

是原载于MCbbs论坛上的命令教程。随着论坛的消失，这系列教程经修缮（可能需要几天时间）后被迁移至此。

本文某种程度上是一个白话版的Wiki

作者不保证对本教程的维护和更新（全随兴致），请您自行判断内容是否过时。不过，若有更多问题欢迎在评论区提出

教程声明：

1. 系列教程默认读者是按照顺序阅读的
2. 教程默认读者有对Minecraft的基础了解
3. 教程主要注重基础原理，实践内容极少
4. 本篇教程适用于Java 1.21，但大部分内容也适用于1.16+

前言 在前几篇说过，玩家的NBT是不能被直接修改的。这导致了，曾经，直接更改玩家的属性——比如攻击力或生命值是一种无法实现的东西。曾经，我们想要更改玩家的属性，只能为其加上状态效果，或是装备。也衍生出了很多可以影响玩家攻击力的方法。但是，从1.16，一个新的命令被加入到Minecraft，改变了这种局面。我们终于可随意地更改任意实体的属性了！这包括攻击力，移动速度，生命值护甲值击退抗性等等等等.....还有一些奇怪的东西，比如跳跃高度，方块交互距离等 这个命令，叫/attribute

1. 属性

玩家的能力有多大？你的**生命值上限**有20点。基础**攻击力**只有1点。而基础**护甲值**，**盔甲韧性**，**击退抗性**抵消的击退的比率都是零。是什么决定了你的各种能力的大小？是**属性（Attributes）**。玩家——或任何生物，都拥有属性。这些属性决定了实体在游戏中的一切能力。但属性并不是永远固定的。玩家的**基础攻击力**只有1点，但是通过持有武器，我们的攻击力可以被**修改**。也就是说，

- 最终的属性由两部分决定：**基础数值**（****基**值）**，与**加成数值**（****属**性修饰符）**

这就是属性的基础机制。每个属性都有一个**属性ID**以作区分。比如，**生命值上限**的属性ID是 generic.max_health。下表列出了Minecraft中存在的所有属性以及ID：

