# 插件安装方法:

1. 解压压缩包，查看压缩包内的文件
2. 将插件MoreLoreEffectsPlugin-1.12.2-a0.3.1.jar放入plugins
3. 将技能包loreskills放入服务端文件夹根目录
4. 打开服务器测试插件是否运行即可

# **插件的使用方法：**

## 游戏内命令

| **命令名称** | **别名** | **描述** | **使用方法** | **需要权限** |
| --- | --- | --- | --- | --- |
| checkloreskill | clsk | 查看某个Lore技能的描述 | /checkloreskill <技能名> | 否 |
| listlore | llr | 查询现版本服务器中可用的所有特殊Lore词条 | /listlore [选项] | 否 |
| listloreskill | llsk | 查询当前服务器数据中所有已设置的Lore技能 | /listloreskill [技能类型] | 是 |
| setloreskill | slsk | 在服务器数据中添加或修改Lore技能 | /setloreskill <技能名> <技能类型> <冷却> <效果> | 是 |
| delloreskill | dlsk | 删除服务器数据中设置的Lore技能 | /delloreskill <技能名> | 是 |
| reloadloreskill | rllsk | 重新读取Lore技能配置文件和lua代码 | /reloadloreskill <技能名> | 是 |
| loremanage | lrm | 简单的Lore管理 | /loremanage <操作> | 是 |
| resetlorecooldown | rlcd | Lore技能冷却时间重置 | /resetlorecooldown <玩家名> <技能名> | 是 |

****注释说明****:

* checkloreskill: 通过技能名查看技能描述。
* listlore: 查询全部可用的Lore词条
  + 可选参数all, uni, atk, dfn。参数用于筛选要查看哪些词条
  + 四种参数分别代表：全部词条、通用词条、攻击词条、防御词条
* listloreskill: 查询已设置的Lore技能，可指定技能类型(技能类型在添加技能的时候由管理员自定义)。若无参数则显示所有技能类型的Lore技能。
* setloreskill: 在服务器数据中添加或修改一项Lore技能
  + 注1：技能类型用于标记技能的分类，若技能分类相同，则同类的技能冷却共享
  + 注2：冷却为整数，单位：毫秒。若冷却为-1，则代表使用一次后永远处于冷却。直到被管理员重置，或者被其他技能重置，或者重启服务器。
  + 注3：效果项是一条要执行的指令，由服务器执行，不用输入‘/’号
  + 注4：指令中用&s来指代物品使用者名称,用&x、&y、&z来指代使用者的坐标
  + 注5：效果项中的&符号将会被转义为颜色符号，而&&连用会被转义为单个&符号
* delloreskill: 从服务器数据中移除指定技能。
* reloadloreskill: 重新从硬盘读取一个Lore技能效果的配置文件和lua代码文件，会替换技能原本的属性设置。一般由管理员在本地修改完代码文件后调用。
* loremanage: 对目前主手手持物品的Lore进行简单的管理。
  + 操作参数可为add, copy, paste, clear中的一种，效果如下:
  + add <Effect> <Value> 添加一条属性Lore到手中物品
  + copy 复制手中物品的Lore
  + paste 粘贴复制的Lore到手中物品
  + clear 清空手中物品的Lore
* resetlorecooldown: 重置玩家的Lore技能冷却时间。
  + 重置技能CD本质上是重置技能类型的CD，重置一个技能时，相同类型的技能的冷却也会一同被重置
  + 可以使用通配置符'\*'，如/rlcd \* \*命令就代表重置所有玩家的所有技能CD
  + 请谨慎使用通配符。因为冷却时间为 -1 (一次性使用，永远冷却)的技能冷却也会被重置。

## **玩家Lore词条阅读指南**

### Lore条目编写规则：

* 确保所有符号均为英文字符，并且内容中不包含空格。
* 特殊符号如[&0]（颜色代码）、%（百分号）及小括号及其内的内容[(\*\*)]在Lore展示时将被忽略。

