变量是个很简单的东西, 但是要小心, 因为写错了容易导致游戏崩溃. Σ(っ ° Д ° ;)っ

---

**首先是需要的简单代码集** (变量名只能用字母和_)

```
set_variable = { 变量名 = 某个数字 }                    创造变量 / 将已存在变量变成某个数字
或者
set_variable = {
        var = 变量名
        value = 某个数字
}

clear_variable = 变量名                                 删除变量 (删除不需要的可以节省性能)

add_to_variable = {变量名 = 要加上的数字}               已存在变量 + 某个数字
subtract_from_variable = {变量名 = 要减去的数字}        已存在变量 - 某个数字
multiply_variable = {变量名 = 要乘的数字}               已存在变量 * 某个数字
divide_variable = {变量名 = 要除的数字}                 已存在变量 / 某个数字
modulo_variable = {变量名 = 要模除的数字}               已存在变量 % 某个数字
或者
add_to_variable = {
        var = 变量名                                    已存在变量
        value = 某个数字或者变量                        要加上的某个数字或者变量内容
}

clamp_variable = {                                      给已存在变量设置最大/最小数
        var = 变量名                                    (比如说, 如果加减乘除.....之后太大, 就自动成最大数)
        min = 最小数
        max = 最大数
}

round_variable = 变量名                                 已存在变量四舍五入成整数
```

---

**触发器 (国策, 决策, 事件, ……)**

```
check_variable = {                                      变量比较
        var = 变量名
        value = 数字或者变量名
        compare = 比较类型                              比较类型(compare)列表在下↓
}

Compare 比较类型有以下几种
        less_than       (小于)          less_than_or_equals     (小于并等于)
        greater_than    (大于)          greater_than_or_equals  (大于并等于)
        Equals          (等于)          not_equals              (不等于)

Check 也可以用↓
check_variable = { 变量名 = 数字或者变量名 }            (等于)
check_variable = { 变量名 < 数字或者变量名 }            (小于)
check_variable = { 变量名 > 数字或者变量名 }            (大于)
(这样用的坏处是没有上面 Compare 右面的功能 [好处是很短])

has_variable = 变量名                                   检测是否存在这个变量
```

---

**游戏内控制台调试:**

```
set_var 变量名 数字     (将变量的内容改成输入的数字)
get_var 变量名          (显示变量内容)
```

**随机数字:**

randomize_temp_variable = {
	var = 变量名
	distribution = 分配算法　　有 uniform, binomial 和 poisson distribution (用第一个就好)
	min = 最小数
	max = 最大数
}

---

**下面这半页不是太重要 , 看一下就好 (毕竟越复杂, 越容易出 Bug)**

---

除了普通变量, 还有一种临时变量, 比如
set_temp_variable = { 临时变量名 = 数字 }　　创造临时变量 / 将已存在临时变量变成某个数字
临时变量在脚本结束后就会被删除所以只能暂时用来计算之后结果储存到普通变量里
(只是提一下, 各位无需使用这个)

可用临时变量代码:

| | | |
|---|---|---|
| set_temp_variable | add_to_temp_variable | subtract_from_temp_variable |
| multiply_temp_variable | divide_temp_variable | modulo_temp_variable |
| round_temp_variable | clamp_temp_variable | |

(简单来说就是在原变量代码的 variable 前加入了 temp_)

---

变量有 3 种地区类型, 来规定这是给那个地方用的代码, 比如说: 国际变量, 国家变量, 地区变量
(我们需要知道的暂时只有两种)

1. 国际变量是所有国家都有资格获取的变量, 全球事件就靠它了
	a) 直接写代码就自动创建是国际变量

例子:
```
set_variable = {
	var = 变量名
	value = 某个数字
}
```

2. 国家变量适合给国家内的国策, 事件, 决策使用 (太懒的话用国际的也行)
	a) 国家变量需要在代码前加入国家 TAG (使用 check 之类的触发器时不用谢)

例子:
```
EER {
	set_variable = {
		var = 变量名
		value = 某个数字
	}
}
```

---

**游戏变量 (Game Variables)**

游戏内含可以调取的变量, 比如说:
set_variable = { 变量名 = political_power }
(这样变量名里就储存了我们的政治点数数量, 要更多变量可以去看 WIKI 网站[1])