属性名称	属性ID	数值范围	适用版本	描述
- 以下属性 所有生物 均拥有				
攻击伤害	generic.attack_damage	0 ~ 2048	1.16+	生物的近战攻击伤害。玩家默认为1
最大生命值	generic.max_health	1 ~ 1024	1.16+	生物能恢复的生命值上限。玩家默认为20点
护甲值	generic.armor	0 ~ 30	1.16+	对物理伤害的抵抗性。每一点可抵挡4%的物理伤害。玩家默认为0
盔甲韧性	generic.armor_toughness	0 ~ 20	1.16+	当实体受的伤害变高时，盔甲所能减免的伤害比例会逐渐降低，盔甲韧性可以减缓它降低的幅度。该属性玩家默认为0
速度	generic.movement_speed	0 ~ 1024	1.16+	生物的地面移动速度。数值越高越快。玩家的默认移速为0.1。你可以使用公式 $43.178 \times 属性值 - 0.022$ 来计算大致的，以格/米作单位的速度
击退抗性	generic.knockback_resistance	-1 ~ 1	1.16+	生物对战斗击退的抵抗强度。0为无抵抗，1为完全抵抗。0~1之间的数值代表减免的百分比。玩家默认为0。（物品栏（如下界合金装备）的属性介绍里该数值会乘以十）
爆炸击退抗性	generic.explosion_knockback_resistance	-1 ~ 1	1.21+	生物对爆炸击退的抵抗强度。其他同击退抗性
最大伤害吸收值	generic.max_absorption	0 ~ 2048	1.20.2+	生物的最大伤害吸收值。默认为0。吃下金苹果会临时提升本数值，并给予伤害吸收
摔落伤害倍数	generic.fall_damage_multiplier	0 ~ 100	1.20.5+	生物受到的摔落伤害的乘数。包括玩家的大部分生物默认为1
重力	generic.gravity	-1 ~ 1	1.20.5+	生物在垂直方向上持续受到的向下加速的加速度，可以为正负1之间。所有生物默认0.08
最大行走高度	generic.step_height	0 ~ 10	1.20.5+	生物不需要跳跃就可以走上的最大高度。玩家默认为0.6
安全摔落高度	generic.safe_fall_distance	-1024 ~ 1024	1.20.5+	生物不受摔落伤害的高度。玩家默认为3
尺寸	generic.scale	0.0625 ~ 16	1.20.5+	生物的尺寸乘数。生物模型和碰撞箱的边长会乘以本属性。所有生物默认为1
额外氧气	generic.oxygen_bonus	0 ~ 1024	1.21+	生物抵抗氧气下降的能力。数值高于0时氧气值应当下降时有几率不下降。所有生物默认为0
击退	generic.attack_knockback	0 ~ 5	1.21+	生物的攻击造成的击退强度。每一点都会增加击退力度。玩家默认为0（无加成）
燃烧时间	generic.burning_time	0 ~ 1024	1.21+	生物着火时效果的乘数。所有生物默认为1
跳跃力度	generic.jump_strength	0 ~ 32	1.21+	生物进行跳跃时获得的垂直速度。玩家默认0.42
移动速率	generic.movement_efficiency	0 ~ 1	1.21+	生物对脚下方块影响移动效果的抵抗（如在灵魂沙上移速降低）所有生物默认为0。若为1则不会造成影响
水下移动速率	generic.water_movement_efficiency	0 ~ 1	1.21+	生物对水影响移动效果的抵抗。为1时不受影响。所有生物默认为0
- 以下属性 只有玩家 拥有				
攻击速度	generic.attack_speed	0 ~ 1024	1.16+	本数值代表了玩家每秒可以进行的全力攻击次数，默认为4
幸运值	generic.luck	-1024 ~ 1024	1.16+	高幸运值使玩家获得更好的战利品，默认为0
方块交互距离	player.block_interaction_range	0 ~ 64	1.20.5+	玩家可以与方块交互的距离（挖掘，交互，放置等），默认为4.5
实体交互距离	player.entity_interaction_range	0 ~ 64	1.20.5+	玩家可以与实体交互的距离（攻击，交互等），默认为3。这是玩家的眼部位置到指定实体的碰撞箱底面中心点的距离
挖掘效率	player.mining_efficiency	0 ~ 1024	1.21+	默认为0。大于0时增加玩家的挖掘速度
方块破坏速度	player.block_break_speed	0 ~ 1024	1.21+	玩家破坏方块的速度乘数。默认为1
潜行速度	player.sneaking_speed	0 ~ 1	1.21+	玩家的潜行时的速度乘数。1为和行走时同速。默认为0.3
水下挖掘速度	player.submerged_mining_speed	0 ~ 1	1.21+	玩家浸没在水中时的挖掘速度乘数。1为和地面上同速，默认为0.2
横扫伤害比率	player.sweeping_damage_ratio	0 ~ 1	1.21+	决定玩家的横扫伤害。玩家的横扫伤害等于正常伤害乘以本属性后再加1。默认为0
- 以下属性 只有特定生物 拥有				
僵尸增援	zombie.spawn_reinforcements	0 ~ 1	<div><div></div></div> 1.16+	僵尸类怪物生成僵尸增援的能力。详见Wiki: 属性/僵尸增援
生物跟随距离	generic.follow_range	0 ~ 2048	1.16+	生物AI可追踪和寻路的最大范围（如怪物追逐玩家，猪追逐萝卜等）。根据不同生物默认在12~100不等。该属性除了玩家和盔甲架以外的生物均拥有本属性
飞行速度	generic.flying_speed	0 ~ 1024	1.16+	生物的飞行移动的速度。只有蜜蜂，洞灵，鹦鹉，和悦灵拥有本属性
- 以下属性在 最新版中 已被 移除或替换				
马跳跃高度	horse.jump_strength	0 ~ 32	至 1.20.4	只对马生效。上方的跳跃力度属性在1.20.5以后替代了本属性，对所有生物生效。

部分内容取自Minecraft Wiki

上表可能已经过时并不符最新版本情况。请以[本页](#)面为准。最后更新：2024年7月，版本1.21。

下面让我们来进入正题：1.16新增的命令/attribute可以获取，或修改指定实体的属性。该命令拥有3个子命令：用于获取属性值的get，用于修改属性基值的set，与用于修改属性加成的modifier。首先让我们来看向，读取子命令get。

2. 获取属性

注意：本命令为Java版独有

命令/attribute的子命令get可以读取指定实体特定属性的基值或最终值。格式：

```
/attribute <目标> <属性ID> [base] get [<倍率>]
```

在该命令中，<目标>应为一个只能选择一个实体（limit=1）的选择器（和/data一样，本命令一次只能对一个实体操作）。中间的[base]是一个选项。**填入**就代表获取该属性的**基值**，**缺省**则意味获取该属性的**最终数值**（经装备，状态效果等加成后）。最后，可以通过[<倍率>]来指定返回的倍率。【注1】比如，上表中可以看到，攻击力的属性ID是generic.attack_damage。现在假设我拿着一把钻石剑（提供攻击力+6），那么该如何得到我当前的攻击力呢？

```
/attribute @s generic.attack_damage get
```

