

词语模块

词语模块说明

电机模块类别

电机模块不仅具备驱动电机运转的功能，还能获取电机的相关信息。其中，“电机模块”是最为常见和通用的模块类别，能够广泛应用于各种电机控制场景中。

使电机运行指定单位数



这个模块具备控制一个按顺时针或逆时针方向运行的能力，允许用户设定电机运行的特定圈数、秒数或度数。此外，电机的速度可以通过“设定电机速度模块”进行灵活调整。

启动电机



这个模块具备让一个持续以顺时针或逆时针方向运转的功能。用户可使用“设定电机速度模块”调整电机的运转速度。

停止电机



这个模块具备停止一个电机运行的功能。当启动此模块时，电机将会进行制动，确保能够迅速且完全地停止运转。需要注意的是，电机在停止后并不会保持其当前位置，可能会因为外部因素或内部惯性而发生位置变动。

设定电机速度



这个模块允许用户设置一个电机的速度，其速度调节范围覆盖了从-100 到 100 的完整区间。在这个范围内，负值表示电机将以反方向运转，而正值则代表电机在正方向上的运转速度。通过此模块，用户可以精确地控制电机的运转速度和方向，默认值为 50%。

电机位置



这个模块具备报告电机当前位置的功能。通过该模块，用户可以实时获取电机的精确位置信息，

电机速度



这个模块能够准确报告电机的当前实际运转速度，而不是用户通过“设定电机速度”模块所设定的速度值

运动模块

此模块能够同步运行两个电机，它们主要用于移动驱动底座。只有相同类型的电机（例如两个中型电机）才能同步。

停止电机，使其惯性滑行

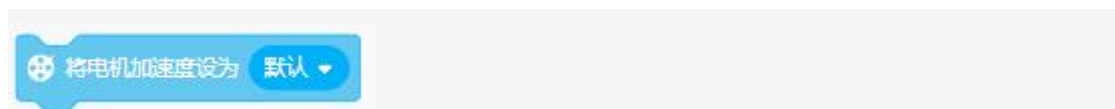


这个模块在运用具备特定单位数的“电机模块”或“停止电机模块”时，用户可以针对电机选择三种不同的停止机制。电机能依照以下三种模式停止工作：

制动模式：作为默认的停止方式，电机在接收到停止指令后会首先启动动力制动，随后施加摩擦以确保电机完全停止。

- 位置保持模式：在此模式下，电机同样采用动力制动来停止。但特别的是，如果电机因外力作用而偏离了设定的停止位置，它会主动调整并返回到原位置。
- 惯性滑行模式：选择此模式时，电机在停止时会直接切断电源，让电机自然依靠惯性逐渐减速并最终停下。

设置电机加速度



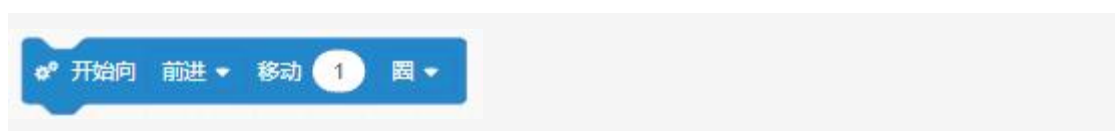
这个模块允许用户为单个设置加速和减速参数。加速度可以灵活调节，包括默认、快、平衡、平滑、慢、更慢多种选择。



这个模块确实为用户提供了调整大电机或中电机 **PID** 值的功能。**PID**，即比例（**P**）、积分（**I**）和微分（**D**）的缩写，。通过调整 **PID** 值，用户可以更精确地控制电机的运行特性。

更多运动模块

持续移动指定单位数



这个模块专为控制驱动底座的移动而设计，能够实现向前、向后移动以及顺时针、逆时针转向移动。用户可以根据需要，设定驱动底座移动或转动的特定时间（以秒数计）或角度（以度数或圈数计）。

开始移动



这个模块具备持续启动驱动底座向前、向后移动或顺时针、逆时针转向移动的功能。用户可以通过该模块灵活控制驱动底座的移动方向。

停止运动



此模块具备通过关闭电机来立即停止驱动底座所有运动的功能。用户可以激活此模块，使电机停止工作。

设置移动速度



这个模块允许用户设置驱动底座的运动速度。速度调节范围涵盖从**-100**到**100**的完整区间。在这个范围内，正值代表驱动底座以正转方式移动，而负值则意味着以反转方式移动。默认的速度设置值为**50%**。