### Lore条目大全

### **通用词条说明**

| **词条类型** | **词条示例** | **词条说明** |
| --- | --- | --- |
| ****等级限制**** | 等级限制: value | 若装备者的等级低于设定值，装备在战斗开始时会自动脱落 |
| ****技能**** | 技能: 技能名 | 手持物品按右键触发指定名称的技能，使用/slsk命令以设置技能名称对应的效果，同类技能之间共享冷却时间 |
| ****消耗: 技能**** | 消耗: 技能名 | 同“技能”，但触发结束后消耗手中物品 |
| ****攻击特效**** | - | 未实现 |

### **攻击者增益**

| **词条类型** | **词条示例** | **词条说明** |
| --- | --- | --- |
| ****注意事项**** | - | 攻击者拥有Lore词条时，禁止使用亡灵杀手与截肢杀手，这会带来无法预料的增伤 |
| ****武器攻击**** | 武器攻击: value | 作为主手近战时，无论其性质和附魔如何，将造成的基础伤害设为此数值。此词条会将跳劈伤害系数改为x1.05 |
| ****攻击力加成**** | 攻击力: +value | 作为任意装备时，增加主手近战攻击力，在百分比增益前结算 |
| ****百分比攻击**** | 百分比攻击: +value% | 作为任意装备时，增加主手近战攻击力 |
| ****吸血效果**** | 吸血: +value | 作为主手近战时，增加攻击造成的回复 |
| ****暴击率提升**** | 暴击率: +value% | 作为任意装备时，增加攻击暴击的几率 |
| ****暴击伤害加成**** | 暴击伤害: +value% | 作为任意装备时，增加攻击暴击时的增伤比，基础暴击伤害倍率为150% |
| ****远程增伤**** | - | （待实现功能） |

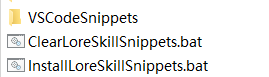
### **防守者增益**

| **词条类型** | **词条示例** | **词条说明** |
| --- | --- | --- |
| ****减伤说明**** | - | 减伤在原版护甲值后结算，减伤只能减免生物造成的伤害，无法减免：摔落，场地，药水，燃烧等异常造成的伤害 |
| ****减伤**** | 减伤: +value | 作为任意装备时，减少被生物攻击的伤害，在百分比增益前结算 |
| ****百分比减伤**** | 百分比减伤: +value% | 作为任意装备时，减少被生物攻击的伤害 |

# 自定义技能开发：

## 添加VSCode代码补全

找到解压压缩包，查看压缩包内的文件。



这三个文件是需要的

脚本文件InstallLoreSkillSnippets.bat 运行后，能为你的VScode编辑器添加代码补全

脚本文件ClearLoreSkillSnippets.bat 运行后，能为你的VScode编辑器删除之前添加的补全

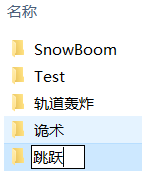
## **添加技能**

首先进入服务器根目录下的loreskills文件夹(应该是之前从压缩包里解压的技能试例)

当然，你也可以手动创建该文件夹。

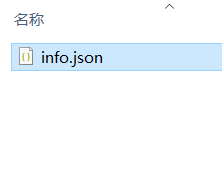


在文件夹下新建一个文件夹作为技能目录，

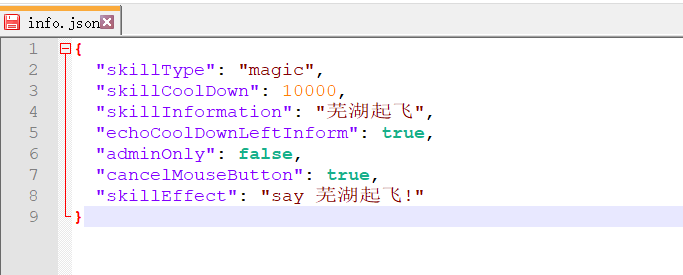


文件夹的名称很重要，它会直接作为游戏内的技能名！

创建一个名字为info.json的文件



接着填入以下内容：



{

  "skillType": "magic",

  "skillCoolDown": 10000,

  "skillInformation": "芜湖起飞!",

  "echoCoolDownLeftInform": true,

"adminOnly": false,

  "cancelMouseButton": true,

  "skillEffect": "say 芜湖起飞!"