返回自己的**最终**攻击力 此时游戏会返回“7”。因为玩家有1点基础攻击力，钻石剑+6，所以最后为7。注意我们缺省了“base”，所以游戏返回的是我们的**最终数值**（经过钻石剑加成后的）。但若是填入“base”：

```
/attribute @s generic.attack_damage base get
```

返回自己的**基础**攻击力 此时游戏不再返回7，而回返回玩家的基础攻击力1。因为本命令获取的是**基值**，即**忽略武器装备的加成**。好，更多的例子：**例1** 假设，一名普通的玩家手持钻石剑（也只手持钻石剑）。获取该玩家目前的攻速：

```
/attribute @p generic.attack_speed get
```

返回：1.6 这代表了，手持钻石剑时，玩家的攻速为1.6（每秒可全力攻击1.6次） 获取玩家的基础攻速：

```
/attribute @p generic.attack_speed base get
```

返回：4 这代表玩家的攻速基值为4（每秒全力攻击4次，是的，手持武器会降低你的攻速） **例2** 获得最近僵尸的追踪距离：

```
/attribute @e[limit=1,type=zombie,sort=nearest] minecraft:generic.foll
```



例3 假设一普通玩家穿着全套钻石装： 获取护甲最终值（加成后）：

```
/attribute @p generic.armor get
```

返回：20（全套钻装提供满护甲值） 获取基值：

```
/attribute @p generic.armor base get
```

返回：0（玩家的装默认护甲值为0，不算装备） 好，例子举到这里。看完了读取.....现在就进入到非常令人激动的部分：修改吧。命令/attribute有两种不同的“修改”。一种是修改基础数值，一种是修改“加成数值”（相当于给你加一个虚拟的武器）。让我们先进入到第一种，修改基值。3. **修改基值** 玩家的基础攻击力也太低了吧.....万一生存第一天就遇到僵尸，使用低的可怜的1点攻击力，你得敲它20多【注2】次才能把它敲死。没关系，这里我们就来修改一下玩家的基础属性： 格式：

```
/attribute <目标> <属性ID> base set <值>
```

我们可以看到本格式与读取子命令十分相近，有区别的部分只是“<值>”与前面的get变成了set。所以，我们实际上可以很容易地得到：

```
/attribute @s generic.attack_damage base set 30
```

将自己的基础攻击力设为30

其中，**generic.attack_damage**为攻击力的属性ID，base set意味着我们要更改基值，而30是要更改的值。现在尝试攻击僵尸，你一拳就可以打倒。当然，我们可以刺激点：

```
/attribute @s generic.attack_damage base set 2000
```

将自己的基础攻击力设为2000 现在末影龙你也能一拳打倒了。那，最大可以设到多高呢？ 很夸张，夸张到可以闭眼按0：

```
/attribute @s generic.attack_damage base set 1000000000000000000000000
```



将自己的攻击力设置为 1×10^{35} 点。（1E35）夸张吧.....那，这该数值的上限是多少呢？这个值是一个double.....意思就是，在极端情况下你可以存308位数。

然而，每种属性都有一个接受范围。如果你输入的值大于这个范围，那么属性就会被设为最大可接受的值。比如，攻击力的最大上限是2048，所以虽然你在上面的命令中输入了一个大的离谱的数字，但是游戏会把属性设为2048。（小于最低值同理）好，从夸张的数字中钻出来，我们也可以设置小数：

```
/attribute @s generic.attack_damage base set 0.000001
```

将自己的攻击力设置为0.000001（基本上相当于没有。）小数在一些属性上使用频繁，比如速度或击退抗性：

```
/attribute @s generic.knockback_resistance base set 0.9
```

使自己免疫90%的击退 我们从上面的表格可以看到对于击退抗性1就代表完全抵抗，所以这里0.9即可抵挡90%。**到这里，我们要再次提醒：**

****本子命令base set修改的是****基础*值，而非加成或最终值****

为了让你明白这一点，与下面的内容区分，让我们举这样一个例子：假设有玩家，其基础攻击力为1：

```
/attribute @s generic.attack_damage base get
```