设置运转电机



这个模块确实具备指定两个驱动电机所连接端口的功能。

设置组合电机速率



这个模块具备独立设置驱动底座两侧电机速率的功能，用户可以根据需要为左侧和右侧的电机分别设定不同的运行速度。



这个模块可以设置驱动底座电机的停止防止，用户可以针对电机选择三种不同的停止机制。电机能依照以下三种模式停止工作：

制动模式：作为默认的停止方式，电机在接收到停止指令后会首先启动动力制动，随后施加摩擦以确保电机完全停止。

位置保持模式：在此模式下，电机同样采用动力制动来停止。但特别的是，如果电机因外力作用而偏离了设定的停止位置，它会主动调整并返回到原位置。

惯性滑行模式：选择此模式时，电机在停止时会直接切断电源，让电机自然依靠惯性逐渐减速并最终停下。

灯模块类别

灯模块能够开启、关闭和设置 **7x9** 矩阵灯。

开启 7x9 矩阵灯



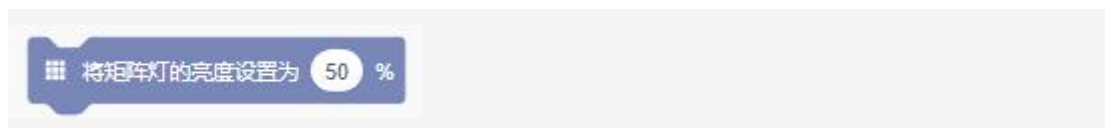
这个模块具备强大的图案创建与显示功能，它能够在 **7x9** 矩阵灯上生成并展示用户自定义的图案和颜色。一旦图案被创建并显示在矩阵灯上，它将持续亮起，直到矩阵灯接收到新的操作指令或整个程序被停止。

在 7x9 矩阵灯上写入内容



此模块具有在 **7x9** 矩阵灯上滚动显示文本字符串的功能，它以逐个字母的方式在显示屏上滚动呈现文本内容。这种滚动显示的方式不仅使得文本能够在有限的显示区域内完整地呈现

设置 7x9 矩阵灯的亮度



这个模块的功能是为程序栈中下一个将要使用矩阵灯的模块设置 **7x9** 矩阵灯的亮度。在使用此模块时，用户可以指定一个亮度值，该值将应用于下一个启动的矩阵灯模块。亮度值的范围通常是 **0%**到 **100%**，其中 **0%**表示完全关闭灯光，而 **100%**则表示灯光达到最大亮度。

