

主动技 enable

例如:

```
enable: 'phaseUse', //出牌阶段发动
```

触发技 trigger

例如:

```
trigger: {  
  player: 'phaseDrawBegin', //摸牌阶段开始时  
},
```

mod技: 时刻生效的技能, 属于锁定技、但可以与主动技或触发技放在一起 mod

例如:

```
mod: {  
  maxHandcard: function(player, num){  
    return num+1; //手牌上限+1  
  },  
  cardUsable: function(card, player){  
    return Infinity; //使用牌无次数限制  
  },  
},
```

技能属性:

```
enable: 'xxx', trigger: {xxx: 'xxx'}, //主动\被动  
mod: {xxx: function,}, //mod技  
frequent: true, //自动发动  
forced: true, //锁定技(且自动触发), 与enable、frequent不兼容  
locked: true, //锁定技, 与frequent兼容  
filter: function(event, player){ //限制条件满足时发动  
  return false;  
},  
filterCard: function(card, [player]){ //牌符合条件时发动  
  //返回牌符合的条件  
},  
filterTarget: function(card, player, target){ //发动目标  
  //目标条件  
},  
usable: x, //每回合限x次。 若x为变量, 则不适用, 用filter  
round: x, //每轮限x次。 若x为变量, 则不适用, 用filter  
priority: x, //同时机技能发动优先级  
check: function(event, player){  
  //ai是否发动技能, 返回true为发动  
},  
check: function(card){}, //ai选牌的限制条件(主动技)  
check: function(button){}, //ai选按钮的限制条件(视为技, chooseButton里, 少见)
```

```

position: 'hej', //可以选择xxx位置的牌
selectCard: xxx, //需要选择xxx张牌才可以发动, -1为所有牌
viewAs: { //视为xxx牌
    name: 'xxx',
},
viewAsFilter: function(player){ //视为技能出现的条件
    //返回条件
},
prompt: 'xxxx', //选择提示语
prompt: function(event){}, //文字提示
onuse: function(result, player){ //使用视为牌时触发内容, result.cards是视为使用前的牌
    xxx
},
unique: true, //特殊技(限定技、觉醒技等)
limited: true, //限定技
mark: true, //标记
marktext: "xx", //标记名字为"xx", 可以填1个汉字或2个英文字符(因为地方不够)
intro: { //标记介绍
    name: "xxx", //标记说明的名字, 默认为技能名
    content: 'xxxx', //标记内容
    content: "card", //一张牌
    content: "cards", //多张牌
    content: "limited", //限定技, 觉醒技专用
    content: "time", //剩余发动次数
    content: "turn", //剩余回合数
    content: "cardCount", //牌数
    content: "info", //技能描述
    content: "character", //武将牌
    content: function(storage, player, skill){}, //自定义
},
skillAnimation: true, //有技能动画
init: function(player){ //初始化
    player.syncStorage("xxx"); //同步标记(每当标记变动都要写这句)
    //注: 标记名必须和技能名相同
    player.storage.xxx=false; //xxx技能未发动
},
init: function(player.skill){}, //获得技能时
onremove: function(player, skill){}, //失去技能时 是否销毁标记
logTarget: function(event, player){}, //记录技能目标
targetprompt: function(target){}, //选择目标时每个目标显示的文本
prepare: function(cards, player, targets){}, //主动技的准备工作
content: function(){}, //内容
onChooseToUse: function(event){}, //需要使用牌时的触发内容
onCompare: function(player){}, //拼点时的触发内容
chooseButton: { //选择按钮
    dialog: function(){ //选择内容
        return ui.create.dialog(['sha', 'shan'], 'vcard'); //可以选择杀和闪(卡牌)
    },
    filter: function(button, player){ //按钮选择的条件
        xxx
    },
    backup: function(links, player){ //之后发动的内容
        return {
            filterCard: function(card){}, //牌的限制条件
            viewAs: {}, //视为
            viewAsFilter: function(player){}, //视为技的限制条件
        }
    }
}