---

[1] https://hoi4.paradoxwikis.com/Variables#check_variable

**其他重要的触发器** (一个东西什么时候能触发, 因为很重要就写上来了)

```
if = {                                  (如果)
   limit = {                            (检测这里面的内容是否为真)
      original_tag = EER                        (比如说当前国家为 EER[平等国])
   }
   add_stability = 0.1                  (那么执行 limit 下面的代码 [增加国家稳定度 10%])
}
else_if = {                             (如果上面不为真)
   limit = {                            (检测这里面的内容是否为真)
      original_tag = EOM                        (比如说当前国家为 EOM[月之帝国])
   }
   add_stability = 0.25                 (那么执行 limit 下面的代码 [增加国家稳定度 25%])
}
else_if = {                             (如果上面两个不为真)
   limit = {                            (检测这里面的内容是否为真)
      original_tag = EQU                        (比如说当前国家为 EQU[小马国])
   }
   add_stability = 0.20                 (那么执行 limit 下面的代码 [增加国家稳定度 20%])
}
else = {                                (如果上面都不为真)
   add_stability = 0.05                 (执行这中间的代码)
}
```

需要更多触发器请看下面的地址[2] (比如说 OR 和 AND [多个里面一个为真和好几个都要为真])

要解释更多的话还要好写几篇（；´д｀）ゞ

[2] https://hoi4.paradoxwikis.com/Conditions

## 变量栗纸:

**例子的状况:** (并不是必须, 这里只是比如)
1. 我们有一个**事件文件** (events\文件夹里), 里面有两个事件
   a) 第一个 ID 是 Test_Event.1 (事件显示大于)
   b) 第二个 ID 是 Test_Event.2 (事件显示小于)
2. 我们有一个**国策文件**(common\national_focus\文件夹里), 里面有一个国策
   a) 第一个 ID 是 Focus_1
3. 我们有一个**决策&任务文件** (common\decisions\文件夹里[一个 txt 和 categories 文件夹里的 txt]), 里面有一个任务
   a) 叫 Decision_1

**代码部分:** (代码不完全, 这里只是展示如何使用变量)

**国策部分: (如何创造和改变变量)**

```
focus = {
        id = Focus_1
        completion_reward = {                                   (这里面写国策的完成效果)
                EER {                                           (说明是国家变量[只有 EER 可用])
                        set_variable = { EER_1 = 10 }           (创建一个名为 EER_1 的变量,
                                                                并且赋值[设定内容]为数字 10)
                        multiply_variable = {EER_1 = 2}         (EER_1 变量的内容乘于 2)
                        divide_variable = {EER_1 = 3}           (EER_1 变量的内容除于 3)
                        round_variable = EER_1                  (将 EER_1 变量四舍五入)
                }
        }
}
```

这个部分就是创造了 变量之后算 10 * 2 / 3 = 6.666, 在之后内容四舍五入成 7

**决策&任务文件: (如何检测变量内容[触发器])**

```
Decision_1 = {
        complete_effect = {                                     (这里面写决策&任务的完成效果)
                if = {                                          (如果)
                        limit = {                               (这里的内容为真)
                                check_variable = { from.EER_1 < 10 }   (检测 EER_1 是否小于 10)
                                                                (在检测时, 变量名前要
                                                                 加入 form.)
                        }
                        country_event = { id = Test_Event.1 }   (如果上面为真, 执行这里的代码
                                                                [触发 ID 为 Test_Event.1 的国家
                                                                 事件])
                }
                else = {                                        (如果上面不为真)
                        country_event = { id = Test_Event.1 }   (执行这中间的代码)
                                                                [触发 ID 为 Test_Event.2 的国家
                                                                 事件])
                 }
        }
}
```

国策部分将 EER_1 的变量内容变成了 7, 而这里会检测 EER_1 的变量内容是否为小于 10, 而 7 小于 10.
所以最终会触发 ID 名为 Test_Event.1 的国家事件

内容我写完没检测, 有什么错误和忘了的事的话, 请见谅.

**如果你看到这里, 而且都明白了, 那么恭喜你已入门编程, 编程书中的 1/2 的内容对你来说已经可以在几小时内理解. 如果没明白, 那么也没关系, 毕竟还是有很多内容的. 各位晚安 ( ◕ ω ◕ )✧**