, aText, aMiMa)
7 -- N0.1 demo
8
9
10
11 -- 编号: N0.2
12 -- 运行加密文本
13 -- aType = 1 时,aText是密文,aMiMa是密码
14 -- aType = 2 时,aText是密文,aMiMa不使用,内部使用脚本登录的用户名解密
15 -- aType = 3 时,aText是密文,aMiMa不使用,内部使用脚本登录的用户名+密码解密,
   修改密码后,下次登录后,密文无法解密
16 RunEncryptLuaText(aType, aText, aMiMa)
17 -- N0.2 demo
```

# 写界面配置

```
1 -- 编号: NO.1
2 -- 文本修改角色配置
3 -- 如何查看配置的字段->界面导出配置json,用编辑器格式化json后查找,一般是拼音缩写
4 -- 注意值的类型要和json一致,数字是字符串类型时要加双引号
5 -- 如果未成功写入查看日志 main.txt
6
7 local menpaiID = Player:GetMenPaiID() -- 获取门派ID
8 local myMenpai = GetMenpaiNameByID(menpaiID) -- 获取门派名称
9 local myName = Player:GetName() -- 获取自己的名字
10
11 local config = {}
12 -- 通用设置
13 config.tysz = {}
14 config.tysz.f11_hide = true -- 通用设置 -> F11屏蔽
15 config.tysz.f12_hide = true -- 通用设置 -> F12屏蔽
16
17 -- 名士赠礼
18 config.mszt = {}
19 config.mszt["金丝冠"] = {text = "苏州-赵钱孙-龟鹤延年"}
20
21 -- 技能设置
22 if myMenpai == "逍遥" then
23     -- 副本攻击技能
24     config.skill = {}
25     config.skill.fuben_gongji = {
26         {data = 0, text = "弹指神功"},
27         {data = 0, text = "溪山行旅"},
28         {data = 0, text = "鲤跃北溟"}
29     }
30 end
31
32 -- 古兰玉兑换宝石
33 config.glydhbs = {}
34 config.glydhbs.zy = "5" -- 兑换5个紫玉 -- 注意类型要和json一致,字符串类
```

型 5 要加双引号

```
35 config.glydhbs.zml = "5" -- 兑换5个祖母绿
36
37 -- 修改配置接口
38 WriteConfig(config)
39
```

# 商会相关

```
1 -- 编号: NO.1
2 -- 购买商会物品
3 -- 参数1: aBuyName    文本型 买物品名称
4 -- 参数2: aBuyNum    整数型 购买数量
5 -- 参数3: aMaxMoney   整数型 最大购买金额, 单位 (铜)
6 -- 参数4: aNameCmpMode 整数型 名称匹配模式, 可以为空
7 ShangHui_BuyItem(aBuyName, aBuyNum, aMaxMoney, aNameCmpMode)
8 -- No.1 demo
9 ShangHui_BuyItem("黄纸",40,10000) -- 买黄纸40个, 不超过10金
10
```

# 宝宝相关

- 编号: NO.1
- 方法名: Pet:GetSelfPet()
- 描述: 获取当前出战珍兽信息
- 返回环境中自己的珍兽信息 ,返回的数据结构与二叉一致, 没有找到返回nil

```
1 local tPet = Pet:GetSelfPet()  
2 PushColorMsg("id = "..tPet.id..",name = "..tPet.name..",level = ".  
   .tPet.level)
```

- 编号: NO.2
- 方法名: Pet:GetTID()
- 返回值: 珍兽的环境ID, 没有找到返回-1

```
1 local nTid = Pet:GetTID()
```

- 编号: NO.3
- 方法名: Pet:GetLevel(aGUID)
- 描述: 通过GUID获取宝宝等级, guid通过9号方法

```
1 Pet:GetLevel(aGUID)
```

- 编号: NO.4
- 方法名: Pet:Feed(idx)
- 描述: 喂养,
- 参数1: 索引

```
1 Pet:Feed(0)
```

-- 编号: NO.5  
-- 方法名: Pet:Relax(aSleep)  
-- 描述: 收回珍兽  
-- 参数1: 是否延时可空  
-- 返回值: true / false

```
1 Pet:Relax(100)
2 Pet:Relax()
```

-- 编号: NO.6  
-- 方法名: Pet:CheckRelax(aID, aType)  
-- 描述: 检查收回珍兽  
-- 参数1: 若参数2为1, 则参数1和自己的宝宝tid相同收回  
-- 参数1: 若参数2为2, 则参数1和自己的宝宝攻击的目标ID相同收回, 比如慕容复反伤害时, 若宝宝攻击慕容复收回

```
1 Pet:CheckRelax(aID, aType)
```

-- 编号: NO.7  
-- 方法名: Pet:GetMaxCount()  
-- 描述: 珍兽携带上限

```
1 local nMaxCount = Pet:GetMaxCount()
2 PushColorMsg(nMaxCount)
```

-- 编号: NO.8  
-- 方法名: Pet:GetCurrentCount()  
-- 描述: 获取当前携带的珍兽总数  
-- 返回值: 整数型

```
1 local nCurCount = Pet:GetCurrentCount()
2 PushColorMsg(nCurCount)
```

-- 编号: NO.9  
-- 方法名: Pet:GetAllInfo()  
-- 描述: 获取所有珍兽  
-- 返回值: 珍兽列表数组

```
1 local nPets = Pet:GetAllInfo()
2 for i=1,#nPets do
3     local tmp = nPets[i]
4     PushColorMsg("index = "..tmp.index..",name = "..tmp.name..",le
    vel = "..tmp.level..",guid = "..tmp.guid)
5 end
```

-- 编号: NO.10  
-- 方法名: Pet:AddtAttribute()  
-- 描述: 珍兽加点  
-- 返回值:

```
1 Pet:AddtAttribute()
```



# 队伍

-- 编号：NO.1  
-- 方法名： Team:CreateTeam()  
-- 描述：创建队伍  
-- 返回值：无返回值  
-- 示例：直接使用创建队伍

```
1 Team:CreateTeam()
```

-- 编号：NO.2  
-- 方法名： Team:LeaveTeam()  
-- 描述：离开队伍  
-- 返回值：无返回值  
-- 示例：退出队伍

```
1 Team:LeaveTeam()
```

-- 编号：NO.3  
-- 方法名： Team:GetUpdateMemberInfo()  
-- 描述：获取队伍所有数据  
-- 返回值：队伍数组  
--[[  
 id // ID  
 name // 名称  
 level //等级  
 x //x坐标  
 y //y坐标  
 hp //血量  
 dst //距离  
 leader //队长  
 sex //性别

```

menpai    //门派
sceneID   //场景ID
guid      //GUID
teamIndex //队伍索引
groupIndex //团队索引
hpcur     // 当前血量
hpmax     //最大血量
-- ]]
-- 示例,获取队伍人员信息并输出调试信息

```

```

1 local tAllMember = Team:GetUpdateMemberInfo()
2 for i=1,#tAllMember do
3     local temp = tAllMember[i]
4     PushColorMsg(fmt("id=%s name=%s level=%s x=%s y=%s hp=%s dst=%s leader=%s sex=%s menpai= %s sceneID=%s guid=%s teamIndex=%s groupIndex=%s hpcur=%s hpmax=%s",temp.id,temp.name,temp.level,temp.x,temp.y,temp.hp,temp.dst,temp.leader,temp.sex,temp.menpai,temp.sceneID,temp.guid,temp.teamIndex,temp.groupIndex,temp.hpcur,temp.hpmax)
5     Sleep(1000)
6 end

```

```

-- 编号：NO.4
-- 方法名：Team:GetLeaderInfo()
-- 描述：获取队长信息
-- 返回值1：有队长返回队长数据表，有上面的字段，没队长返回 nil
-- 返回值2：自己是队长返回 true
-- 示例：

```

```

1 local ret = Team:GetLeaderInfo()

```

```

-- 编号：NO.5
-- 方法名：Team:GetMemberCount()
-- 描述： 队伍人数
-- 返回值：整数型

```

-- 示例

```
1 local ret = Team:GetMemberCount()
```

-- 编号：NO.6

-- 方法名：Team:KickTeamMember(aIndex)

-- 描述：踢人 队伍索引（0-5）

-- 参数：队伍索引

-- 返回值：无

-- 示例：

```
1 Team:KickTeamMember(0)
```

-- 编号：NO.7

-- 方法名：TeamChangeLeader(5)

-- 描述：更换队长

-- 参数1：整数型 门派id

-- 参数2：文本型 名字（可空）

-- 返回值：无

-- 示例：

```
1 TeamChangeLeader(5)
```

# 技能 目标 环境遍历

```
1  --[[
2      功能描述:          EnumEnvObj 遍历所有环境对象
3          返回值数量:      1
4          返回值类型:      数组
5          返回数组字段:
6              id, //环境id
7              name ,    //名称
8              class ,    //类型
9              title,    //称号
10             x ,        //坐标x
11             y ,        //坐标y
12             dst ,      //距离
13             hp,        //血百分比
14             mpp,       //蓝百分比
15             sex ,      //性别
16             guid ,     //固定的ID
17             owner ,    //怪归属guid
18             menpai ,   //门派
19             petzrid ,  //珍兽主人id
20             state ,    //状态(2走路,3轻功,5运气,9死亡,10摆摊)
21             zwType,    //种地作物成熟类型
22             level ,    //等级
23             target ,   //目标id
24             model ,    //模型id
25             bhid,      //帮会id
26             tmid,      //同盟id
27             zmid,      //战盟id
28             inteam ,   //是否有队伍
29             leader,    //是否队长
30             follow,    //
31             attackType,//可攻击类型(0可攻击)
32             baseptr    //对象地址
33      --字段说明:      环境id,名称,类型,称号,坐标x,坐标y,距离,血百分比,蓝百分比,
```

性别, 固定的ID, 怪归属guid, 门派, 珍兽主人id, 状态(2走路, 3轻功, 5运气, 9死亡, 10摆摊), 种地作物成熟类型

34 等级,目标id,模型id,帮会id,同盟id,战盟id,是否有队伍,  
是否队长,可攻击类型(0可攻击),对象地址

35      参数:      无

36 11

37

## 38 -- 示例