```

```

        onuse: function(result, player){}, //视为技为使用时的触发内容
        precontent: function() {}, //技能发动前的内容
    };
},
prompt: function(links, player){ //选择时弹出的提示
    return 'xx';
}
},

```

常见trigger

```

//任何时机都能通过添加"Before","Begin","End","After"获得不同时间点的新时机
//任何事件函数都有时机(例如:draw,recover,chooseToDiscard,changeHp)
//任何牌都有时机(例如:taoAfter,"juedouAfter","huogongAfter")
//注1:init其实是global:"gameStart"和player:"enterGame",时机
//注2:没有changeHujia这个时机,但有changeHujiaBegin等时机(因为没trigger)
//注3:changeHp没有changeHpBegin等时机(因为false了)
global:"gameDrawAfter",//游戏发牌后
global:"roundStart",//一轮开始时
player:"phaseBegin",//回合开始时(已被"phaseZhunbeiBegin"取代)
player:"phaseEnd",//回合结束时(已被"phaseJieshuBegin"取代)
player:"phaseJudgeEnd",//判定阶段结束时
player:"phaseDrawEnd",//摸牌阶段结束时
player:"phaseUseEnd",//出牌阶段结束时
player:"phaseDiscardEnd",//弃牌判定阶段结束时
player:"damageEnd",//受到伤害结束时
source:"damageEnd",//造成伤害结束时
player:"shaBegin",//你使用杀结算开始时
target:"shaBegin",//你成为杀的目标结算开始时
player:"dyingBegin",//你的濒死阶段开始时
player:"dieBegin",//确定死亡开始时
player:"useCardEnd",//使用牌结束时
player:"useCardToEnd",//你使用牌对目标结算结束时
player:"chooseToRespondBegin",//需要响应牌开始时
player:"chooseToUseBegin",//需要使用牌开始时
player:"chooseToCompareAfter",//拼点后
player:"judge",//判定时
player:"chooseCard",//选牌时
player:"respond",//响应时
player:"loseEnd",//失去牌结束时
player:"gainEnd",//获得牌结束时
player:"linkAfter",//连环后
player:"turnOverAfter",//翻面后

```

阶段:(可搭配时机)

pahse	//回合
phaseZhunber	//准备阶段
phaseJudge	//判定阶段
phaseDraw	//摸牌阶段
phaseUse	//出牌阶段
phaseDiscard	//弃牌阶段
phaseJieshu	//结束阶段

卡牌相关:(可搭配时机)

```
sha      //使用杀
juedou   //使用决斗
lose     //失去牌
gain     //获得牌
useCard  //使用牌
useCardTo //使用牌指定目标
respond  //打出牌
draw     //摸牌
equip    //装备装备牌
```

体力相关:(可搭配时机)

```
damage    //受到伤害
loseHp    //流失体力
recover   //回复体力
changeHp   //体力改变后(不可以搭配时机)
loseMaxHp  //减少体力上限
gainMaxHp  //增加体力上限
```

状态类:(可搭配时机)

```
chooseToRespond //打出牌
turnOver       //翻面
link           //横置
dying          //濒死
```