****读取该玩家基础攻击力，**返回1**

他手持钻石剑，为他增加了6点加成数值：

```
/attribute @s generic.attack_damage get
```

****读取该玩家最终攻击力，**返回7**

现在运行命令：

```
/attribute @s generic.attack_damage base set 4
```

将本玩家的基础攻击力设为4点

再次读取其攻击力：

```
/attribute @s generic.attack_damage get
```

读取该玩家最终攻击力，返回10

因为攻击力较之前增加了3点，所以总攻击力变成了10。注意，我们所增加的是**基础攻击力**，所以：

```
/attribute @s generic.attack_damage base get
```

****读取该玩家基础攻击力，**返回4**

如何？

下面是更多的例子，可以跳过到下一节

1.

```
/attribute @e[type=zombie,limit=1] generic.follow_range base set 1000
```



将一只僵尸的追踪距离设置为1000格 注意/attribute只能处理一个实体，所以我们添加了limit=1。

2.

```
/attribute @s generic.max_health base set 1
```

将自己的最大生命值设为1点 注意在低版本中（1.20.4以下），最大生命值更改后，需要在生命值变动之后才会刷新最大生命值。

3.

```
/attribute @r generic.movement_speed base set 100
```

将随机玩家的移动速度设为100 玩家的默认移动速度为0.1，所以100是一个非常快的数值。

4.

```
/attribute @s generic.attack_damage base set -1
```

将自己的攻击力设为 - 1 即使是负数也可以接受为值。但是攻击力属性不接受负数，所以会被设成0。然而其他属性，比如幸运，接受负值)

5.

```
/attribute @s minecraft:generic.gravity base set -0.05
```

将自己的重力设置为-0.05

这会让你向上飘，很有趣不是么？

4. 属性修饰符 在上面，我们探讨了如何更改**基础值**。我们可以轻易地将玩家变成神，抑或是手无搏鸡之力的恐怖地图受害者。但是一款游戏，角色只有基础数值是的话是很空虚的。除了游戏角色本身的属性，当其佩戴装备时，一般会有属性的加成。Minecraft不例外哈。玩家的基础攻击力只有1点，但是钻石剑拥有+6点的属性加成。基础数值经过加成数值的“洗礼”后，才是最终的属性，7。(1+6) 钻石剑为我们提供了加成数值。但这所谓的“加成数值”是有学名的。

这些加成，被称作**属性修饰符(Attribute Modifiers)**

我们说，钻石剑提供一个攻击力+6的****属性修饰**符**

当实体的基础数值经过属性修饰器的运算后，就会得到最终数值。正常来讲啊，在普通生存中，属性修饰器一般由装备给予，比如，上面讲的，使用钻石剑。但实际上不止装备，修饰器有时也会由其他道具，比如药水（状态效果）添加。比如，I 级的力量药水会给予3点额外攻击力.....或者说，拥有1级力量效果会给予玩家一个攻击力+3的属性修饰器。或者.....属性修饰器也可以由游戏机制给予.....比如，你是怎么疾跑的？你之所以可以疾跑，是因为在疾跑时游戏给予了你一个速度+130%的属性修饰器。为什么女巫在喝药水

的时候会减速（你也会）？因为游戏为其添加了一个速度 - 0.25的属性修饰器。 你可能看出来了，

只要，你的某属性与基础值有差，那一定是因为某个属性修饰器

以上提到的所有属性修饰器都是由某种装备或机制提供的。

但接下来的命令，/attribute的第三个子命令，/attribute modifier，即可为指定的实体添加一个永久的属性修饰器。

大概.....相当于为目标装备一个“虚拟”的装备吧。

5. 使用/attribute modifier 命令/attribute的子命令/attribute modifier用于管理目标上的自定义属性修饰符。

该子命令的语法在1.21之后大幅度地更改了。下面会依次描述不同版本的情况。

首先是1.20.6以及之前的版本：

先来说，如何添加一个属性修饰器： 格式：

```
/attribute <目标> <属性ID> modifier add <UUID> <名称> <值> <运算>
```

嚯，这个语法里有挺多陌生的东西嘛.....

```
/attribute <目标> <属性ID> modifier add
```

首先，来看一下UUID是什么。各位知道**ID（Identity, 识别码）**是什么吧。用于区分不同东西的一个“身份证”。那知道UID（Unique Identifier, 唯一识别码）是什么么？和ID差不多的东西。B站就在用，区分不同用户。Unique是在说，这种ID具有唯一性**。**（不然怎么区分不同用户）那么，UUID又是什么呢？**通用唯一识别码（Universally Unique Identifier**，UUID**）**。看名字就知道，这和ID差不多，实际上就是一串用于区分不同东西的数字而已。只不过，“唯一”指明了这串码不会重复，“通用”说明这是国际通用的标准。是的，不止Minecraft在用，UUID也不是Mojang发明的。世界上所有的UUID都由一些算法生成。简单概括：