}

然后保存退出。

你的简单自定义技能就创建完毕了。

接下来只要你启动服务器，技能会被自动加载到数据中，可以通过词条"技能: 跳跃"添加到物品上。

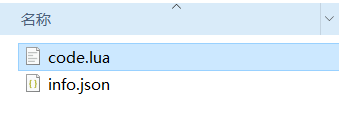
如果服务器已经在运行，又不想重新启动服务器，则管理员可以使用/reloadloreskill命令加载技能到服务器中。

当然，你可以根据你的需要定制你的技能配置文件，配置文件的说明表格如下：

|  |  |  |
| --- | --- | --- |
| 可配置项 | 类型 | 说明 |
| skillType | string | 技能类型，用于标记技能的分类，若技能分类相同，则同类的技能冷却时间共享。 |
| skillCoolDown | int | 技能的冷却时间，只能是整数，单位毫秒。若冷却为-1，则代表使用一次后永远处于冷却。直到被管理员重置，或者被其他技能重置，或者重启服务器。 |
| skillInformation | string | 技能在游戏内的描述，会影响/checkloreskill和/listloreskill命令显示的内容 |
| echoCoolDownLeftInform | bool | 如果为true，则当玩家右键物品，但技能又处于冷却时，不会显示任何提示。如果为false，则当玩家右键物品时会显示当前技能还剩多少冷却时间转好。 |
| adminOnly | bool | 如果为true，则这个技能只有管理员可以使用 |
| cancelMouseButton | bool | 如果为true，则物品取消原有的右键效果。比如方块的右键效果就是放置。开启此选项可以防止道具丢失。 |
| skillEffect | string | 填写一条游戏内的指令，但是不包括符号"/"。代表当你触发此技能时，服务器会触发的效果。  此配置项所配置的技能和用指令/setloreskill设置的技能效果相同，指令中用&s来指代物品使用者的游戏名称,用&x、&y、&z来指代使用者的坐标。&符号将会被转义为颜色符号(如&1就是颜色§1)，而&&连用会被转义为单个&符号 |

## **高级技能自定义----添加技能脚本**

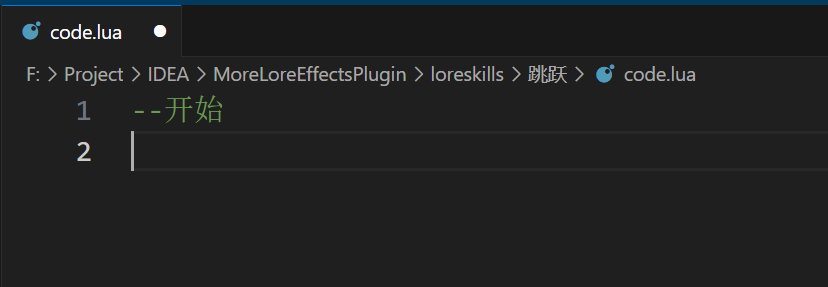
注意：如果你添加了技能脚本，则之前在info.json内添加的skillEffect项自动失效