设置矩阵灯颜色



此模块专门用于设置 **7x9** 颜色矩阵灯的颜色，为用户提供了灵活且强大的颜色控制功能。

清除内容



这个模块具备清除 **7x9** 矩阵灯显示内容的功能，用户可以使用它快速清除矩阵灯上的所有显示信息。

以坐标轴方式设置像素格亮灯

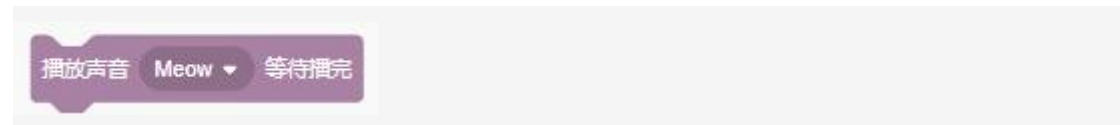


这个模块能够以坐标轴的方式精准控制 **7x9** 矩阵灯中的单个或多个像素格，实现其点亮功能。

声音程序块

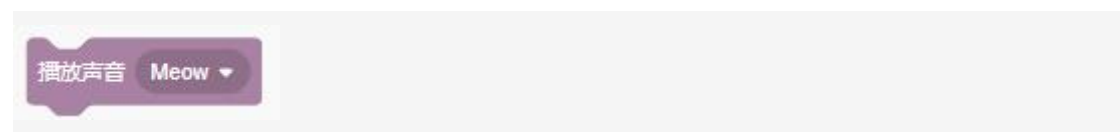
声音程序块能够播放设备中的声音。

播放声音直到结束



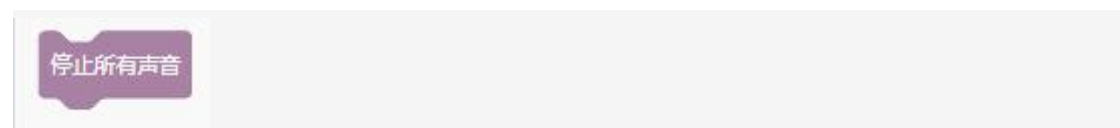
这个模块的功能允许用户在设备上播放选定的声音，并且在声音播放期间，程序堆将被暂停执行，以确保声音播放的完整性和连贯性。在播放声音时，程序堆会暂时停止执行其他任务，直到声音播放结束后，程序堆才会继续执行后续的任务。

开始播放声音



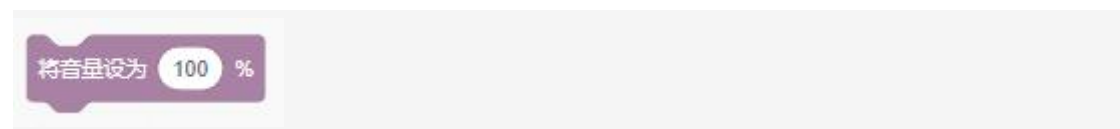
这个模块的功能是在设备上开始播放选定的声音，并且与此同时，它会立即运行程序堆中的下一个模块。这意味着声音播放与程序执行是并行进行的，而不是像之前的模块那样等待声音播放结束后再继续执行。

停止所有声音



这个模块具备能够立即停止当前播放的所有声音。

设置音量



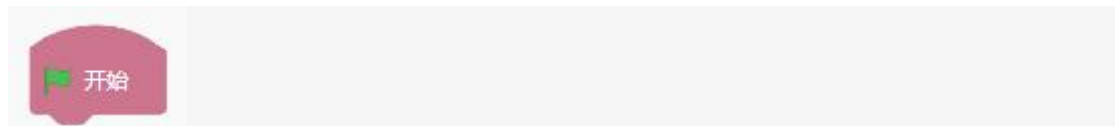
此模块具备设置音量的功能，允许用户根据需要调整声音的播放音量。音量范围通常是从 **0%** 到 **100%**，其中 **0%** 表示静音，即声音完全关闭；而 **100%** 则表示音量达到最大，声音以最大响度播放。

事件模块

事件模块是由“帽子模块”构成的独特组件，它在程序堆中占据着一个不可或缺且固定的位置——始终作为第一个模块存在。其他所有模块都必须按照逻辑顺序，逐一拼接在帽子模块的下方。这种结构确保了程序的清晰性和执行的顺序性。

在启动整个程序堆时，帽子模块扮演着至关重要的角色。它的核心功能是作为触发机制，当某个特定的事件发生时，帽子模块将被激活并启动后续模块的执行。这种事件驱动的方式使得程序能够灵活响应外部或内部的变化，实现预期的功能。

程序启动时



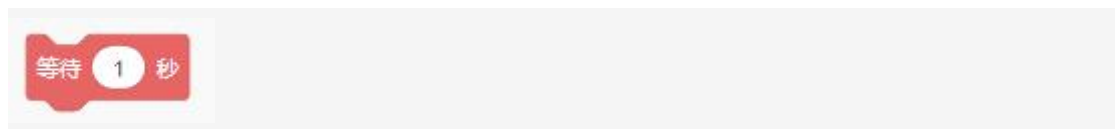
程序一旦启动，这个核心模块会自动执行一系列操作，即依次运行所有连接在其下方的模块，遵循从上到下的顺序。用户可以选择通过界面上的“运行按钮”来触发整个程序。另外，如果系统当前没有处于“正在运行模式”，用户还可以通过物理操作，即按下主机上的“中间按钮”，来启动程序。