UUID就是具有唯一性的，由一种通用算法生成的ID

我们不会深入接触，有兴趣的话可以在去百科搜索UUID查看。这里你只需要知道这是一种ID就好了。回到命令上：为什么添加一个属性修饰符需要一个

UUID呢？当然是为了区分多个不同的UUID啊。你想想，你可以添加那么多不同的属性修饰器，当然需要某种方法来区分啦。UUID长这样：数字-数字-数字-数字-数字 由5个数字组成，中间以连字符相接。比如：

0-0-0-0-0

这就是一个UUID了。任意更改数字就可以创造出其他UUID。比如：

88888888-4632-4444-3333-3141592657

可以随意乱编，但注意对于任意实体，其所有的属性修饰器的UUID必不能重复。同时你要把UUID记下来，不然无法对一个已添加的属性修饰符做出更改或删除——游戏完全使用UUID来区分不同的属性修饰器。所以我推荐别弄得太复杂。注意：1. 这只是UUID的一种表现形式，还有其他的比如高低位，以及在NBT说过的整形数列等。2. 这是十六进制。所以可以用A~F，比如1-A-fab3-231-ABCDEF123456这样。 5.2 <名称>

/attribute <目标> <属性ID> modifier add

接下来：什么是名称？

很简单：随便填

讲完了。是的，这里可以随便填。什么都行，愿意填什么填什么，重复也可以。这有点像是记分板或队伍的显示名称一样的东西。除了使用/data，你在加入这个属性修饰器后就再也无法看见它。

不过还是尽量只用字母或数字为好。如果要用空格等特殊字符，你需要把名称用双引号围起来。（这个名称大概是用方便记录的。毕竟UUID一串数很难记。用/data可以筛选出拥有特定名称的属性修饰器.....但还是没啥用因为对于玩家想删除还是要用/attribute，还是要用UUID） 5.3 <值> 与 <运算>

/attribute <目标> <属性ID> modifier add

让我们看向最后两个元素：值与运算。这是什么呢？这两个元素是在说，该属性修饰器要对属性作出怎样的更改。其中<运算>其实就是运算。比如是加法“攻击力+6”还是乘法“攻击力×150% (攻击力×1.5)”。而值就是那个被加或者乘上去的数字。属性修饰器是相当于给基础数值套个公式，算出最终属性。可以利用的运算，有三种：即**增加**(add)，**倍率**(multiply_base)，与**最**

终倍乘(multiply)。先看那么多复杂的玩意，我们只用增加，先把我们的第一条命令写出来：

```
/attribute @s generic.attack_damage modifier add 0-0-0-0-0 sha_dou_xin
```

这条命令为目标的**攻击力**属性增加了一个攻击力+3的属性修饰器，其UUID为0-0-0-0-0。解析：

这就是你的第一条命令啦 但是除了运算add，也就是增加，我们还有另外两种运算：下面将深入了解：**5.4 运算**可以利用的运算，有三种：增加(add)，倍率(multiply_base)，与最终倍乘(multiply)。公式是这样的：**最终属性值 = [(基础值 + **修饰符: **增加数值) × (1 + **修饰符: **倍率)] × (1 + **修饰符: 最终倍乘)**** 让我们一个个来：****1. 增加（又名：增量操作）****(add) 首先，你可以对一个属性做加（减）法。假设某玩家的基础攻击力是1，**没有**属性修饰器，那么该玩家的最终攻击力就是：

$$\text{最终属性值} = [(\text{基础数值} + \text{增加数值}) \times (1 + \text{倍率})] \times (1 + \text{最终倍乘})$$

然后我们用/attribute命令为其添加了一个攻击力+3的属性修饰器——

```
/attribute @s generic.attack_damage modifier add 0-0-0-0-0 sha_dou_xin
```

那么，该玩家的最终攻击力就是：

$$\text{最终属性值} = [(\text{基础数值} + \text{增加数值}) \times (1 + \text{倍率})] \times (1 + \text{最终倍乘})$$

- **拥有多个增加类属性修饰器时，会相加：**

假设我们又给了该玩家一个攻击力+1的属性修饰器。

```
/attribute @s generic.attack_damage modifier add 0-0-0-0-1 sha_dou_xin
```