```
39 local tEnv = EnumEnvObj()
```

```
40 for i = 1, #tEnv do
```

```
41     local temp = tEnv[i]
```

```
42     if temp.class == "玩家" and temp.attackType == 0 then
```

```
43      if Target:FindBuff(temp, "虚弱：所受伤害放大") then
```

```
44 PushColorMsg(temp.name .. "被放大了,集火秒")
```

45 end

46           end

```
47 PushColorMsg(fmt("名称: %s 类型: %s", temp.name, temp.class))
```

48 end

49

50 --示例：通过TID寻找附近唯一的目标 您可以直接使用 FindObjByTID()

```
51 function Demo_FindObjByTID(aTid)
```

```
52     if type(aTid) ~= "number" or aTid < 0 then
```

```
53         return nil
```

54 end

```
55     local tobj = EnumEnvObj()
```

```
56     for i = 1, #t0bj do
```

```
57         if tObj[i].id == aTid then
```

```
58         return t0bj[i]
```

59 end

```
60      end
```

```
61     return nil
```

62 end

63

64 --示例：通过GUID寻找附近唯一的目標 您可以直接使用 FindObjByGUID()

```
65 function Demo_FindObjByGUID(aGUID)
```

```
66 if type(aGUID) ~= "number" then
```

```
67         return nil
```

68 end

```
69     local tObj = EnumEnvObj()
```

```
70     for i = 1, #tObj do
```

```

71         if tObj[i].guid == aGUID then
72             return tObj[i]
73         end
74     end
75     return nil
76 end
77
78
79 -- 示例： 按名称过滤环境，您可以使用内部封装的Enum2XByName()
80 -- 参数1：文本型 要枚举的名称，多个名称空格分开同时设置 aPiPeiType匹配模式
81 -- 参数2：表结构（数组） 类型过滤表，1保留，0排除，第1个.npc 第2个.怪物 3.玩家
82 -- 4.珍兽 5.宝箱
83 -- 如：{1,1,1,1,1},{1,1,0,0,0}
84 -- 参数3：表结构（数组） 过滤血 距离 {0,-1}
85 -- 参数4：整数型 默认0.不排序 1.先近后远 2.先远后进，3先血少后血多，4先血多
86 -- 后血少
87 -- 参数5：整数型 名称匹配模式 0完全相同，默认3是前后带空格的， 1 2 4
88 -- 返回值：表结构（数组） 如：{"id" = "xxx",["name"] = "xxx",.....}
89 function Demo_Enum2XByName(aName, aClassType, aFilter, aSort, aPiPeiType)
90     if type(aName) ~= "string" or aName == "" then
91         return {}
92     end
93     aPiPeiType = aPiPeiType or 0
94
95     local tObj = EnumEnvObj()
96     local tObj2 = ObjFilterByStr(tObj, aName, aPiPeiType, "name"
97 ) -- 名字过滤
98     tObj2 = ObjFilterByClass(tObj2, aClassType) -- 类型过滤
99
100     if aFilter ~= nil then
101         -- 小于0不判断 1就是>hp 过滤掉死亡的
102         tObj2 = ObjFilterByInt(tObj2, aFilter[1], 1, "hp")
103         --小于0不判断. 2就是<=dst -- 不兼容0
104         local cmp_dist = aFilter[2]
105         tObj2 = ObjFilterByInt(tObj2, cmp_dist, 2, "dst")
106     end
107
108     ObjSort(tObj2, aSort) -- 排序
109     return tObj2

```

```

107 end
108
109 -- 请求DEMO
110 local tObj = Demo_Enum2XByName("◆__晨耀αべ",{0,0,1,0,0},{1,15},0,0
    )
111 for i=1,#tObj do
112     local tmp = tObj[i]
113     PushColorMsg("id = "..tmp.id..",name = "..tmp.name..",guid = ".
        .tmp.guid)
114     Sleep(1000)
115 end

```

-- 编号: NO.1

-- 方法名: Skill:UseQingGong(aPosX, aPosY)

-- 描述: 使用轻功

-- 参数1: aPosX, 整数型。x坐标

-- 参数2: aPosY, 整数型, y坐标

-- 返回值: nil

-- 示例:

```

1 Skill:UseQingGong(200, 300)

```

-- 编号: NO.2

-- 方法名: Skill:UseSpeed()

-- 描述: 使用加速

-- 返回值: true:使用成功 / false: 使用失败

-- 示例:

```

1 Skill:UseSpeed()

```

-- 编号: NO.3

-- 方法名: Skill:UseShunYi(aPosX, aPosY)  
-- 描述: 使用瞬移  
-- 参数1: aPosX, 整数型, x坐标  
-- 参数2: aPosY, 整数型, y坐标  
-- 返回值: true:使用成功 / false: 使用失败  
-- 示例:

```
1 Skill:UseShunYi(200, 300)
```

-- 编号: NO.4  
-- 方法名: Skill:UsePetSkill(aSkillName, aPosX, aPosY, aSleep)  
-- 描述: 使用珍兽技能 冰天雪地 烈火燎原  
-- 参数1: aSkillName, 文本型, 技能名称  
-- 参数2: aPosX, 整数型, x坐标, 可以为空  
-- 参数3: aPosY, 整数型, y坐标, 可以为空  
-- 参数4: aSleep, 整数型, 延时时间, 可以为空  
-- 返回值: true:使用成功 / false: 使用失败  
-- 示例:

```
1 local ret1 = Skill:UsePetSkill("烈火燎原")  
2 local ret2 = Skill:UsePetSkill("高级神佑")
```

-- 编号: NO.5  
-- 方法名: Skill:UseSkillByName(技能名称)  
-- 描述: 按技能名使用技能,您也可以使用内部封装的Skill:UseSkillByName()  
-- 参数: 技能名称  
-- 返回值: true:使用成功 / false: 使用失败  
-- 示例: 内部封装

```
local ret1 = Skill:UseSkillByName("鹰击长空")
```

-- 编号: NO.6  
-- 方法名: Skill:GetSkillByName(aSkillName)  
-- 描述: 通过技能名称获取技能ID  
-- 参数: 技能名称



```
-- 返回值：技能数组，结构如下
-- {
    cdidx,          // 用于计算冷却
    id,             // 技能ID。如果与skill.baseID相等，则是门派平推技能
    dist,           // 技能有效距离
    consume,        // 技能消耗的具体数值。
    consume_type,   // 技能消耗类型，气，怒，共情值等
    baseskl,        // 基础技能/包括像平推的技能,必须是skill.baseID == 当前ID。则是平推技能
    attack,         // 是否为攻击技能
    diandi,         // 是否需要点地
    qun,            // 群攻技能
    jiasu,          // 加速技能
    shunyi,        // 瞬移技能
    name,           // 技能名称
    buff_check,     // 指定buff就不用此技能
    buff_need,      // 技能是否依赖buff
}

-- 示例1：通过门派技能名称获取技能ID
```

```
1 local ret1 = Skill:UseSkillByName("鹰击长空")
2 PushColorMsg(ret1.id) -- 如上数组结构，获得技能ID
3 -- 示例2：通过技能函数进行内部封装的DEMO
4 function Skill_UseSkillByName(aSkillName, aTargetID, aX, aY, aSleep)
5     aTargetID = aTargetID or -1
6     aX = aX or -1
7     aY = aY or -1
8     local skill = Skill:GetSkillByName(aSkillName)
9     return Skill:CheckUse(skill, aTargetID, aX, aY, aSleep)
10 end
```

```
-- 编号：NO.7
-- 方法名：Skill:UseSkillByID(aSkillID)
-- 描述：按技能ID使用技能,您也可以使用内部封装的Skill:UseSkillByID()
-- 参数1：aSkillID 整数型 门派技能ID，可以使用【编号：NO.6】获取技能ID
-- 返回值：true:使用成功 / false：使用失败
-- 示例：
```

```

1 function Skill_UseSkillByID(aSkillID, aTargetID, aX, aY, aSleep)
2     aTargetID = aTargetID or -1
3     aX = aX or -1
4     aY = aY or -1
5     local skill = Skill:GetSkillByID(aSkillID)
6     return Skill:CheckUse(skill, aTargetID, aX, aY, aSleep)
7 end

```

-- 编号: NO.8  
 -- 方法名: Skill:GetSkillByID(aSkillID)  
 -- 描述: 技能是否可以使用, 检测冷却蓝值状态  
 -- 参数: aSkillID 整数型 门派技能ID  
 -- 返回值: 技能数组, 结构如上 【编号: NO.6】  
 -- 示例:

-- 编号: NO.9  
 -- 方法名: Skill:CanUseSkill(aSkill)  
 -- 描述: 技能是否可以使用, 检测冷却蓝值状态  
 -- 参数: 一个技能表, Skill:GetSkillByName(aSkillName) 或 Skill:GetSkillByID(aSkillID)返回值  
 -- 返回值: true:表示技能可以使用 / false:表示技能在冷却中, 不能使用  
 -- 示例:

```

1 local ret1 = Skill:CanUseSkill(Skill:GetSkillByName("鹰击长空"))
2 PushColorMsg(ret1)

```

-- 编号: NO.10  
 -- 方法名: Skill:IsMiJi(aSkill)  
 -- 描述: 是否秘籍技能  
 -- 参数: 一个技能表, Skill:GetSkillByName(aSkillName) 或 Skill:GetSkillByID(aSkillID)返回值  
 -- 返回值: 0:是 / -1:否  
 -- 示例:

```
1 local ret1 = Skill:IsMiJi(Skill:GetSkillByName("御气诀·气宗"))
2 PushColorMsg(ret1)
```

-- 编号: NO.11  
-- 方法名: Skill:GetJinGongFarSkill()  
-- 描述: 获取进攻的远程技能  
-- 返回值: 返回技能数组  
-- 示例:

```
1 local ret1 = Skill:GetJinGongFarSkill()
2 for i = 1, #ret1 do
3     local tmp = ret1[i];
4     PushColorMsg(tmp.name)
5     Sleep(1000)
6 end
```

-- 编号: NO.12  
-- 方法名: Skill:RefreshSkill()  
-- 描述: 重新刷新所有技能数据, 存储在Skill.skill里, 一般不需要刷新, 像有时活动变身了, 可能增加了一些技能  
-- 返回值: nil  
-- 示例:

```
1 Skill:RefreshSkill()
```

-- 编号: NO.13  
-- 描述: Skill.baseID 整数型的值, 门派的平推技能ID  
-- 示例: 返回基础技能的ID值

```
1 local ret1 = Skill.baseID
2 PushColorMsg(ret1)
```

-- 编号: NO.14  
-- 方法名: Target:Attack(aTid, aSleep, aSelect)  
-- 描述: 平推目标  
-- 参数1: 攻击的目标ID, 通过文章开关的demo获取对应的目标ID  
-- 参数2: 延时 整数型, 毫秒值  
-- 参数3: 是否选中目标 1:选中, 0:未选中  
-- 返回值: true:已经击中了目标ID / false:没有击中目标ID  
-- 示例:

```
1 local tEnv = EnumEnvObj()  
2 for i = 1, #tEnv do  
3     local temp = tEnv[i]  
4     if temp.name == "天山小雪怪" then  
5         PushDebugMessage(fmt("名称: %s 类型: %s tid:%s", temp.name,  
temp.class,temp.target))  
6         local tmp = Target:Attack(temp.target, 100, 0)  
7         PushDebugMessage(tmp)  
8         Sleep(1000)  
9     end  
10 end
```

-- 编号: NO.15  
-- 方法名: Target:PetAttack(aTargetID)  
-- 描述: 珍兽攻击某个目标  
-- 参数1: aTargetID 整数型 通过文章开关的demo获取对应的目标ID  
-- 返回值: nil  
-- 示例:

```
1 local tEnv = EnumEnvObj()  
2 for i = 1, #tEnv do  
3     local temp = tEnv[i]  
4     if temp.name == "天山小雪怪" then
```

```

5         PushDebugMessage(fmt("名称: %s 类型: %s tid:%s", temp.name,
temp.class,temp.target))
6         local tmp = Target:PetAttack(temp.target)
7         PushDebugMessage(tmp)
8         Sleep(1000)
9     end
10 end

```

-- 编号: NO.16  
-- 方法名: Target:Select(aTid, aSleep)  
-- 描述: 选中目标  
-- 参数1: aTid, 整数型, 目标ID  
-- 参数2: aSleep, 整数型, 延时时间毫秒值  
-- 返回值: nil  
-- 示例:

```

1 local tEnv = EnumEnvObj()
2 for i = 1, #tEnv do
3     local temp = tEnv[i]
4     if temp.name == "天山小雪怪" then
5         PushDebugMessage(fmt("名称: %s 类型: %s tid:%s", temp.name,
temp.class,temp.id))
6         local tmp = Target:Select(temp.id,100)
7         PushDebugMessage(tmp)
8         Sleep(1000)
9     end
10 end

```

-- 编号: NO.17  
-- 方法名: Target:FindBuff(aTarget, aBuffName)  
-- 描述: 查找目标是否有状态  
-- 参数1: 目标环境对象 可通过二叉遍历获取 EnumEnvObj  
-- 参数2: 状态名  
-- 返回值: true:获取到了状态 / false:未获取到指定状态名

-- 示例：

```
1 local tEnv = EnumEnvObj()
2 for i = 1, #tEnv do
3     local temp = tEnv[i]
4     if temp.name == "飘飘鸭" then
5         PushDebugMessage(fmt("名称: %s 类型: %s tid:%s", temp.name,
6             temp.class,temp.target))
7         local tmp = Target:FindBuff(temp, "僵硬: 移动速度减慢")
8         PushDebugMessage(tmp)
9         Sleep(1000)
10    end
11 end
```

-- 编号：NO.18

-- 方法名：CDG:checkDiFu()

-- 描述：离开地府

-- 示例：

```
1 CDG.defaultCity = 1 -- 默认出地府地图苏州
2 CDG:checkDiFu()
```

-- 编号：NO.19

-- 方法名：CDG:checkHP\_Player()

-- 描述：给玩家(界面设置的玩家)加血

-- 返回值：返回加血玩家的名字

-- 示例：

```
1 local add_hp_name = CDG:checkHP_Player()
2 if add_hp_name == "" then
3     -- 给自己加血 参数1是否允许共生回血
4     CDG:checkHP(true)
5 end
```

-- 编号：NO.20  
-- 方法名：CDG:checkHP(true)  
-- 描述：给自己加血  
-- 参数1：true:允许共生回血 / false:不允许共生回血  
-- 返回值：  
-- 示例：

```
1 CDG:checkHP(true)
```

-- 编号：NO.21  
-- 方法名：CDG:checkMP(true)  
-- 描述：给自己加蓝  
-- 参数1：true:允许血祭 / false:不允许血祭  
-- 返回值：  
-- 示例：

```
1 CDG:checkMP(true)
```

-- 编号：NO.22  
-- 方法名：CDG:checkHP\_Pet(false)  
-- 描述：给珍兽加血  
-- 参数1：true:允许加血 / false:不允许加血  
-- 返回值：无  
-- 示例：

```
1 CDG:checkHP_Pet(false)
```

-- 编号：NO.23  
-- 方法名：CDG:checkDXYS()  
-- 描述：掉血隐身

-- 示例：

```
1 CDG.dxys = true
2 CDG:checkDXYS()
```



# 背包仓库锦囊

-- 编号: NO.1  
-- 方法名: Bank\_GetItem(aItemName,aGetNum,aBindState,aLockState,aPackUpBag,aCallBack)  
-- 描述: 仓库取物  
-- 参数1: aItemName 必填 物品名或者类型, 多个名字用空格隔开  
-- 参数2: aGetNum 选填 取出物品数数量, 通常是一个格子的数量。不会是具体的数值  
-- 参数3: aBindState 选填 取出的绑定属性, 0=不管绑定与否, 1=不绑定, 2=绑定  
-- 参数4: aLockState 选填 取出的锁定属性, 0=不管绑定与否, 1=不绑定, 2=绑定  
-- 参数5: aPackUpBag 选填 逻辑型 整理背包  
-- 参数6: aCallBack 选填 回调方法,返回false结束取物  
-- 返回值: 取出数量,是否超时  
-- 示例:

```
1 Bank_GetItem("黄纸",1,0,0,1,aCallBack)
```

-- 编号: NO.2  
-- 方法名: TempBank2Bag(aAllItem, aBindState, aTakeNum)  
-- 描述: 锦囊取物  
-- 参数1: aAllItem 文本型,物品名或类型, 多个用空格隔开  
-- 参数2: aBindState 可空nil,取出的绑定属性 0无所谓 1不绑的 2绑定的  
-- 参数3: aTakeNum 可空nil,取出的物品数量  
-- 返回值: 逻辑型,是否取到  
-- 例, 取出物品黄纸, 炼玉10级,

```
1 TempBank2Bag("黄纸 炼玉10级",0,10)
```

-- 编号: NO.3

-- 方法名: Bag\_Move2TempBank(altemName, aBindState, aBagItem, aPackUpBag, aNotWithUserConfig)  
-- 描述: 背包物品存入锦囊  
-- 参数1: altemName: 要存锦囊的名称,多个空格分开  
-- 参数2: aBindState: 可空nil,绑定属性 0无所谓 1不绑的 2绑定的  
-- 参数3: aBagItem: 可空nil是道具+材料栏物品数组,你可以对物品数组过滤处理后传入  
-- 参数4: aPackUpBag: 可空nil,是否整理锦囊  
-- 参数5: aNotWithUserConfig:可空nil,若为true不包含界面设置的物品,若为false需先调用 GetConfig() 获取用户配置  
-- 返回值: 移动的数量

```
1 Bag_Move2TempBank(aItemName, aBindState, aBagItem, aPackUpBag, aNotWithUserConfig)
```

-- 编号: NO.4  
-- 方法名: Bank\_SaveItem(altemName, aKeepMoney, aBindState)  
-- 描述: 仓物存物品  
-- 参数1: altemName 文本型 要存入的物品,多个空格分开  
-- 参数2: aKeepMoney 可空nil,-1表示不存钱, 其它表示存钱时保留多少(铜)  
-- 参数3: aBindState: 可空nil,0无所谓 1不绑的 2绑定的  
-- 返回值:  
-- 示例:

```
1 Bank_SaveItem("黄纸 炼玉10级", 0, 0)
```

-- 编号: NO.5  
-- 方法名: Bank\_SaveMoney(aKeepTong)  
-- 描述: 仓库存钱  
-- 参数1: aKeepTong 可空nil为不保留,整数型保留多少铜  
-- 返回值:  
-- 示例: 保留0铜, 全部存入仓库

```
1 Bank_SaveMoney(0)
```

```
-- 编号：NO.6
-- 方法名：Bank_GetMoney(aGetMoney, aToSceneName)
-- 描述：仓库取钱
-- 参数1 aGetMoney，整型值，正数表示取多少铜，负数表示仓库留多少铜
-- 参数2 是否指定地图
-- 返回值：
-- 示例：洛阳仓库取1金
```

```
1 Bank_GetMoney(10000 , "洛阳")
```

```
-- 编号：NO.7
-- 方法名：Bag:EnumAllObj(aWithNullObj)
-- 描述：遍历背包物品(道具栏+材料栏)
-- 参数1：aWithNullObj：可空nil不包含,是否包含背包空位置
-- 返回值：物品数组，详情如下
-- 字段属性：
    id：物品ID,
    obj:物品对象地址,为0表示是一个空位,
    name:名称,
    class：类型名,
    itemType：整数类型,
    childType：子类型,
    itemIndex：物品表索引,
    level：物品等级、
    index：在背包的索引,
    xing：装备星,
    count：物品数量,
    bind：绑定状态,
    lock：锁定状态,
    sale:出售状态,
    author：装备制造者,
    ybpValue：元宝票面值
-- 示例：遍历背包里面的所有物品
```

```

1 local tItem = Bag:EnumAllObj()
2 for i = 1, #tItem do
3     local tmp = tItem[i]
4     local strMsg = fmt("位置: %d, 名称: %s, 文本类型: %s 整数类型: %d, 绑定: %d", tmp.index, tmp.name, tmp.class, tmp.itemType, tmp.bind)
5     PushColorMsg(strMsg)
6 end

```

-- 编号: NO.7  
 -- 方法名: Bag:EnumDaoJuItem(aWithNullObj)  
 -- 描述: 遍历道具包物品,  
 -- 参数1: aWithNullObj: 可空nil不包含,是否包含背包空位置  
 -- 返回值: 返回数组结构, 如上【编号: NO.7】  
 -- 示例: 返回当前背包道具栏物品信息