当颜色为

控制模块

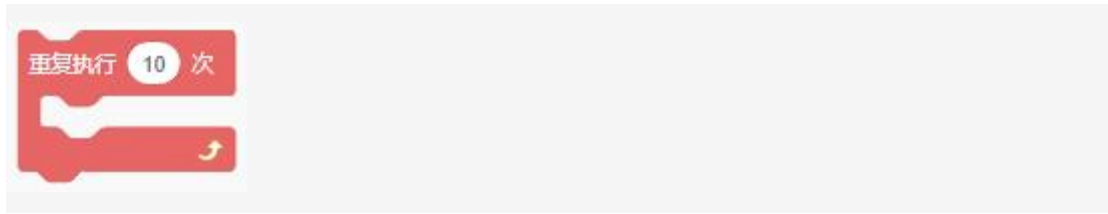
控制模块类别涵盖了所有能够调整模块执行顺序和流程的模块，比如“等待”结构、循环和条件判断等。

等待指定秒数



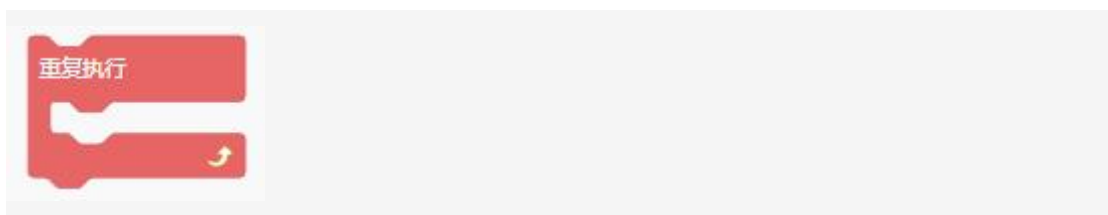
这个模块的功能是使程序暂停执行指定的秒数。无论用户输入的是整数还是小数，该模块都能够准确地暂停相应的时间长度。

重复循环



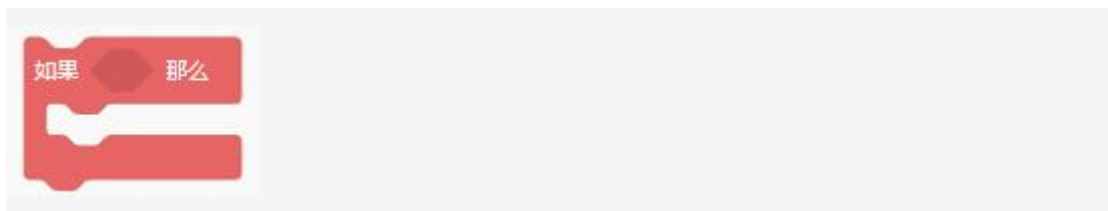
这个模块具备循环执行的功能，它允许其包含的所有模块按照指定的次数进行重复执行。在达到设定的循环次数后，程序将跳出循环并继续执行后续的模块或操作。这种循环机制在程序设计中非常实用，尤其是在需要重复执行某段代码或一系列操作时。

永久循环



这个模块具备无限循环的功能，它会持续不断地执行其包含的所有模块，形成一个永久性的循环过程。这种循环会一直进行下去，直到用户主动采取停止措施。

如果那么



这个模块负责检查一个特定的布尔条件是否成立。根据检查的结果，模块的行为会有所不同：

如果该布尔条件为“真”（即条件成立），则此模块会按顺序执行其内部包含的所有子模块。这些子模块会按照它们在模块内的排列顺序依次运行，完成各自的任务。

如果该布尔条件为“伪”（即条件不成立），则此模块所包含的所有子模块将被跳过，不会执行任何操作。

•

如果那么否则



这个模块具备条件判断的功能，它会根据所指定的布尔条件来决定执行哪个区块内的模块。

具体来说，当条件为“真”时，此模块会执行其第一区块内的所有模块，并按照顺序运行它们。完成第一区块的模块执行后，程序将继续运行后续的程序堆，即继续执行此模块之后的模块。

相反，如果条件为“伪”，此模块将跳过第一区块内的模块，转而执行其第二区块内的所有模块。在条件不满足的情况下，程序将执行与第一区块不同的代码路径。

等待直到



这个模块具备暂停程序执行的功能，它会等待直到指定的布尔条件变为“是”后才会继续执行后续的程序堆。

重复直到式循环



这个模块的功能是创建一个循环结构，该循环会持续执行其内部包含的所有模块，直到指定的布尔条件变为“是”。在循环期间，这些模块会反复运行，直到满足终止条件。一旦指定的布尔条件变为“是”，循环将终止，并且程序将跳出此模块，继续执行此模块下方拼接的模块。