时机:

```
before //之前
begin  //开始时
end    //结束时
after  //结束后

miss //未命中(搭配卡牌)
```

```
player //代表当前角色
source //代表事件来源
target //代表目标角色
global //代表任意角色
```

```
player.draw([num], ['bottom']); //角色摸num张牌(默认为1)，默认牌堆顶
"第一个参数为数量，默认为1"
player.draw(2); //摸两张牌
"/          一些少见的参数          /"
player.draw(target); //target令你摸一张牌
player.draw().log=false; //无记录
```

```

player.draw("bottom");//从牌堆底摸一张牌
player.draw("nodeLAY");//没有摸牌动画
player.draw("visible");//摸一张牌并展示
"结果result"
result    //摸到的牌，为数组，result[0]是摸到的第一张牌
"简单的例子"
content:function(){//内容:
    "step 0"//第0步(必须从0开始)
    player.draw("visible");//摸一张牌并展示
    "step 1"//第1步
    event.card=result[0];//用event.card记录摸到的第一张牌
    "step 2"//第2步
    if(event.card.name=="sha"){//如果摸到的牌是"杀"
        player.recover();//你回复一点体力
    }
},
"draw事件"
event.player    //摸牌的人
event.num    //摸牌数
event.result    //摸到的牌
event.source    //摸牌来源角色
"另外讲一下phaseDraw事件，注意和draw事件的区别"
event.player    //进行摸牌阶段的人
event.num    //摸牌数
event.cards    //摸到的牌
event.numFixed    //放弃摸牌
event.attachDraw    //要额外摸到的牌
//注:draw是gain的一种，gain获得的牌是event.cards

addMark(技能名,数量,log);//添加标记(只适用于标记是数字);
addSkill('xxx') //获得技能xxx
addTempSkill('xxx', 'ttt') //获得技能xxx直到ttt时刻
awakenSkill('xxx') //xxx必须是限定技，xxx不可再次使用。需要放在
player.storage.xxx=true;后
changeGroup('xxx') //切换国籍为xxx
changeHujia([xxx]) //角色获得xxx点护甲，默认为1，可填负数

//函数介绍chooseCard()
player.chooseCard(数量,位置,提示,强制,限制,ai);
//第一个参数为数量，默认为1"
player.chooseCard(2);//选两张手牌
//第二个参数为位置，默认为手牌
player.chooseCard("e");//选一张装备区的牌
//注:判定区的牌点不到，请用choosePlayerCard()或chooseButton()
//第三个参数为提示，默认为请选择x张牌
player.chooseCard("请选择一张手牌");//提示为:请选择一张手牌
//第四个参数为强制，默认为false
player.chooseCard(true);//必须选一张手牌
//第五个参数为限制，默认为无
player.chooseCard({color:"red"});//选择一张红色手牌
player.chooseCard(function(card){//注:新版本添加了参数player
    return card.name=="sha";//选择手牌中的一张杀
});
//第六个参数为ai，默认为无,最好用set
player.chooseCard().set("ai",function(card){
    return get.value(card);//选择价值高的牌
});
//结果result

```

```

result.bool    //是否选择了牌。如果点了取消，就是false
result.cards   //选择的牌，为数组
//简单的例子
content:function(){//内容：
    "step 0"//第0步(必须从0开始)
    player.chooseCard("he").set("ai",function(card){//选择一张牌
        if(card.name=="sha")return 1;//如果牌名字是杀，返回1(大于0代表选)
        else return 0;//否则返回0(即不选)
    });
    "step 1"//第1步
    if(result.bool){//如果上一步没取消
        if(result.cards[0].name=="sha"){//如果选的(第一张)牌名字是"杀"
            player.draw();//你摸一张牌
        }
    }
},

chooseDrawRecover() //摸牌或回血

chooseTarget(function(card,player,target){//请选择一个目标
    return player!=target;//限制条件:不能选你
}).set("ai",function(target){//ai选目标的限制条件
    var player=get.player();//定义变量player为选目标的发起者(不懂可以先不写)
    return get.attitude(player,target);//选友军
});

chooseToDiscard([x],[true])    //弃置x张牌。默认为1 可以取消。true不可取消
clearSkills()    //清空所有技能
countCards([位置],[条件])    //获取牌数
countMark(技能名);//有多少该技能的标记(若标记为数组，返回数组长度)
delete()    //角色神圣死亡

player.damage(数量,属性,来源,牌);
"第一个参数为数量，默认为1"
player.damage(1);//受到1点伤害
"第二个参数为属性，默认为无"
player.damage("fire");//受到1点火焰伤害 雷:thunder
"第三个参数为来源，默认为当前事件的角色"
target.damage(player);//target受到你造成的1点伤害
"第四个参数为牌，默认为当前事件弃置的牌"
player.damage({name:"sha"});//受到杀造成的1点伤害
"/          一些少见的参数          /"
player.damage("nosource");//受到1点无来源伤害
player.damage("nocard");//受到1点非牌造成的伤害
player.damage("nottrigger");//受到一点神圣伤害
/*注：
1. 你的技能里面，你为伤害来源
2. 主动技里面，牌为主动技弃置的牌(例如:强袭、黠武弃置的牌，所以要加"nocard")*/
"另外讲一下damage事件"
event.player    //受到伤害的角色
event.source    //伤害来源
event.num    //伤害大小
event.nature    //伤害的属性
event.card    //造成伤害的虚拟牌
event.cards    //造成伤害的实体牌
//可以看到，函数和事件是一一对应的
"时机"
player:"damageBefore",//受到伤害前，免疫伤害的技能

