

## 布尔值的历史

19 世纪的英国数学家乔治·布尔发明了仅使用 1 和 0 构成(或者真 (true) 和假 (false) )的逻辑系统,因此,使用布尔 (Boolean) 纪念他对逻辑运算的特殊贡献。布尔代数最终成为现代计算机科学的基础。

在现实生活中,我们无时无刻都在使用布尔表达式。例如,计算机使用它来决定到底执行程序的哪一个分支;机械手臂也需要布尔表达式,当它检查流水线上移动着的零件时,如果 `goodQuality` (质量好) =true,则将零件移动到第一个盒子,如果 `goodQuality`=false,则移动到第二个盒子;在家庭安全系统中,若输入了错误的代码,即 `correctCode` (代码正确) =false,则警报声响起,若 `correctCode`=true,关闭警报声;当你在超市购物刷卡时,若银行卡的状态是有效的 (true),则授权访问远程服务器,反之,若状态是无效的 (false),则拒绝访问远程服务器;当汽车中的计算机检测到发生碰撞时 (`collision`=true),它便会下令弹出气囊;当电池电量不足时 (`batteryLow`=true),你的手机就会显示一个警告图标,但若电量还不算太低 (`batteryLow`=false),警告图标就会消失。

以上案例都说明了不同的布尔值让计算机采取了不同的行为。

注意,表 6-1 中的积木块都是六边形的。正如第 5 章所讲,六边形意味着该积木块的求值结果是非真即假的布尔值。因此,这种表达式也叫作布尔表达式。

例如,布尔表达式 `price<2000` 测试变量 `price` 的值是否小于 2000。如果 `price` 小于 2000,则整个积木块的求值结果(或者说返回的结果)为 true;否则返回 false。因此,你可以用如下形式的语句构建决定条件:“如果 (`price<2000`),那么买车”。

在详细介绍如果...那么积木之前,让我们先通过一个简单的案例说明 Scratch 中的布尔表达式是如何求值的。