停止



停止全部程序：当选择此功能时，模块会立即中断当前所有正在执行的程序堆，确保它们停止运行。

停止这个：如果选择此项，模块将仅停止当前所在的程序堆，而不会影响其他并行运行的程序堆。

跳出循环



这个模块具备在循环模式中跳出当前循环并继续执行后续程序的功能。

传感器模块

传感器模块能够接收来自传感器（如颜色传感器、距离传感器、力传感器、陀螺仪传感器等）的信息。

是否为指定颜色？



当颜色传感器检测到指定的颜色时，该模块会按照其预先设定的逻辑或编程指令返回一个信号，这个信号可能是一个电子信号、数字代码或文本信息。

颜色



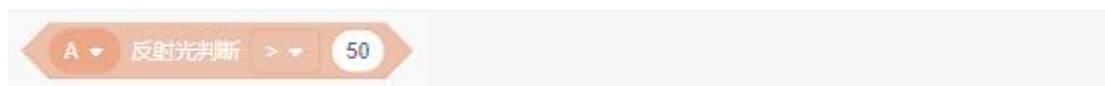
如果颜色传感器模块能够返回所检测到的颜色的当前值，那么它通常具备读取传感器数据并将其转换为可读格式。

识别原始颜色



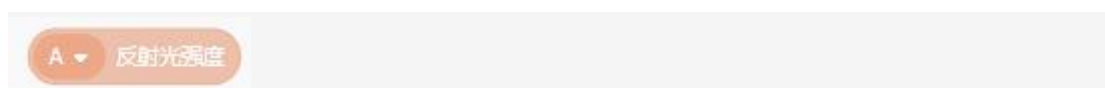
颜色传感器确实具备读取和识别三原色中红、绿、蓝三种颜色的能力。

反射光是否满足条件？



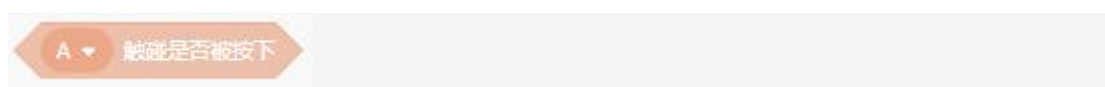
当反射回颜色传感器的光大于、等于或小于指定的百分比时，此模块返回“是”。

反射光



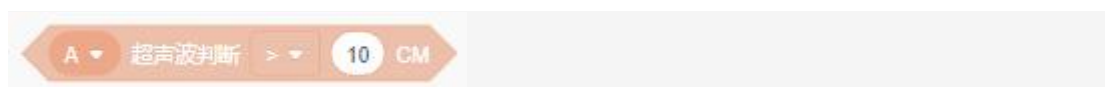
颜色传感器模块能够报告反射回颜色传感器的光线的当前值。

是否被按压？



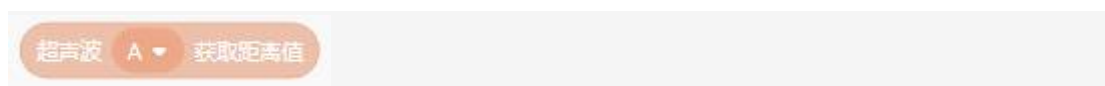
当力传感器被按压或松开时，此模块返回的逻辑是为“真”。

距离是否满足条件？



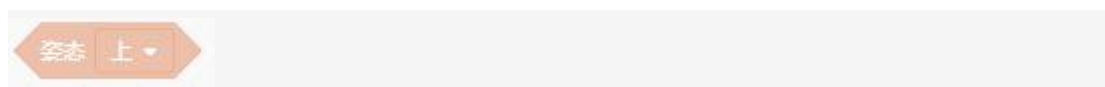
当超声波传感器检测到与指定距离之间的关系（大于、小于或等于）时，模块返回逻辑为“真”。