```

```

player:"damageBegin",//受到伤害时,旧时机就不多讲了
player:"damageZero",//受到0点伤害时,触发这个就不触发别的
source:"damageBegin1",//造成伤害时1,可以改变伤害大小,例如裸衣
source:"damageBegin2",//造成伤害时2,不能改变伤害大小,例如寒冰剑
player:"damageBegin3",//受到伤害时1,可以改变大小,可以转移(标天香)
player:"damageBegin4",//受到伤害时2,不能改变大小,可以取消(界天香)
player:"damage",//受到伤害,此时血量已经减少了
source:"damageSource",//造成伤害后,此时目标的血量已经减少了
player:"damageEnd",//受到伤害后,此时血量已经减少了,卖血技的时机

die()    //角色立刻死亡
disableEquip(x/'equip2')    //废除x栏
disableJudge()    //废除判定区(不可恢复)
discard([x])    //弃置x张牌,默认为1.
enableEquip(x/'equip4')    //恢复x栏
equip(x) //装备栏 1武器 2防具 3加一马 4减一马 5宝物
gain([x], [true])    //获得x张牌,默认为1 可以 取消。true为必须获得
gainMaxHp([x])    //体力上限+x,默认为1
getDamagedHp()    //获取已损体力值
player.getDamagedHp.apply(target);    //每个函数有apply()
goMad() //角色陷入混乱
hasMark(技能名);//是否有该技能的标记
init('xxx') //将角色替换为xxx武将
isDamaged()    //是否受伤
isDisable(x/'equip3')    //x栏是否已废除
isEmpty()    //装备区未被废除且无装备
link([false])    //默认横置, false为解除横置
lose(['xxx','xxx'], ui.special, 'toStorage')    //将xxx、xxx移出游戏, ui.special代表
游戏外, toStorage标记触发周妃技能
loseMaxHp([x])    //体力上限-x,默认为1
loseHp([x]) //流失x点体力,默认为1
markSkill(技能名);//显示该技能的标记
recover([x])    //回复x点体力
remove()    //角色移出游戏(不可恢复)
removeMark(技能名,数量,log);//移除标记(只适用于标记是数字);
removeSkill('xxx')    //失去技能xxx
revive()    //角色复活
skip('ttt') //角色跳过ttt阶段
syncStorage(技能名);//同步该技能标记
trunOver([false])    //默认翻面, false为解除翻面
unmarkSkill(技能名);//不显示该技能的标记
updateMarks();//同步所有技能的标记