```

1 local tItem = Bag:EnumDaoJuItem()
2 for i = 1, #tItem do
3     local tmp = tItem[i]
4     local strMsg = fmt("位置: %d, 名称: %s, 文本类型: %s 整数类型: %d, 绑定: %d", tmp.index, tmp.name, tmp.class, tmp.itemType, tmp.bind)
5     PushColorMsg(strMsg)
6 end

```

-- 编号: NO.8  
 -- 方法名: Bag:EnumCailiaoItem(aWithNullObj)  
 -- 描述: 遍历材料包物品  
 -- 参数1: aWithNullObj: 可空nil不包含,是否包含背包空位置  
 -- 返回值: 返回数组结构, 如上【编号: NO.7】  
 -- 示例: 返回当前背包材料栏物品信息

```

1 local tItem = Bag:EnumCailiaoItem()

```

-- 编号：NO.9  
-- 方法名：Bag:EnumRenWuItem(aWithNullObj)  
-- 描述：遍历任务包物品信息  
-- 参数1：aWithNullObj：可空nil不包含,是否包含背包空位置  
-- 返回值：返回数组结构，如上【编号：NO.7】  
-- 示例：返回当前背包任务栏物品信息

```
1 local tItem = Bag:EnumRenWuItem()
```

-- 编号：NO.10  
-- 方法名：Bag:EnumTempBagItem(aWithNullObj)  
-- 描述：遍历锦囊包物品  
-- 参数1：aWithNullObj：可空nil不包含,是否包含背包空位置  
-- 返回值：返回数组结构，如上【编号：NO.7】  
-- 示例：返回当前锦囊物品信息

```
1 local tItem = Bag:EnumTempBagItem()
```

-- 编号：NO.11  
-- 方法名：Bag:EnumBankItem(aWithNullObj)  
-- 描述：仓库物品,需要打开仓库  
-- 参数1：aWithNullObj：可空nil不包含,是否包含背包空位置  
-- 返回值：返回数组结构，如上【编号：NO.7】  
-- 示例：返回当前仓库物品信息。

```
1 local tItem = Bag:EnumBankItem()
```

-- 编号：NO.12

-- 方法名: Bag:EnumLootItem(aWithNullObj)  
-- 描述: 地面包裹物品, 需要打开地面包  
-- 参数1: aWithNullObj: 可空nil不包含,是否包含背包空位置  
-- 返回值: 返回数组结构, 如上【编号: NO.7】  
-- 示例: 返回当前地面包裹物品信息。

```
1 local tItem = Bag:EnumLootItem()
```

-- 编号: NO.13  
-- 方法名: Bag:EnumJiaoYiSelfItem(aWithNullObj)  
-- 描述: 交易时自己的物品  
-- 参数1: aWithNullObj: 可空nil不包含,是否包含背包空位置  
-- 返回值: 返回数组结构, 如上【编号: NO.7】  
-- 示例: 返回交易栏里面的物品信息

```
1 local tItem = Bag:EnumJiaoYiSelfItem()
```

-- 编号: NO.14  
-- 方法名: Bag:EnumJiaoYiOtherItem(aWithNullObj)  
-- 描述: 交易时对方的物品  
-- 参数1: aWithNullObj: 可空nil不包含,是否包含背包空位置  
-- 返回值: 返回数组结构, 如上【编号: NO.7】  
-- 示例: 返回 交易时对方的物品信息

```
1 local tItem = Bag:EnumJiaoYiOtherItem()
```

-- 编号: NO.15  
-- 方法名: Bag:PackUp(aTabIndex)  
-- 描述: 整理背包

-- 参数1: aTabIndex: -1全整理,0: 道具包, 1 材料包  
-- 返回值:  
-- 示例:

```
1 Bag:PackUp(0)
```

-- 例: 搜索物品名 您也可以直接使用内部封装的 Bag:SearchItemName()  
-- 参数1,要搜索的物品名称,多个空格隔开  
-- 参数2: 被搜索的数组,可空nil道具+材料  
-- 返回搜索到的数组

```
1 function Bag_SearchItemName(aItemName, aObj)
2     aObj = aObj or Bag:EnumAllObj()
3     local tName = Text2Table(aItemName, " ") --通用有此接口说明
4     local tSearch = GetTableFieldValue(aObj, "name", tName) --通用
    有此接口说明
5     return tSearch
6 end
```

-- 例: 搜索物品类型, 您也可以直接使用内部封装的 Bag:SearchItemType()  
-- 参数1,要搜索的物品类型,多个空格隔开  
-- 参数2: 被搜索的数组,可空nil道具+材料  
-- 返回搜索到的数组

```
1 function Bag_SearchItemType(aTypeName, aObj)
2     aObj = aObj or Bag:EnumAllObj()
3     local tName = Text2Table(aTypeName, " ")
4     local tSearch = GetTableFieldValue(aObj, "class", tName)
5     if string.find(aTypeName, " 所有宝石 ", nil, true) then -- 如果
        物品类型有 所有宝石
6         local tGem = {}
7         for i = 1, #aObj do
8             local tmp = aObj[i]
9             if tmp.itemType == 5 then
```

```

10             table.insert(tGem, tmp)
11         end
12     end
13     TableInsertTable_arr(tSearch, "index", tGem) -- 按背包物品
        序号，避免重复插入
14     end
15
16     return tSearch
17 end

```

-- 编号：NO.16

-- 方法名：Bag:GetItemCount(altemName, aCheckBind, aCheckLock, altem)

-- 描述：获取物品数量

-- 参数1：物品名称

-- 参数2：可空nil 绑定 0无所谓 1不绑的 2绑定的

-- 参数3：可空nil 锁定 0无所谓 1不绑的 2绑定的

-- 参数4：可空nil道具+材料,指定统计的物品数组

-- 返回值：找到的数量

-- 示例：

```

1 Bag:GetItemCount(aItemName, aCheckBind, aCheckLock, aItem)
2 local n = Bag:GetItemCount("愿灵泉") -- 取背包的物品数量
3 local n = Bag:GetItemCount("愿灵泉", nil, nil, Bag:EnumTempBagItem())
    -- 取锦囊的物品数量

```

-- 编号：NO.17

-- 方法名：Bag:FindItemInfo(altemName, aCheckBind, aCheckLock, altem)

-- 描述：查找物品

-- 参数1：altemName 单个物品名称

-- 参数2：aCheckBind 绑定 0无所谓 1不绑的 2绑定的

-- 参数3：aCheckLock 绑定 0无所谓 1不绑的 2绑定的

-- 参数4：altem：可空,指定被查找的物品数组

-- 返回值：被查找到的物品

-- 示例：



```

1 Bag:FindItemInfo(aItemName, aCheckBind, aCheckLock, aItem)
2
3 -- 示例：使用背包物品1，您也可以使用内部封装的 Bag:UseItemByName
4 function Bag_UseItemByName(aItemName, aSleep, aItem)
5     local tItem = Bag:FindItemInfo(aItemName, 0, 0, aItem)
6     if #tItem > 0 then
7         Bag:UseItem(tItem[1].index, aSleep) -- 直接使用
8         return true
9     end
10    return false
11 end
12 -- 示例：使用背包物品2，有时1无法成功。您也可以使用内部封装的 Bag:UseItemByName_click
13 function Bag_UseItemByName_click(aItemName, aSleep)
14     local tItem = Bag:FindItemInfo(aItemName)
15     if #tItem > 0 then
16         local tmp = tItem[1]
17         Bag:ItemBtnClicked(tmp.index) -- 点击使用物品
18         if aSleep then
19             Sleep(aSleep)
20         end
21         return true
22     end
23     return false
24 end

```

-- 编号：NO.18

-- 方法名：Bag\_DeleteBagItem(aOtherName, aBagItem, aMustCheckMiBao)

-- 描述：自定义销毁背包（道具+材料）物品，默认销毁分控界面里面的物品

-- 参数1：aOtherName 可为空 单独额外销毁的物品名称

-- 参数2：aBagItem 可为空 物品数组，不填就是遍历道具+材料

-- 参数3：aMustCheckMiBao 可为空

-- 返回值：销毁物品格子的数量（一个格子不管物品数量算1个格子）

```

1 GetConfig() -- 用于实时读取当前分控界面销毁物品里面的数据
2 Bag_DeleteBagItem()

```

3

```
4 local ret = Bag_DeleteBagItem("花生口粮 玉米口粮")
```

-- 编号: NO.19  
-- 方法名: Bag\_DeleteTempBagItem(aOtherName)  
-- 描述: 自定义销毁 (锦囊) 物品  
-- 参数1: aOtherName 可为空 单独额外销毁的物品名称  
-- 返回值:

```
1 GetConfig() -- 用于实时读取当前分控界面销毁物品里面的数据  
2 Bag_DeleteTempBagItem()
```

-- 编号: NO.20  
-- 方法名: Bag:GetSpaceNum(aIndex)  
-- 描述: 背包剩余空间  
-- 参数1: aIndex 整数型 0=道具栏, 1=材料栏, 2=任务栏  
-- 返回值: 整数型 返回具体剩余空白格子数量

```
1 local nRet = Bag:GetSpaceNum(0)  
2 PushColorMsg(nRet)  
3 local nRet1 = Bag:GetSpaceNum(1)  
4 PushColorMsg(nRet1)  
5 local nRet2 = Bag:GetSpaceNum(2)  
6 PushColorMsg(nRet2)
```

-- 编号: NO.21  
-- 方法名: EatBindItem(nCount,aEatItemName,aMustOpenBox)  
-- 描述: 吃掉消化物品  
-- 参数1: nCount 整数型 消化次数  
-- 参数2: aEatItemName 指定消化的东西, 不指定填写nil  
-- 参数3: aMustOpenBox 是否必须消化副本箱子  
-- 返回值: 整数型 返回具体剩余空白格子数量