距离



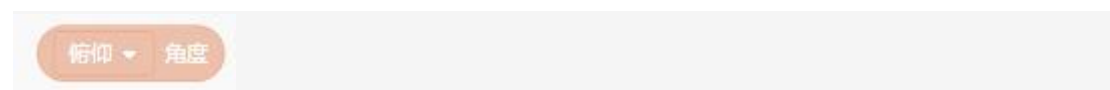
此模块能够实时提供超声波传感器当前检测到的具体距离数据。

是否倾斜？



当主机自水平状态（按钮朝上）起，开始向指定方向倾斜时，该模块将返回逻辑为“真”。

智能集线器俯仰角、横滚角和偏航角



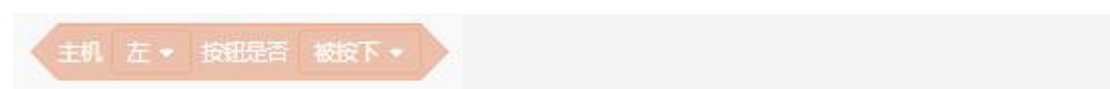
这个模块负责提供主机当前的俯仰角、横滚角和偏航角数据。这些术语——俯仰、横滚和偏航——原本是用于描述飞机在空中的运动状态，但同样适用于任何在三维空间中发生旋转的物体。以飞机为例，这些角度可以精准反映其飞行姿态的变化。

俯仰角是描述飞机机头相对于水平线向上或向下倾斜的角度。

横滚角则是指机翼相对于水平线向上或向下倾斜的角度。

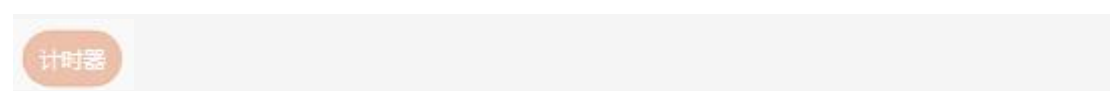
而偏航角则用来表示飞机相对于地面的方向，即飞机机头所指的方向与地面某一固定方向（如正北方向）之间的夹角，它决定了飞机在水平面上的航向。

主机按钮是否被按压？



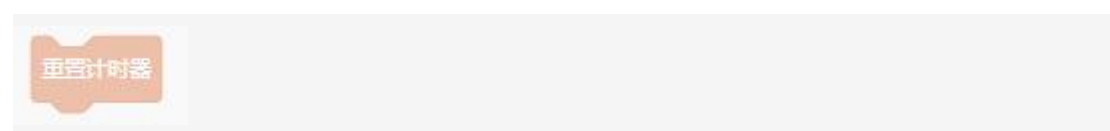
如果“左按钮”或“右按钮”中的任意一个被按下或松开，该模块将返回逻辑为“真”。

计时器



程序启动后，这个模块会启动一个计时器，以秒为单位记录经过的时间。每当程序重新启动时，计时器也会相应地进行重置，开始新一轮的计时。

重置计时器



这个模块可以用来重置计时器。

获取声音强度



这个模块能够捕捉并获取到接收到的声音信号，并将其转换为相应的数值数据。

坐标轴的加速度



这个模块具备获取三维坐标轴 **XYZ** 上加速度数据的功能，能够实时检测并输出 **X** 轴、**Y** 轴和 **Z** 轴上的加速度值。

手柄按键指令



这个模块能够检测手柄按键的状态，当按键被按下或松开时，它能够判断并返回相应的逻辑值。如果检测到按键的状态发生了变化（无论是被按下还是被松开），该模块会返回逻辑“真”。

手柄摇杆指令



这个模块能够捕获手柄摇杆在 **X** 轴和 **Y** 轴上的坐标数据，从而实现对摇杆位置的精确读取。

运算符模块

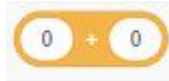
运算符模块能够实现所有数学运算。

挑选随机数



这个模块具备在指定范围内随机选择数字的功能，且包括该范围的两个端点值。无论是整数还是小数，都可以作为最小值和最大值的设定。若设定值中包含小数，该模块将能够返回相应的小数作为随机结果。

加



这个模块具备加法运算功能，能够将两个值进行相加并返回它们的和作为结果。无论是整数还是小数，都可以作为输入值进行相加运算。

减



这个模块具备减法运算功能，可以接收两个值作为输入，并将第一个值减去第二个值，然后返回运算结果。无论输入的是整数还是小数，模块都能正确执行减法运算并返回相应的差值。