```

位置: 'h'手牌、'e'装备区、'j'判定区、's'木牛流马内、'hs'手牌和木牛流马内、'he'你的牌、'hej'你区域里的牌、'ej'你场上的牌

条件

条件: 牌名(如: 'sha'杀)、多个牌名(用数组)、牌对象(对象属性有 颜色'color'、花色'suit'、牌名'name'、点数'number')、函数

花色: 红桃'heart'、方片''、梅花''、黑桃'diamond'、

content:function({}),的分步写法

如果技能处理的两步可以同时处理,可并为一歩,否则需要分步(例如:回血摸牌可以并为一歩,选目标并弃置目标手牌就不能并为一歩)

```

content:function(){//内容:
    "step 0"//必须从0开始, 引号可以是单引号, 但是整个技能里面不能变
    player.chooseTarget();//选目标
    "step 1"//第1步
    if(result.bool){//如果上一步没取消
        result.targets[0].draw();//目标摸一张牌
        if(result.targets[0].countCards("h")>3){//如果目标手牌数大于3
            event.finish();//这一步结束后, 整个事件结束
        }
        if(result.targets[0].countCards("h")<2){//如果目标手牌数小于2
            event.goto(0);//这一步结束后, 返回第0步
        }
    }
    "step 2"//第2步
    player.draw();//你摸一张牌
    if(player.countCards("h")<2){//如果你的手牌数小于2
        event.redo();//这一步结束后, 再进行这一步
    }
},

```

get

```

get.type()      //获取牌的类型
get.type2()     //区别是否为延时锦囊
get.attitude()  //获取态度, 大于0代表右方, 小于0代表敌方
get.distance()  //获取距离
get.effect()    //获取效果, 大于0代表有益
get.value()     //获取价值, 大于0代表有用
get.position()  //获取牌的位置
get.name()      //获取牌名
get.suit()      //获取牌花色
get.color()     //获取牌颜色
get.nature()    //获取属性
get.subtype()   //获取子类别
get.bottomCards() //获取牌堆底的牌
get.cards()     //获取牌堆顶的牌

```

AI框架

```

ai:{
    save: true, //此技能可以用于自救
    respondTao: true, //此技能可以用于救人, 一般用于视为技
    respondShan: true, //此技能可以响应杀, 一般用于视为技。作用是告诉AI手里没『闪』也可能
    出『闪』,防止没『闪』直接掉血;
    respondSha: true, //此技能可以响应闪, 一般用于视为技
    order: 8, //ai发动技能的优先度 10~1
        /*比什么先:
            order:function(){//例子, 比打出杀的优先度高1
                return get.order({name:'sha'})+1;
            },
        */
    expose: 0.2, //发动技能是身份暴露度 (0~1, 相当于概率)
    maixie: true, //卖血技
    threaten: 0.5, //嘲讽值0~4

```



```

    noh: true, //技能标签『无牌』,目前只出现在“连营”和“伤逝”中,用于其它AI检测是否含有标签『无牌』,从而告诉其他AI不要拆迁(因为生生不息)。
    result: { //主动技专属
        target: function(player, target) { //ai如何选择目标
            //返回负, 选敌人, 返回正, 选队友
            //没有返回值则不选
            //写了这个就不用写player: function(player) {}了
            //因为player可以在这里进行判断
            if(player.hp > 2) {
                return -1; //例子, 玩家体力大于2时, 发动技能并选择敌人
            }
        },
        player: function(player) { //ai是否发动此技能, 返回正, 发动, 否则不发动
            if(player.countCards('h') < 2) return 1; //例子, 玩家手牌数小于2时, 发动
        },
    },
    effect: { //影响ai出牌(例如什么时候不出杀)等
        player: function(card, player, target) {}, //牌对你的影响
        target: function(card, player, target) {}, //一名角色以你为牌的目标时对你的影响
    },
    skillTagFilter: function(player) { //视为技专属, ai什么时候可以发动视为技
        player.countCards('h', {type: 'basic'}) > 0; //例子, 手牌中有基本牌时, ai发动技能
    },
},

```

AI获取角色:

```

ai.get.attitude(玩家1, 玩家2) //『态度值检测』检测玩家1对玩家2的态度值;
ai.get.effect(target, content, player, viewer) //『效果检测』检测卡牌/技能对目标角色的效果值, content代表卡牌或者技能, 例如{name: 'sha'}/'挑衅'; viewer代表视角, 一般填player或者target, 例如我方杀敌方满血『曹丕』, 对我方来说是负效果, 但对敌方是正效果, 视角决定效果的正负。
ai.get.damageEffect(玩家1, 玩家2, viewer, nature) //『检测伤害效果』这里检测的是[玩家2对玩家1]的伤害效果, nature代表属性(雷/火);
ai.get.recoverEffect(玩家1, 玩家2, viewer) //『检测回复效果』与上段类似;