```
1 TempBank2Bag(GetFubenBox())
2 EatBindItem(9999,nil,true)
```

-- 编号: NO.22  
-- 方法名: Shop\_BuyItem(nItemName,nCount)  
-- 描述: 杂货商店购买物品  
-- 参数1: nItemName 文本型 购买物品名称  
-- 参数2: nCount 整数型 数量  
-- 返回值:

```
1 GoToSceneNPC("芮福祥",347, 284, LUOYANG_ID)
2 QuestFrameOptionClicked_Item("买卖杂货",1, 800)
3 Shop_BuyItem("沙蚕",100)
```

-- 编号: NO.23  
-- 方法名: MainBankTakeItem()  
-- 描述: 取出物品, 默认取出界面里面配置的物品  
-- 返回值:

```
1 GetConfig()
2 MainBankTakeItem()
```

-- 编号: NO.24  
-- 方法名: Bag\_SellItem(otherName, aScriptName)  
-- 描述: 单独出售物品  
-- 参数1: otherName文本型 可空, 单独需要卖出的物品名称, 如果不填写, 默认是界面里面的卖出物品名称  
-- 参数2: aScriptName 文本型 执行的脚本名称, 一般你们调用不用输入, 可空  
-- 返回值:

```
1 Bag_SellItem()
```

-- 编号：NO.25  
-- 方法名：Bag\_BuyPetItem(aScriptName)  
-- 描述：单独买珍兽物品，默认是界面配置的保护物品  
-- 参数2：aScriptName 文本型 执行的脚本名称，一般你们调用不用输入，可空  
-- 返回值：

```
1 Bag_BuyPetItem()
```

# NPC任务寻路

-- 编号：NO.1  
-- 方法名：TalkNPC(aName,aTick)  
-- 描述：对话NPC,并延时，延时时间可以为空  
-- 参数1：文本型 名字  
-- 参数2：整数型 延时毫秒,可空  
-- 返回值：无  
-- 示例：TalkNPC("孙行者")

```
1 TalkNPC("孙行者")
```

-- 编号：NO.2  
-- 方法名：OpenTalk(aNPC\_ID,aTime)  
-- 描述：对话NPC，并延时。延时时间可以为空  
-- 参数1：整数型 npc id  
-- 参数2：整数型 延时毫秒,可为空  
-- 返回值：无  
-- 示例：OpenTalk(123,500)

```
1 OpenTalk(123,500)
```

-- 编号：NO.3  
-- 方法名：GetSceneNameByID(aSceneID)  
-- 描述：用地图ID取地图名  
-- 参数1：地图id  
-- 返回值：1个文本型  
-- 示例：

```
1 local ret = GetSceneNameByID(0)  
2 PushColorMsg(ret)
```

-- 方法名: GetSceneIDByName(aSceneName)  
-- 描述: 用地图名取地图ID  
-- 参数1: 地图名  
-- 返回值: 1个整数型  
-- 示例: local rSceneID = GetSceneIDByName("洛阳")

```
1 local rSceneID = GetSceneIDByName("洛阳")
```

-- 编号: NO.3  
-- 方法名: MoveTo(aX, aY, aSceneID, aSpace, aMount, aCallBack, aCallBackMove, aMoneyRoad, aTransferItem, aFollowName, aFollowCallBack)  
-- 描述: 寻路(简单使用的话,一般输入前3个参数就可以)  
-- 参数1: aX, 整数型, X坐标  
-- 参数2: aY, 整数型, Y坐标  
-- 参数3: aSceneID, 整数型, 地图ID, 地图id,可为: -1, -1表示当前地图  
-- 参数4: aSpace, 整数型, 与坐标距离超出多少寻路  
-- 参数5: aMount 逻辑型, 是否用坐骑, 默认true  
-- 参数6: aCallBack: 回调函数, 默认为true, 用来处理死亡或其它需求的回调, 返回false结束寻路  
-- 参数7: aCallBackMove: 回调函数, 默认为true, 比较常用, 可以更新坐标(例如跟随队长), 返回false结束寻路  
-- 参数8: aMoneyRoad: 是否收费传送,  
-- 参数9: aTransferItem: 收费使用定位符道具  
-- 参数10: aFollowName: 字符型, 跟随人物名称  
-- 参数11: aFollowCallBack: 回调: 跟随回调, 默认返回为true

```
1 -- 例: 跟随队长,只是简单展示MoveTo高级用法,您可以直接使用内部封装的GoToLeader()  
  效果更佳  
2 function Demo_GoToLeader(aMoveCallBack, aCallBack, bUseShouFeiNpc  
  , bUseDingWeiFu, aKeepDist)  
3     -- 回调1  
4     local _cb1 = function()  
5         if Player:GetState() == 9 then  
6             PushDebugMessage("寻路:角色死亡自动出窍,结束寻路!")  
7             GetLuaValue("dmReliveHandle(-1)") -- 出窍  
8             Sleep(1000)  
9             return false
```

```

10         end
11         return true
12     end
13     -- 回调2
14     local _cb2 = function(fX, fY, nScene)
15         local leader, isleader = Team:GetLeaderInfo()
16         if not leader then
17             PushColorMsg("没有队长数据")
18             return false, nil
19         end
20         if isleader then
21             PushColorMsg("我是队长, 不用跟随")
22             return false, nil
23         end
24         if Player:IsFollow() then
25             PushColorMsg("组队跟随")
26             return false, nil
27         end
28         if FollowFilterScene(leader.sceneID) ~= -1 then
29             DebugStr("跟随队长: 队长不在正确的地图, 暂不跟随!")
30             return false, nil
31         end
32         DebugStr("跟随队长: 移动到 x:%s y:%s scene:%s", 1, math.floor(leader.x), math.floor(leader.y), leader.sceneID)
33         if leader.x > 0 and (leader.x ~= fX or leader.y ~= fY or leader.sceneID ~= nScene) then
34             AutoRunToTarget(leader.x, leader.y, leader.sceneID, true)
35             Sleep(300)
36             return true, leader -- 此时返回的leader会更新寻路坐标, leader 需要有x y sceneID字段
37         end
38         return true, nil
39     end
40     --
41     aKeepDist = aKeepDist or 3
42     aMoveCallBack = aMoveCallBack or _cb2
43     aCallBack = aCallBack or _cb1
44     local leader, isleader = Team:GetLeaderInfo() -- 获取队长数据
45     if not leader or isleader then

```

```

46         return -1
47     end
48     local leaderObj = FindObjByGUID(leader.guid) -- 看看队长在附近吗
49     if leaderObj then
50         leader.x = leaderObj.x
51         leader.y = leaderObj.y
52     end
53     -- 开始寻路
54     MoveTo(leader.x, leader.y, leader.sceneID, aKeepDist, nil, aC
allBack, aMoveCallBack, bUseShouFeiNPC, bUseDingWeiFu)
55 end
56
57 -- 例：移动并对话NPC，您可以直接使用内部封装的MoveToNPC()
58 -- 参数1: npc名称
59 function Demo_MoveToNPC(aNPCName, ...)
60     MoveTo(...)
61     Sleep(500)
62     TalkNpc(aNPCName, 500)
63     PlayerStateTimeOut(2, 10) -- 人物走路状态为2时等待，10秒超时
64 end
65
66 -- 例：非副本移动，您可以直接使用内部封装的GoToScene()
67 -- 传了一个默认的MoveCallback_NoFuben的回调判断是否在副本
68 function Demo_GoToScene(aX, aY, aSceneID, aSpace, aMount, aCallBa
ck, aCallBackMove, aMoneyRoad, aTransferItem)
69     MoveTo(aX, aY, aSceneID, aSpace, aMount, aCallBack, MoveCallb
ack_NoFuben, aMoneyRoad, aTransferItem)
70 end
71
72
73 -- 例：非副本移动并对话npc，您可以直接使用内部封装的GoToSceneNPC()
74 -- 传了一个默认的MoveCallback_NoFuben的回调判断是否在副本
75 function Demo_GoToSceneNPC(aNPCName, aX, aY, aSceneID, aSpace, aM
ount, aCallBack, aCallBackMove, aMoneyRoad, aTransferItem)
76     MoveTo(aX, aY, aSceneID, aSpace, aMount, aCallBack, MoveCallb
ack_NoFuben, aMoneyRoad, aTransferItem)
77     Sleep(500)
78     TalkNpc(aNPCName, 500)
79     PlayerStateTimeOut(2, 10)
80 end

```



```
81
82
83 -- 例
84 GoToScene(358, 422, 762) -- 在副本自动退出寻路
85 GoToSceneNPC("萧陵", 358, 422, 762) -- 在副本自动退出寻路
86 MoveTo(358, 422, 762) -- 副本内会导致一直在寻路去金陵，无法退出
87 MoveToNPC("萧陵", 358, 422, 762) -- 副本内会导致一直在寻路去金陵，无法退出
```

```
-- 编号：NO.4
-- 方法名：GoToGuild()
-- 描述：进入帮派
-- 返回值：false没有帮派
```

```
1 GoToGuild()
```

```
-- 编号：NO.5
-- 方法名：GoToGuiShi()
-- 描述：前往鬼市
-- 返回值：无
```

```
1 GoToGuiShi()
```

```
-- 编号：NO.6
-- 方法名：GoToZhuangYuan(aMustTeam)
-- 描述：进入庄园
-- 参数1：true:进入组队庄园 / false:进入自己的庄园
-- 返回值：无
```

```
1 GoToZhuangYuan(false)
```

```
-- 编号：NO.7
-- 方法名：GetPlayerMission(aOption, aCmp)
```

-- 描述：获取任务列表指定任务(明文)  
-- 参数1：字符型：任务名  
-- 参数2：整数型：任务名对比模式 1：完全相同，3：部分相同  
-- 返回值1：任务索引，没有此任务返回-1  
-- 返回值2：任务id，没有此任务返回-1  
-- 返回值3：任务的脚本id，没有此任务返回-1  
-- 示例：

```
1 local ret1,ret2,ret3 = GetPlayerMission("惩凶打图",1)
2 PushColorMsg("任务索引 = "..ret1..", 任务id = "..ret2..", 脚本id = "..ret3)
```

-- 编号：NO.8  
-- 方法名：QuestFrameMissionAbnegate(aScriptID)  
-- 描述：放弃任务  
-- 参数1：任务的脚本ID，通过上个方法获取任务的脚本ID  
-- 返回值：无