乘



这个模块具备乘法运算功能，可以接收两个值作为输入，并将它们相乘，然后返回运算结果。无论是整数还是小数，都可以作为输入值进行乘法运算。

除



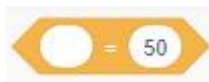
这个模块具备除法运算功能，可以接收两个值作为输入，将第一个值除以第二个值，并返回运算结果。无论输入的是整数还是小数，模块都能正确执行除法运算，并返回相应的商。

小于



这个模块负责比较两个值的大小关系。具体来说，它会检查第一个值是否小于第二个值。如果第一个值确实小于第二个值，那么模块将返回“是”作为确认；如果第一个值不小于第二个值，无论是等于还是大于，模块都会返回“否”来表示比较结果。

等于



这个模块负责比较两个值是否相等。它会接收两个输入值，然后检查它们是否完全相同。如果第一个值等于第二个值，无论它们是整数还是小数，模块都会返回“是”，表示两者相等。如果它们不相等，模块则会返回“否”，表示两者不等。

大于



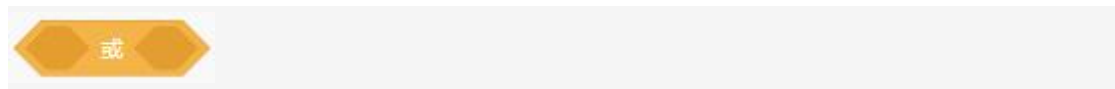
这个模块的功能是对比两个数值的大小。具体来说，它会接收两个输入值，并检查第一个值是否大于第二个值。如果第一个值确实大于第二个值，那么模块会返回“是”，表示第一个值较大。反之，如果第一个值不大于第二个值（即等于或小于），模块则会返回“否”，表示第一个值不是更大的那个。

与



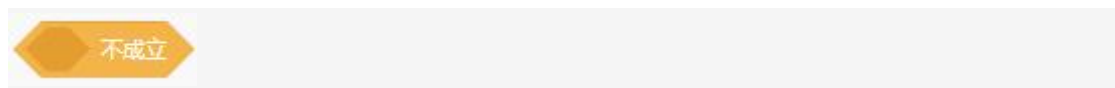
这个模块的功能是作为逻辑运算符，使用“与”（**AND**）条件来连接两个“布尔模块”。具体来说，它接收两个来自“布尔模块”的输入信号，只有当这两个输入信号都为真（**True**）时，该模块才会返回真（**True**）作为输出。如果任一输入信号为假（**False**），则输出为假（**False**）。

或



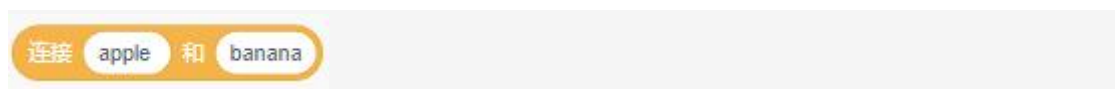
此模块的功能是作为逻辑运算符，使用“或”（**OR**）条件来连接两个“布尔模块”。它接收两个来自“布尔模块”的输入信号，并检查这两个信号中至少有一个是否为真（**True**）。只要有一个输入信号为真，该模块就会返回真（**True**）作为输出。只有当两个输入信号都为假（**False**）时，输出才会是假（**False**）。

非



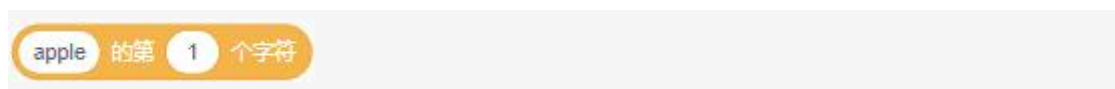
此模块的功能是逆转或取反其内部条件的布尔值。具体来说，它接收一个布尔值（真或假）作为输入，并返回该值的相反结果。如果输入是真（**True**），模块会将其逆转为假（**False**）；如果输入是假（**False**），模块会将其逆转为真（**True**）。

连接字符串



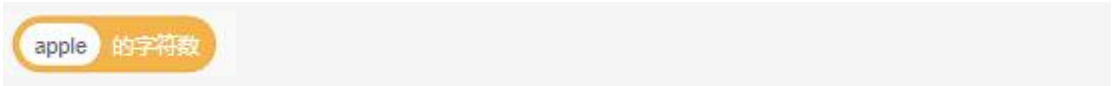
这个模块的功能是合并或连接两个文本值，并将它们组合成一个新的文本字符串。当用户将两个文本值，如“你好”和“世界”，输入到模块中时，模块会将它们拼接在一起，去除中间的任何空格或分隔符，并返回一个新的字符串“你好世界”。