如要使用自定义的技能脚本，请在info.json文件旁边添加名为code.lua的代码文件

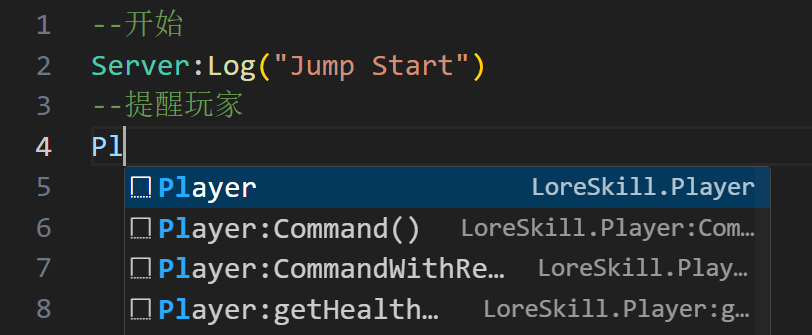
使用VScode编辑器打开此文件

此插件的技能脚本使用Lua语言，请自行了解Lua语言

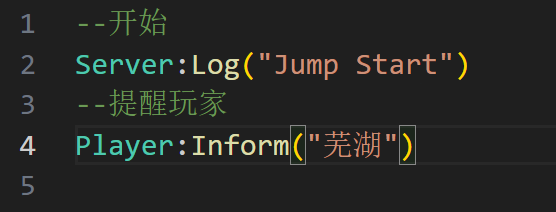
在这里我简单介绍一下写法，首先使用连续两个-符号可以添加注释



接下来自由发挥



如果你在之前的步骤中添加了代码补全，那么这里会显示有哪些函数可用。



其中：

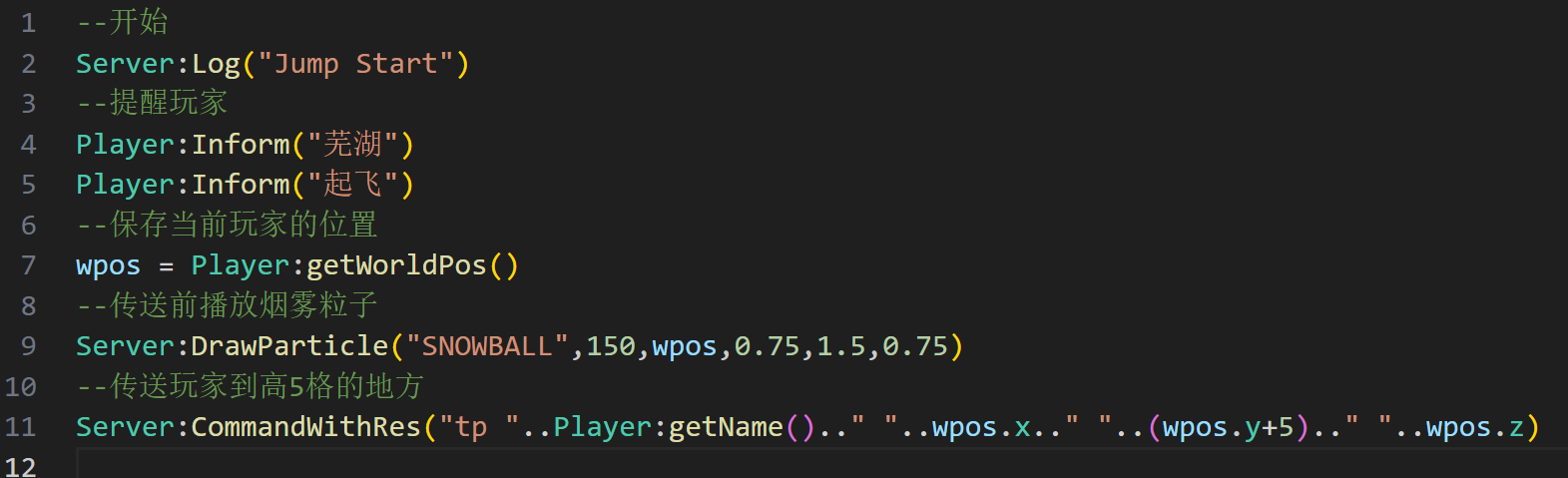
Server:Log("Jump Start")

是一个在服务器控制台上输出日志的命令，而

Player:Inform("内容")

使用这个函数可以给玩家发消息，此消息仅此玩家可见

既然是跳跃技能，接下来将移动玩家的位置



可以添加for循环使得过程更加有戏剧性



本次试例的完整代码如下：

--开始

Server:Log("Jump Start")

--提醒玩家

Player:Inform("芜湖")

--循环5次

maxTime = 5

for i=1,maxTime do

    Server:Wait(500)

    Player:Inform("起飞")

    --保存当前玩家的位置

    wpos = Player:getWorldPos()

    --传送前播放烟雾粒子

    Server:DrawParticle("SNOWBALL",150,wpos,0.75,1.5,0.75)

    --传送玩家到高5格的地方

    Server:CommandWithRes("tp "..Player:getName().." "..wpos.x.." "..(wpos.y+5).." "..wpos.z)

end

Player:Inform("飞不起来了")