```
1 QuestFrameMissionAbnegate(229020) -- 通过上面demo获取到惩凶打图脚本ID为229020
```

-- 编号：NO.9  
-- 方法名：GetPlayerMission\_Item(aOption)  
-- 描述：获取任务列表指定任务(字典)，通过 【编号：NO.10】 获取任务字典数据  
-- 参数：字符型：任务名  
-- 返回值1：任务索引，没有此任务返回-1  
-- 返回值2：任务id，没有此任务返回-1  
-- 返回值3：任务的脚本id，没有此任务返回-1  
-- 示例：

```
1 local ret1,ret2,ret3 = GetPlayerMission_Item("#{SQXY_09061_4}")
```

-- 编号: NO.10  
-- 方法名: GetPlayerMissionAllName()  
-- 描述: 获取任务列表所有任务  
-- 返回值: 所有任务字典名,已用空格隔开  
-- 示例:

```
1 local aMissionName = GetPlayerMissionAllName()  
2 PushColorMsg(aMissionName)  
3 MessageBox(aMissionName)
```

-- 编号: NO.11  
-- 方法名: GetMissionVariable(nIndex, aIndex)  
-- 描述: 取任务参数  
-- 参数1: 任务索引 通过上述方法获取  
-- 参数2: 参数索引 0-7  
-- 返回值: 一个整数型

```
1 local ret1,ret2,ret3 = GetPlayerMission("惩凶打图",1)  
2 PushColorMsg("任务索引 = "..ret1..", 任务id = "..ret2..", 脚本id = "..  
    ret3)  
3 local nRet = GetMissionVariable(ret1, 0)  
4 PushColorMsg(nRet)
```

-- 编号: NO.12  
-- 方法名: QuestFrameAcceptClicked(aSleep)  
-- 描述: 点击接受任务  
-- 参数1: 可空, 点击后延时毫秒  
-- 返回值:

```
1 QuestFrameAcceptClicked(100)
```

-- 编号: NO.13  
-- 方法名: QuestFrameNextClicked(aSleep)  
-- 描述: 点击继续  
-- 参数1: 可空延时  
-- 返回值:

```
1 QuestFrameNextClicked()
```

-- 编号: NO.14  
-- 方法名: QuestFrameMissionComplete(aIdItem, aSleep)  
-- 描述: 点击完成任务  
-- 参数1: 可空默认0, 要选择的完成奖励  
-- 可空点击后延时  
-- 返回值:

```
1 QuestFrameMissionComplete(0, 100)  
2 QuestFrameMissionComplete()
```

-- 编号: NO.15  
-- 方法名: GetMissionComplete(aMissionIndex)  
-- 描述: 获取任务完成度(其实就是封装上面11的参数索引传入了0)  
-- 参数1: 任务索引  
-- 返回值:  
-- 示例:

```
1 local ret1,ret2,ret3 = GetPlayerMission("惩凶打图",1)  
2 PushColorMsg("任务索引 = "..ret1..", 任务id = "..ret2..", 脚本id = "..  
   ret3)  
3 local n = GetMissionComplete(ret1)
```

-- 编号: NO.16  
-- 方法名: GetNPCEventListAllItem()  
-- 描述: 获取npc对话框所有内容,已用空格分开

-- 返回值:

```
1 local strItem = GetNPCEventListAllItem()
```

-- 编号: NO.17

-- 方法名: GetNPCEventList\_Item(0)

-- 描述: 获取npc对话框第几项内容

-- 参数1: 整数型 第几项

-- 返回值:

-- 示例:

```
1 local strItem = GetNPCEventList_Item(0)
```

-- 编号: NO.18

-- 方法名: DialogOptionClicked("离开副本", 1, 700)

-- 描述: 点击npc选项(明文模式)

-- 参数1: 文本型 选项名称

-- 参数2: 整数型 可空默认1, 名称对比模式 1: 完全相同, 2: 部分相同 3部分相同

-- 参数3: 整数型 可空, 延时毫秒

-- 返回值: 逻辑型, 是否点击到选项

-- 示例 假设实际名称是 "离开副本",以下3种均可

```
1 -- 示例1
2 DialogOptionClicked("离开副本", 1, 700)
3 -- 示例2
4 DialogOptionClicked("我现23在想离开副本随变打的字", 2, 700)
5 -- 示例3
6 DialogOptionClicked("离开", 3, 700)
```

-- 编号: NO.19

-- 方法名: QuestFrameOptionClicked\_Item("#{FMZLW\_111219\_03}", 1, 700)

-- 描述: 点击npc选项(字典模式)

-- 参数1: 文本型 选项字典名称  
-- 参数2: 整数型 名称对比模式 1: 完全相同, 2: 部分相同 3部分相同  
-- 参数3: 整数型 可空, 延时毫秒  
-- 返回值: 逻辑型, 是否点击到选项  
-- 示例:

```
1 QuestFrameOptionClicked_Item("#{FMZLW_111219_03}", 1, 700)
2 PushColorMsg("#{FMZLW_111219_03}")
```

-- 编号: NO.20  
-- 方法名: GetMissionDesc(aMissionName)  
-- 描述: 取任务描述  
-- 参数1: 任务名(字典) 通过 【GetPlayerMissionAllName()】 获得任务字典  
-- 返回值: 1个文本型  
-- 示例:

```
1 local nRet = GetMissionDesc("#{SQXY_09061_4}")
2 PushColorMsg(nRet)
3 MessageBox(nRet)
```

-- 编号: NO.21  
-- 方法名: IsWindowVisible("Quest", "this:Hide()")  
-- 描述: 窗口是否可视  
-- 参数1: 窗口名  
-- 参数2: 可空, 如果可视的话, 执行的lua文本  
-- 参数3: 可空, 如果不可视 执行的lua文本  
-- 返回值: 逻辑型  
-- 示例:

```
1 IsWindowVisible("Quest", "this:Hide()")
```

-- 编号：NO.22  
-- 方法名：XiaoChangHuiXue(aHpCmp,aUseMount)  
-- 描述：校场回血  
-- 参数1：整数型 百分比 1-100，血低于多少百分比去执行加血。  
-- 参数2：逻辑型 是否使用坐骑，  
-- 返回值：逻辑型  
-- 示例：

```
1 -- 血低于60%就去加血  
2 XiaoChangHuiXue(60)
```

-- 编号：NO.23  
-- 方法名：RunTo(aX,aY,aDist,aSleep)  
-- 描述：寻路简单版，需要详细的请查看MoveTo, 此寻路不只限于当前地图，不需要指定场景ID  
-- 参数1：整数型 aX x坐标  
-- 参数2：整数型 aY y坐标  
-- 参数3：整数型 距离，可以为空，  
-- 参数4：整数型 延时，可以为空，  
-- 返回值：逻辑型  
-- 示例：

```
1 RunTo(100,99)
```

# 通用

-- 编号: NO.1  
-- 方法名: `GetConfig()`  
-- 描述: 用户的配置,执行后所有的界面配置保存在全局table型变量 `g_config`  
-- 获取正在运行的脚本名 `g_runing_script`  
-- 注意: 一些其它接口需要先调用此接口获取用户配置,比如存仓、锦囊存物等  
-- 参数: 无  
-- 返回值: 无  
-- 示例:

```
1 GetConfig()  
2 -- 获取百花缘环数  
3 PushDebugMessage(g_config.baihuayuan.huanshu)  
4 PushDebugMessage("正在运行: " .. g_runing_script)
```

-- 编号: NO.2  
-- 方法名: `GetLuaValue()`  
-- 描述: 获取/执行游戏的lua  
-- 参数1: lua文本  
-- 参数2: 可空nil,表示没有返回值。可以是 `n b s` 的组合, `s`代表文本型 `n`代表整数型 `b`代表逻辑型  
-- 返回值: 按参数2决定,返回类型按参数2顺序决定  
-- 示例:

```
1 local strName, nLevel = GetLuaValue("local name = Player:GetName()  
   local level = Player:GetLevel() return name,level", "sn")  
2 PushColorMsg("我的名字:" .. strName .. ",等级: " .. nLevel)
```

-- 编号: NO.3  
-- 方法名: `LoadScript("脚本名")`  
-- 描述: 执行脚本,支持自带脚本和自定义目录脚本  
-- 参数1: 脚本名称



-- 返回值：若你的脚本有返回值

-- 示例：

```
1 local value = LoadScript("脚本名")
```

-- 编号：NO.4

-- 方法名：LoadUserPathScript("c:\\a.lua")

-- 描述：执行任意路径的脚本 注意:单斜杠需要转换成双斜杠

-- 参数1：脚本路径

-- 返回值：无

-- 示例：

```
1 LoadUserPathScript("c:\\a.lua")
```

-- 编号：NO.5

-- 方法名：g\_json.encode(nTable)

-- 描述：全局处理函数，将LUA表结构转换成json结构

-- 参数1：nTable 表类型

-- 返回值：JSON对象

-- 示例

```
1 local table = {  
2     ["name"] = "张三";  
3     ["sex"] = "男";  
4 }  
5  
6 g_json.encode(table)
```

-- 编号：NO.6

-- 方法名: `g_json.decode(nString)`  
-- 描述: 将json字符串结构转换成Lua的表结构  
-- 参数1: nString 字符型  
-- 返回值: lua表结构  
-- 示例1

```
1 local nStr = ' [{"name"]="张三";["sex"]="男"} ]';  
2  
3 g_json.decode(nStr)
```

-- 编号: NO.7  
-- 方法名: `SplitString("你 爱 我 ", " ", true)`  
-- 描述: 分割文本  
-- 参数1: 需要分割的文本  
-- 参数2: 分割符  
-- 参数3: 逻辑型,是否移除空文本  
-- 返回值: 文本数组  
-- 示例:

```
1 local arrStr = SplitString("你 爱 我      ", " ", true)  
2 for i = 1, #arrStr do  
3     PushDebugMessage("数组大小" .. #arrStr .. " " .. arrStr[i])  
4 end
```

-- 编号: NO.8  
-- 方法名: `SubString("你又胖了", "又", "了")`  
-- 描述: 取文本中间  
-- 参数1: 源字符串  
-- 参数2: 文本型 欲取字符串前的文本  
-- 参数3: 文本型 欲取字符串后的文本  
-- 返回值: 欲取的文本  
-- 示例:

```
1 local strText = SubString("你又胖了", "又", "了")  
2 PushColorMsg("取到的是胖吗: " .. strText)
```

-- 编号: NO.9  
-- 方法名: `GetNextDaySec()`  
-- 描述: 距离明天还要多少秒  
-- 参数: 无  
-- 返回值: 1个整数型  
-- 示例:

```
1 local nSec = GetNextDaySec()
```

-- 编号: NO.10  
-- 方法名: `ShowTipMsg(文本内容)`  
-- 描述: 弹出一个信息框  
-- 参数1: 文本  
-- 返回值:  
-- 示例:

```
1 ShowTipMsg("名士赠礼满了")
```

-- 编号: NO.11  
-- 方法名: `IsMount()`  
-- 描述: 是否坐骑  
-- 返回值: 1个逻辑型  
-- 示例:

```
1 local bM = IsMount()
```

-- 编号: NO.12  
-- 方法名: `UseMount()`

-- 描述：使用坐骑

```
1 UseMount()
```

-- 编号：NO.13

-- 方法名：Dismount(aSleep)

-- 描述：下坐骑

-- 参数1：aSleep 延时可空

```
1 Dismount(100)
```

-- 编号：NO.14

-- 方法名：TalkSendPrivateMsg("名字","内容")

-- 描述：私聊

```
1 TalkSendPrivateMsg("°君. 莫笑丶","最好用的脚本，大漠脚本")
```

-- 编号：NO.15

-- 方法名：GetSceneID()

-- 描述：取当前地图ID

-- 返回值：整形值

```
1 local nSceneID = GetSceneID()
2 PushColorMsg(nSceneID)
```

-- 编号：NO.16

-- 方法名：Sleep(aSleep)

-- 描述：延时毫秒

-- 参数：aSleep 整数型

```
1 Sleep(500)
```

-- 编号: NO.17  
-- 方法名: GetCurrentSceneName()  
-- 描述: 取当前地图名  
-- 返回值: 文本型

```
1 local sceneName =GetCurrentSceneName()  
2 PushColorMsg(sceneName)
```

-- 编号: NO.18  
-- 方法名: Text2Table(strText, split\_text, field)  
-- 描述: 将文本转换为数组  
-- 参数1: 要转换的文本  
-- 参数2: 分割符,可空nil,默认为空格  
-- 参数3: 字段名,可空nil不用字段  
-- 返回值: 数组  
-- 示例:

```
1 Text2Table(strText, split_text, field)
```

-- 编号: NO.19  
-- 方法名: GetTableFieldValue(aTable, aField, atCmp)  
-- 描述: 算法 将参数1的数组按参数2、参数3过滤  
-- 参数1: 被查找的数组带字段的数组  
-- 参数2: 参数1对比的字段名  
-- 参数3: 用来对比的数组,没有字段名  
-- 返回值: 参数1过滤后的数组  
-- 示例:

```
1 GetTableFieldValue(aTable, aField, atCmp)
```

```
-- 编号：NO.20
-- 方法名：TableInsertTable_arr(aTable, field, tInsert)
-- 描述：算法:将2个数组合并
-- 参数1：被合并的数组
-- 参数2：判断过滤的字段，避免重复插入
-- 参数3：欲插入的新字段
-- 返回值：合并后的数组
-- 示例：
```

```
1 TableInsertTable_arr(aTable, field, tInsert)
```

```
-- 编号：NO.21
-- 方法名：GC_Set(aField, aText)
-- 描述：公共写入配置，配置文件在【配置\公共配置.ini】用于同一台电脑不台角色之间公共读写部分。
如A角色写入，B角色直接取值
-- 参数1：aField  文本型    数据标识名（俗称键）
-- 参数2：aText   文本型    数据的值（俗称值）
-- 参数3：        文本型    节名
-- 返回值：无
-- 示例：
```

```
1 -- 会在配置目录里面生成一个公共配置.ini的文件
2 GC_Set("key", "value", "设置")
```

```
-- 编号：NO.22
-- 方法名：GC_Get(aField)
-- 描述：公共配置接口，读取配置，用于同一台电脑不台角色之间公共读写部分。如A角色写入，B角色
直接取值
-- 参数1：aField  文本型    数据标识名（俗称键）
-- 参数2：节名,文本型可空
```

-- 返回值：无

-- 示例：

```
1 local ret = GC_Get("key","设置")
2 PushColorMsg(ret)
```

-- 编号：NO.23

-- 方法名：IniConfigSet(aField,aText,aFileName,aJieName)

-- 描述：写INI配置文件

-- 参数1：aField 文本型 键

-- 参数2：aText 文本型 值，内容

-- 参数3：aFileName 文本型 保存的ini文件名

-- 参数4：aJieName 文本型 ini配置文件项名称，用“[]”括起来的。

-- 返回值：无

-- 示例：

-- 编号：NO.24

-- 方法名：IniConfigGet(aField,aFileName,aJieName)

-- 描述：读取ini指定文件名的项名称下的键对应的值内容

-- 参数1：aField 文本型 键名

-- 参数2：aFileName 文本型 文件名称

-- 参数3：aJieName 文本型 项名称

-- 返回值：文本型

-- 示例：

-- 编号：NO.25

-- 方法名：OpenPackage(aID)

-- 描述：打开地面背包

-- 参数1：aID 文本型

-- 返回值：无

-- 示例：

```
1 -- 通过遍历环境，获取aID
2 OpenPackage(aID)
```

-- 编号：NO.26  
-- 方法名：OpenTalk(aID, aSleep)  
-- 描述：打开地面背包(地面包，钓鱼都用这个)  
-- 参数1：aID 文本型  
-- 参数2：aSleep 整数型 延时时间，毫秒值，1000毫秒=1秒  
-- 返回值：无  
-- 示例：

```
1 -- 通过遍历环境，获取aID  
2 OpenTalk(aID,200)
```

-- 编号：NO.27  
-- 方法名：OpenDitan(aID)  
-- 描述：打开地摊  
-- 参数1：aID 文本型  
-- 返回值：无  
-- 示例：

```
1 -- 通过遍历环境，获取aID  
2 OpenDitan(aID)
```

-- 编号：NO.28  
-- 方法名：DiscardBagItem(aIndex, aSleep)  
-- 描述：销毁背包指定位置物品  
-- 参数1：aIndex 整数型 背包位置  
-- 参数2：aSleep 整数型 延时时间，毫秒值，1000毫秒=1秒  
-- 返回值：无  
-- 示例：

```
1 -- 通过遍历环境，获取aID  
2 DiscardBagItem(1,200)
```



```

1 -- 编号: N0.29
2 -- 方法名: WaitForTime(aHour, aMinute)
3 -- 描述: 等待时间到 **时**分
4 -- 参数1: aHour      整数型      时, 0-23
5 -- 参数2: aMinute    整数型      分: 0-59
6 -- 返回值: 无
7 -- 示例:
8 WaitForTime(12,12)

```

```

1 -- 编号: N0.30
2 -- 方法名: IsInDebugMsg(aMsg, aIsDel)
3 -- 描述: 是否在调试信息里面, 获取最近一条调试信息
4 -- 参数1: aMsg      文本型      需要比较的文本信息
5 -- 参数2: aIsDel    可空
6 -- 返回值: 逻辑型,
7 -- 示例:
8 if IsInDebugMsg("再次点击F12") then
9     -- 为真的逻辑
10 else
11     -- 为假的逻辑
12 end

```

```

1 -- 编号: N0.31
2 -- 方法名: OffLine(aFalseTrue, aLogStr)
3 -- 描述: 退出游戏
4 -- 参数1: aFalseTrue  逻辑型 是否,false=退出游戏
5 -- 参数2: aLogStr     文本型 退出的日志信息
6 -- 返回值
7 -- 示例:
8 function OfflineCheck()
9     local nSocket = GetSocket()
10    if nSocket == 0xFFFFFFFF then -- -1 4294967295

```

```

11      DebugColorMsg("与服务服务器失去连接...")
12      if m_offline_check_tick == -1 then
13          m_offline_check_tick = GetTickCount()
14      else
15          if GetTickCount() - m_offline_check_tick > 3 * 60 * 1
000 then
16              local str = "与服务服务器失去连接3分钟,10秒后自动下线"
17              PushColorMsg(str)
18              Log(str)
19              Sleep(10 * 1000)
20              -- 参数1: 是否关闭掉线登录
21              -- 参数2: 日志
22              OffLine(false)
23          end
24      end
25  else
26      m_offline_check_tick = -1
27  end
28 end

```

```

1 -- 编号: N0.32
2 -- 方法名: DS_CalcTime(aHour, aMinute)
3 -- 描述: 定时 计算json配置格式中的时间
4 -- 参数1: aHour      整数型 24进制时, 0-23
5 -- 参数2: aMinute   整数型 分钟 0-59
6 -- 返回值: 整数型, 大漠内部时间格式
7 -- 示例:
8 function CalcDSTime(aHour, aMinute)
9     return aHour << 16 | aMinute
10 end
11
12 local h = 13
13 local m = 27
14 local time = PushColorMsg(CalcDSTime(h, m))

```

```

1 -- 编号: N0.33
2 -- 方法名: DS_CalcWeek(a1, a2, a3, a4, a5, a6, a7)
3 -- 描述: 定时 计算json配置格式中的星期,
4 -- 参数: a1-a7      整数型 7个参数代表一周,周几不运行就在对应的位置填nil
5 -- 返回值: 计算json配置格式中的星期
6 -- 示例:
7 function DS_CalcWeek(a1, a2, a3, a4, a5, a6, a7)
8     local shi = 0
9     if a1 then
10         shi = shi | 1 << 1
11     end
12     if a2 then
13         shi = shi | 1 << 2
14     end
15     if a3 then
16         shi = shi | 1 << 3
17     end
18     if a4 then
19         shi = shi | 1 << 4
20     end
21     if a5 then
22         shi = shi | 1 << 5
23     end
24     if a6 then
25         shi = shi | 1 << 6
26     end
27     if a7 then
28         shi = shi | 1 << 7
29     end
30     return shi
31 end
32
33 local nWeek = DS_CalcWeek(1, 2, 3, 4, 5, 6, 7)
34 PushColorMsg(nWeek)

```