```

AI获取:

```

useful() //回合外留牌的价值(弃牌阶段按useful顺序选)
value() //牌的使用价值(五谷按value顺序选)

//举例:
ai.get.useful(card)/ai.get.value(card)

```

全局变量

1. event.xx(只在此技能内有效)

```

content:function(){//内容:
    "step 0"//必须从0开始, 引号可以是单引号, 但是整个技能里面不能变
    player.chooseTarget(true);//必须选一个目标
    "step 1"//第1步
    event.targets=result.targets;//用event.targets记录选的目标
    "step 2"//第2步
    event.targets[0].draw(2);//(选的第1个)目标摸2张牌
    "step 3"//第3步
    event.targets[0].damage();//(选的第1个)目标受到一点伤害
},

```

2. 标记, player.storage.xx(整局游戏有效)"

```

content:function(){//内容:
    "step 0"//必须从0开始, 引号可以是单引号, 但是整个技能里面不能变
    player.storage.num=player.hp;//用player.storage.num记录你的体力
    "step 1"//第1步
    player.draw(player.storage.num);//你摸你体力值张牌
},

```

3. 不用var 定义的变量是全局变量(只要不重启, 一直有效)"

```

//注意, 这种全局变量最好不要出现在技能里面
content:function(){//内容:
    "step 0"//必须从0开始, 引号可以是单引号, 但是整个技能里面不能变
    cards=player.getCards("h");//用全局变量cards记录你的手牌
    "step 1"//第1步
    player.discard(cards);//你弃置所有手牌
},

```

```

"//判定"
player.judge(名字, 函数);
"第1个参数为判定时显示的文字, 默认为技能名"
event.judgestr="闪电";//用event.judgestr记录"闪电"
player.judge(event.judgestr);//你进行名字为闪电的判定
//注意, event.judgestr是judge()里面的变量, 不能用其他变量记录
"第2个参数为判定时判断结果的函数"
"返回值大于0时, 你头像上出现“洗具”, 返回值小于0时, 你头像上出现“杯具”。"
//注意: 这个参数是用来教ai改判的
content:function(){//内容:
    "step 0"//必须从0开始, 引号可以是单引号, 但是整个技能里面不能变
    player.judge(function(card){//你进行一次判定
        return card.color=="red"?1:-1;//红色返回1, 否则返回-1
    });
},
"判定的result比较多, 还以上面的为例, 假如判定牌是黑桃6青釭剑"
result.card=="qinggang";//result.card为黑桃6青釭剑
result.suit=="spade";//result.suit为黑桃
result.color=="black";//result.color为黑色
result.number==6;//result.number为6
result.judge==-1;//result.judge为-1
result.bool==false;//result.bool为false
/*注意:
当result.judge大于0时, result.bool为true

```

```

当result.judge等于0时，result.bool为null*/
"如果判定的结果很多(例如每种花色一个不同效果)，可以用switch(){}"
content:function(){//内容：
    "step 0"//必须从0开始，引号可以是单引号，但是整个技能里面不能变
    player.judge();//你进行一次判定
    "step 1"//第1步
    switch(result.suit){//根据判定牌的花色
        case "spade":player.loseHp();break;//黑桃:你失去一点体力
        case "heart":player.draw();break;//红桃:你摸一张牌
        case "club":player.damage();break;//梅花:你受到一点伤害
        default:player.insertPhase();//其他情况:你进行一个额外回合
    }
},