--结束

Server:Log("Jump End")

更多技能的代码试例可以查看loreskills文件夹下的预置代码文件

## **LUA语句文档**

**目前提供的LUA语句大全：**

Server:Log("")

-- 在服务器控制台打出日志

Server:Wait(millisecond)

-- 等待一段时间后继续执行

Server:Command("")

-- 以服务器身份执行命令

Server:CommandWithRes("")

-- 以服务器身份执行命令，并且获取命令的返回值（该命令比普通的Command多1tick的延迟，用于获取返回值）

Server:buildWorldPos(worldPos)

-- 复制一个WorldPos对象

Server:buildWorldPos(location)

-- 用Location对象构建一个WorldPos对象

Server:buildWorldPos("worldName",x,y,z)

-- 用3维坐标构建一个WorldPos对象

Server:buildWorldPos("worldName",x,y,z,yaw,pitch)

-- 用5维坐标构建一个WorldPos对象

Server:DrawParticle(particleName,p\_num,worldPos,range\_x,range\_y,range\_z)

-- 播放粒子

Skill:getDefCd()

-- 获取当前技能在配置文件中所设置的技能cd时间，单位毫秒

Skill:SetCoolDown(millisecond)

-- 设置当前技能所在类技能的冷却时间(就是让该技能进入冷却)

Skill:SetTypeCoolDown("typeName",millisecond)

-- 使用类型名来设置某一类技能的冷却时间(就是让该技能进入冷却)

Player:Inform("")

-- 给玩家发消息

Player:Command("")

-- 以释放技能的玩家的身份执行命令

Player:CommandWithRes("")

-- 以释放技能的玩家的身份执行命令，并且获取命令的返回值（该命令比普通的Command多1tick的延迟，用于获取返回值）

Player:Teleport(worldPos)

-- 传送玩家到指定WorldPos

Player:Teleport(x,y,z)

-- 传送玩家到当前世界的指定坐标

Player:getName()

-- 返回玩家的玩家名

Player:getWorldName()

-- 返回玩家所在的世界名，可以用来构造WorldPos

Player:getX()

-- 返回玩家所在的坐标x

Player:getY()

-- 返回玩家所在的坐标y

Player:getZ()

-- 返回玩家所在的坐标z

Player:getYaw()

-- 返回玩家所在的旋转坐标yaw

Player:getPitch()

-- 返回玩家所在的旋转坐标pitch

Player:getWorldPos()

-- 返回玩家所在的WorldPos

Player:getMaxHealth()

-- 返回玩家的最大生命值

Player:getHealth()

-- 返回玩家的当前生命值

Player:setHealth(double)

-- 修改玩家的当前生命值

**附录 - DrawParticle语句中可用的粒子效果名称particleName(字符串形式)：**

    EXPLOSION\_NORMAL

    EXPLOSION\_LARGE

    EXPLOSION\_HUGE

    FIREWORKS\_SPARK

    WATER\_BUBBLE

    WATER\_SPLASH

    WATER\_WAKE

    SUSPENDED

    SUSPENDED\_DEPTH

    CRIT

    CRIT\_MAGIC

    SMOKE\_NORMAL

    SMOKE\_LARGE

    SPELL

    SPELL\_INSTANT

    SPELL\_MOB

    SPELL\_MOB\_AMBIENT

    SPELL\_WITCH

    DRIP\_WATER

    DRIP\_LAVA

    VILLAGER\_ANGRY

    VILLAGER\_HAPPY

    TOWN\_AURA

    NOTE

    PORTAL

    ENCHANTMENT\_TABLE

    FLAME

    LAVA

    FOOTSTEP

    CLOUD

    REDSTONE

    SNOWBALL

    SNOW\_SHOVEL

    SLIME

    HEART

    BARRIER

    WATER\_DROP

    ITEM\_TAKE

    MOB\_APPEARANCE

    DRAGON\_BREATH

    END\_ROD

    DAMAGE\_INDICATOR

    SWEEP\_ATTACK

    TOTEM

    SPIT