```

1 -- 编号: N0.34
2 -- 方法名: GetShanGuiInfo()
3 -- 描述: 获取今天出的山妖山鬼信息
4 -- 返回值: 数组结构: 默认2个城市的数组,
5 -- 示例:
6 local infoTab = GetShanGuiInfo()
7 for i = 1, #infoTab do
8     local tmp = infoTab[i]
9     --                                雁南正东
        帮派名称                                时间戳
10     PushColorMsg(tmp.sceneInfo .. " " .. tmp.cityName .. " " .. tm
        p.tick)
11 end

```

# 玩家

-- 编号: NO.1  
-- 方法名: Player:GetName()  
-- 描述: 取自己的名称  
-- 返回值: 返回1个文本型返回值  
-- 示例:

```
1 local strName = Player:GetName()  
2 PushColorMsg(strName)
```

-- 编号: NO.2  
-- 方法名: Player:GetTID()  
-- 描述: 取自身环境id  
-- 返回值: 返回1个整数型返回值  
-- 示例:

```
1 local nTid = Player:GetTID()  
2 PushColorMsg(nTid)
```

-- 编号: NO.3  
-- 方法名: Player:GetGUID()  
-- 描述: 取自身GUID  
-- 返回值: 返回1个整数型返回值  
-- 示例:

```
1 local nGuid= Player:GetGUID()  
2 local nGuid2= Player:GetGUID2()  
3 local strGUID = GetLuaValue("return Player:GetGUID()", "s") -- 推  
  荐  
4 PushColorMsg(nGuid)  
5 PushColorMsg("完整的GUID:" .. nGuid2)
```

-- 编号: NO.4  
-- 方法名: Player:GetTongMengID()  
-- 描述: 取自身同盟ID  
-- 返回值: 返回1个整数型返回值  
-- 示例:

```
1 local nTmid = Player:GetTongMengID()  
2 PushColorMsg(nTmid)
```

-- 编号: NO.5  
-- 方法名: Player:GetTargetID()  
-- 描述: 取自身攻击的目标环境ID  
-- 返回值: 返回1个整数型返回值  
-- 示例:

```
1 local nMbid = Player:GetTargetID()  
2 PushColorMsg(nMbid)
```

-- 编号: NO.6  
-- 方法名: Player:GetPos()  
-- 描述: 取自身坐标  
-- 返回值: 返回1个table,有x字段和y字段  
-- 示例:

```
1 local pos = Player:GetPos()  
2 PushDebugMessage(pos.x .. " " .. pos.y)
```

-- 编号：NO.7  
-- 方法名：Player:GetState()  
-- 描述：取自身状态  
-- 返回值：返回1个整数型返回值 0:静止 2:走路 3: 轻功 5:运气 7:攻击 9:死亡 10:摆摊  
-- 示例：

```
1 local nState = Player:GetState()  
2 PushColorMsg(nState)
```

-- 编号：NO.8  
-- 方法名：Player:GetBangHuiID()  
-- 描述：取自身帮会ID  
-- 返回值：返回1个整数型返回值  
-- 示例：

```
1 local nBhid = Player:GetBangHuiID()  
2 PushColorMsg(nBhid)
```

-- 编号：NO.9  
-- 方法名：Player:GetHP()  
-- 描述：取当前血值  
-- 返回值：返回1个整数型返回值  
-- 示例：

```
1 local nHP = Player:GetHP()  
2 PushColorMsg(nHP)
```

-- 编号：NO.10  
-- 方法名：Player:GetHPP()  
-- 描述：取当前血值百分比  
-- 返回值：返回1个浮点型返回值

-- 示例：

```
1 local nHpp = Player:GetHPP()  
2 PushColorMsg(nHpp)
```

-- 编号：NO.11

-- 方法名：Player:GetMP()

-- 描述：取当前蓝值

-- 返回值：返回1个整数型返回值

-- 示例：

```
1 local nMp = Player:GetMP()  
2 PushColorMsg(nMp)
```

-- 编号：NO.12

-- 方法名：Player:GetMPP()

-- 描述：取当前蓝值百分比

-- 返回值：返回1个浮点型返回值

-- 示例：

```
1 local nMpp = Player:GetMPP()  
2 PushColorMsg(nMpp)
```

-- 编号：NO.13

-- 方法名：Player:GetRage()

-- 描述：取怒气/共情

-- 返回值：返回1个整数型返回值

-- 示例：

```
1 local nNq = Player:GetRage()  
2 PushColorMsg(nNq)
```



-- 编号: NO.14  
-- 方法名: Player:GetFightPetGUID()  
-- 描述: 取出战珍兽GUID  
-- 返回值: 返回1个整数型返回值  
-- 示例:

```
1 local nGuid = Player:GetFightPetGUID()  
2 PushColorMsg(nGuid)
```

-- 编号: NO.15  
-- 方法名: Player:GetDataInt("LEVEL")  
-- 描述: 取自己等级  
-- 返回值: 返回1个整数型返回值  
-- 示例:

```
1 local nLevel = Player:GetDataInt("LEVEL")  
2 PushColorMsg(nLevel)
```

-- 编号: NO.16  
-- 方法名: Player:GetMoney(aGetType)  
-- 描述: 取当前金币/交子  
-- 参数1: 可空nil。aGetType为2时取的是交子  
-- 返回值: 返回1个整数型返回值  
-- 示例:

```
1 local nMoney = Player:GetMoney()  
2 PushColorMsg(nMoney)
```

-- 编号: NO.17

-- 方法名: Player:GetMaxMoney(aLevel)  
-- 描述: 取当前可携带金币上限  
-- 参数1: 可空nil代表自己当前等级。不为空时获取指定等级的金币携带上限  
-- 返回值: 返回1个整数型返回值  
-- 示例:

```
1 local nMaxMoney = Player:GetMaxMoney(99)
2 PushColorMsg(nMaxMoney)
```

-- 编号: NO.18  
-- 方法名: Player:GetMenPaiID()  
-- 描述: 取门派ID  
-- 返回值: 返回1个整数型返回值  
-- 示例:

```
1 local nMpid = Player:GetMenPaiID()
2 PushColorMsg(nMpid)
```

-- 编号: NO.19  
-- 方法名: Player:GetMySeX()  
-- 描述: 取自己性别  
-- 返回值: 返回1个整数型返回值  
-- 示例:

```
1 local nSex = Player:GetMySeX()
2 PushColorMsg(nSex)
```

-- 编号: NO.20  
-- 方法名: Player:GetPKValue()  
-- 描述: 取杀气  
-- 返回值: 返回1个整数型返回值

-- 示例：

```
1 local nPk = Player:GetPKValue()  
2 PushColorMsg(nPk)
```

-- 编号：NO.21

-- 方法名：Player:YuanBaoToTicket(aCount)

-- 描述：兑换元宝票(需要在钱庄)

-- 参数1：要兑换的数量

-- 返回值：无

-- 示例：

```
1 Player:YuanBaoToTicket(100)
```

-- 编号：NO.22

-- 方法名：Player:IsWaiGong()

-- 描述：是否外攻

-- 返回值：返回1个逻辑型返回值

-- 示例：

```
1 local bWG = Player:IsWaiGong()
```

-- 编号：NO.23

-- 方法名：Player:IsJinGong()

-- 描述：是否近攻

-- 返回值：返回1个逻辑型返回值

-- 示例：

```
1 local bJG = Player:IsJinGong()
```

-- 编号：NO.24  
-- 方法名：Player:IsInTeam()  
-- 描述：是否在队伍  
-- 返回值：返回1个逻辑型返回值  
-- 示例：

```
1 local bTeam = Player:IsInTeam()
```

-- 编号：NO.25  
-- 方法名：Player:IsInRaid()  
-- 描述：是否在团队  
-- 返回值：返回1个逻辑型返回值  
-- 示例：

```
1 local bRaid = Player:IsInRaid()
```

-- 编号：NO.26  
-- 方法名：Player:IsLeader()  
-- 描述：是否队长  
-- 返回值：返回1个逻辑型返回值  
-- 示例：

```
1 local bLeader = Player:IsLeader()
```

-- 编号：NO.27  
-- 方法名：Player:IsFollow()  
-- 描述：是否跟随  
-- 返回值：返回1个逻辑型返回值  
-- 示例：

```
1 local bFollow = Player:IsFollow()
```

-- 编号: NO.28  
-- 方法名: Player:TeamFollow(true, 300)  
-- 描述: 组队跟随/取消  
-- 参数1: 逻辑性 true: 组队跟随, false: 取消跟随  
-- 参数2: 整数型, 延时毫秒  
-- 返回值: 无  
-- 示例:

```
1 Player:TeamFollow(true, 300)  
2 Player:TeamFollow(false, 300)
```

-- 编号: NO.29  
-- 方法名: Player:GetBuff("坐骑")  
-- 描述: 获取指定状态索引,无此状态返回-1  
-- 参数1: 文本型,状态名,可以是状态的部分文本  
-- 返回值: 返回1个整数型返回值  
-- 示例:

```
1 local nIdx = Player:GetBuff("坐骑")
```

-- 编号: NO.30  
-- 方法名: Player:DispelBuffByName("坐骑")  
-- 描述: 取消指定状态  
-- 参数1: 文本型,状态名,可以是状态的部分文本  
-- 返回值: 返回1个逻辑型返回值  
-- 示例:

```
1 local bDis = Player:DispelBuffByName("坐骑")
```

-- 编号：NO.31  
-- 方法名： Player:GetAttackAttr()  
-- 描述：取自己的冰火玄毒属性  
-- 返回值：返回1个table数组,数组第一个是主属性  
-- 示例：

```
1 local tAA = Player:GetAttackAttr()  
2 for i = 1, #tAA do  
3     local tmp = tAA[i]  
4     PushDebugMessage(tmp.name .. "属性：" .. tmp.attr)  
5 end
```

-- 编号：NO.32  
-- 方法名： GetMenpaiNameByID  
-- 描述：通过门派ID获取门派名称，如编号：NO.8  
-- 返回值：返回门派名称  
-- 示例：

```
1 local nMpName = GetMenpaiNameByID(Player:GetMenPaiID())  
2 PushColorMsg(nMpName)
```