那么，该玩家的最终攻击力就是：

最终属性值 = [(基础数值 + 增加数值) × (1 + 倍率)] × (1 + 最终倍乘)

如何？ **2. 倍率（又名：**倍率操作**）(multiply_base) ****倍率**，或者说“倍率增量”，会将指定一个倍率。或者说，乘以一个值。注意这在增加运算后运行，换句话说，乘的是已经被增加后的值。**继续上面的假设：** 在拥有上面说过的攻击力+3，+1的同时，我们再添加一个倍率属性修饰器攻击力+50%：

```
/attribute @s generic.attack_damage modifier add 0-0-0-0-2 whatever 0.
```

那么，该玩家的最终攻击力就是：

最终属性值 = [(基础数值 + 增加数值) × (1 + 倍率)] × (1 + 最终倍乘)

注意，倍率在加法之后算。

- **若有多个倍率的属性修饰器，倍率数值将按加法叠加：**

再添加一个攻击力 - 20%的属性修饰器：

```
/attribute @s generic.attack_damage modifier add 0-0-0-0-3 whatever_2
```

那么，该玩家的最终攻击力就是：

最终属性值 = [(基础数值 + 增加数值) × (1 + 倍率)] × (1 + 最终倍乘)

3. **最终倍乘** (multiply) 最终倍称也是乘法，在倍率之后运行。于倍率不同之处在于多个最终倍称**以乘法叠加**而非加法：继续上面的例子：我们在原来的基础上再为玩家添加一个值为2的最终倍乘：

```
/attribute @s generic.attack_damage modifier add 0-0-0-0-4 final_1 2 m
```

那么，该玩家的最终攻击力就是：

$$\text{最终属性值} = [(\text{基础数值} + \text{增加数值}) \times (1 + \text{倍率})] \times (1 + \text{最终倍乘})$$

若有多个最终倍乘的属性修饰器，数值将按乘法叠加。

如，在原来的基础上我们再添加一个最终倍乘4的属性修饰器：

```
/attribute @s generic.attack_damage modifier add 0-0-0-0-5 sha_dou_xin
```

那么，该玩家的最终攻击力就是：

$$\text{最终属性值} = [(\text{基础数值} + \text{增加数值}) \times (1 + \text{倍率})] \times (1 + \text{最终倍乘})$$

以上就是所有的关于添加属性修饰器的格式了。当然，一个完整的命令不会只能添加不能删除的。所以，让我们继续：

5.5 读取与删除 如果你突然忘记了某个属性修饰器是什么了，可以使用以下命令读取：

```
/attribute <目标> <属性ID> modifier value get <UUID> [<倍率>]
```

比如：

```
/attribute @p generic.movement_speed modifier value get 0-0-0-0-0
```

读取最近玩家速度属性的，UUID为0-0-0-0-0的属性修饰器 这里要求你提供UUID，这就是为什么需要你记下UUID的原因。

最后，如果你不想要一个属性修饰器，可以将其删除：

```
/attribute <目标> <属性ID> modifier remove <UUID>
```

比如：

```
/attribute @p generic.movement_speed modifier remove 0-0-0-0-0
```

将最近玩家速度属性的，UUID为0-0-0-0-0的属性修饰器删除 就这些了哦 最后做提醒，给玩家的属性修饰器在玩家死亡时会消失。5.6 **版本**1.21+的**modifier子命令** 以上modifier子命令的内容都只适用于1.20.4版本以及之前。在本教程此时的最新版，1.21更改了命令的格式。

大部分逻辑没有改变，除了以下几点：

1. 不再适用UUID + 名称区分各不同的属性修饰符，而是适用命名空间ID。这已经是之前探讨过多次的概念。

2. 更改了几个运算的名称：

- add → add_value
- multiply_base → add_multiplied_base
- multiply → add_multiplied_total

几个例子：

```
/attribute @s minecraft:generic.max_health modifier add foo:bar 10 add
```



给自己添加一个属性修饰符，内容为增加10点最大生命值

```
/attribute @s minecraft:generic.max_health modifier remove foo:bar
```

删除刚刚添加的属性修饰符

注释【1】 由于一个Bug，倍率后的值无法在聊天框中被显示出来。虽然使用/execute store 可以得到正确的数字。考略到没讲到/execute，这里跳过倍率没有举任何例子。【2】僵尸有1点基础护甲值，会减免4%的伤害。因此要21下才打的死20生命值的僵尸。