```

```

"//选目标"
"这里只介绍触发技的选目标写法，主动技的请看主动技框架"
player.chooseTarget(提示,数量,强制,条件,ai);
"第1个参数为选目标时的提示，默认为“请选择#个目标”"
player.chooseTarget("请选择一个目标");//选择1个目标，提示为“请选择一个目标”
"第2个参数为选目标的个数，默认为1"
player.chooseTarget(2);//选2个目标
player.chooseTarget([1,2]);//选1到2个目标
player.chooseTarget([1,Infinity]);//选1个以上目标
"第3个参数为强制，默认为false"
player.chooseTarget(true);//必须选1个目标
"第4个参数为限制条件，默认没有条件"
player.chooseTarget(function(card,player,target){//选1个目标
    return player!=target;//限制条件:你不是目标
});
"第5个参数为ai，默认第2个函数参数是ai，如果怕混淆，用set()"
player.chooseTarget(function(card,player,target){//选1个目标
    return player!=target;//限制条件:你不是目标
},function(target){//ai:
    var player=get.player();//定义变量player为选目标的发起者(不懂可以先不写)
    return -get.attitude(player,target);//选敌人
});
player.chooseTarget().set("ai",function(target){//设置ai:
    return get.attitude(player,target);//选友军
});
"选目标的result"
result.bool//是否选了目标，选了为true，没选为false
result.targets//选择的目标，为数组，result.targets[0]为选的第1个目标
"还是拿之前的作为例子"
content:function(){//内容：
    "step 0"//第0步(必须从0开始)
    player.chooseTarget();//你可以选择一个目标
    "step 1"//第1步
    if(result.bool){//如果选了目标(没取消)
        result.targets[0].draw();//(选的第一个)目标摸一张牌
    }
    else player.draw();//否则，你摸一张牌
},

```

```

"拼点"
player.chooseToCompare(目标);
"首先得有能拼点的目标"
filter:function(event,player){//发动限制条件
    return game.hasPlayer(function(current){//场上有角色
        return player.canCompare(current);//你能和他拼点
    });
},
"拼点的result"
//假如player的拼点牌为青釭剑，目标的为古锭刀
result.bool==true;//6>1
result.tie==false;//不是平局
result.player    //你的拼点牌
result.target    //目标的拼点牌
result.num1==6    //你的点数为6
result.num2==1    //目标的点数为1
"改变拼点用的牌"
onCompare:function(player){//拼点时拼点牌为
    return game.cardsGotoSpecial(get.cards()).cards;//牌堆顶一张牌
},//摘自秦宓的"天辩"
"其他少见的操作"
player.chooseToCompareMultiple(目标数组);//同时和多人拼点
//player.chooseToCompareMultiple的目标拼点牌为result.targets
player.chooseToCompare(targets).callback=lib.skill.gushe.callback;
//摘自王朗的"鼓舌"
callback:function(){
    "step 0"
    //和content的写法一样，event.num1为player的点数，event.num2为目标的点数
},
ai:{
    noCompareTarget:true,//不能成为拼点目标(技能标签，告诉ai的)
},

```

技能组与子技能

```

group:["quanji","zili"],//拥有技能"权计"，"自立"
//注:发动技能时显示该技能名(权计，自立)
group:["longdan_sha","longdan_shan"],//拥有"龙胆"的2个子技能
//发动技能时显示"龙胆"
/*特别注意:不能group一个未定义技能，或含有group的技能*/
"假设技能名为jn"
subSkill:{//子技能
    sha:{//子技能名字为"sha"
        sub:true,//是子技能
    },
    "1":{//子技能名字为"1"
        sub:true,//是子技能
        content:function(){//内容:
            player.draw();//你摸一张牌
        },
    },
},
enable:"phaseUse",//出牌阶段发动
content:function(){//内容:
    player.addTempSkill("jn_sha");//你获得子技能"杀"直至该回合结束
}

```

```
player.useSkill("jn_1");//你发动子技能"1"
},
/*注：
1.sub:true,如果不写会自动补上
2.自己group的子技能的mark:true,不会显示标记，可以用markSkill显示
3.子技能和主技能共用技能名，但不共用技能描述
4.多个子技能可以有不同配音
5.如果主技能不是锁定技，子技能会被界马超封，即使子技能是锁定技
*/
```