字符串字母



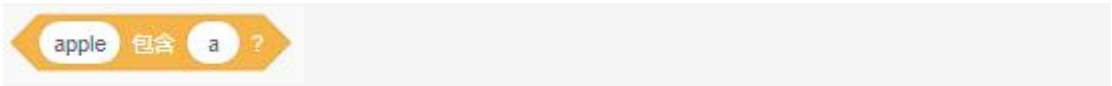
这个模块的功能是提取给定字符串中指定位置的字符。用户输入一个字符串，并指定一个位置（通常是以 **0** 或 **1** 为起点的索引），然后模块会返回该位置上的字符。例如，如果输入的字符串是“**EST**”，而指定的位置是 **1**，模块将返回字符串的第一个字符“**E**”

字符串长度



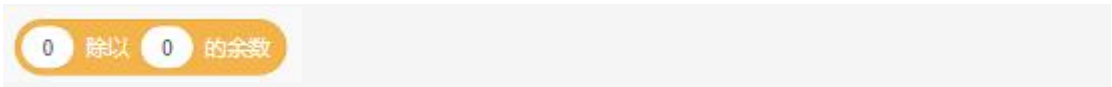
这个模块的功能是计算并返回给定字符串中的字符数。用户输入一个字符串，模块会统计该字符串中包含的字符数量（不包括空格或其他分隔符），并返回相应的整数值。例如，如果输入的字符串是“EST”，模块会计算出该字符串包含 3 个字符，并返回数字“3”。

字符串包含



这个模块的功能是检查指定的字符是否包含在特定的字符串中。用户需要输入一个字符串和一个字符，模块会遍历字符串中的每个字符，检查是否存在与指定字符相同的字符。如果找到了匹配的字符，模块会返回“是”，表示该字符确实存在于字符串中。如果字符串中不包含指定的字符，模块则会返回“否”。

求余

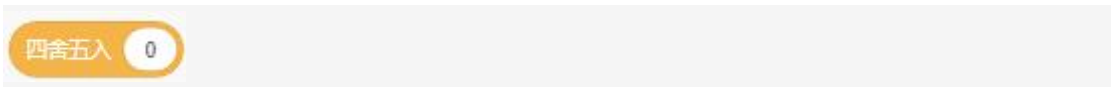


这个模块执行的是取余运算，它返回第一个值除以第二个值后的余数。对于正数的除法，这一运算过程是直观的。但在负数运算时，需要特别注意，余数必须始终为正数，因此计算过程略有不同。

例如：当第一个输入是 10，第二个输入是 3 时，10 除以 3 的余数是 1，这是标准的除法取余运算。然而，当处理负数时，比如第一个输入是 -10，第二个输入是 3 时，如果按照常规的除法运算，-10 除以 3 的商是 -3，余数是 -1。但在此模块中，为了保持余数为正数，我们会采用一种不同的计算方式。实际上，我们会将 -10 加上 3 的倍数，直到结果是一个非负数且小于 3 的数。在这种情况下，我们需要加上 3 的 4 倍（即 12），这样 -10 就变成了 2，2 除以 3 的余数是 2。因此，此模块会返回 2 作为 -10 除以 3 的余数。

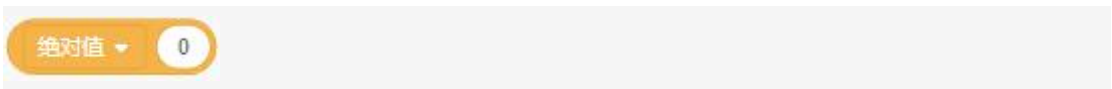
这样的设计是为了确保在负数运算中，余数的结果始终为正数。

四舍五入



这个模块的功能是将给定的数字舍入到最接近的整数，它遵循标准的舍入规则。具体来说，当给定数字的小数部分等于或大于 **0.5** 时，模块会将其向上舍入到下一个更高的整数；而当小数部分小于 **0.5** 时，模块会将其向下舍入到上一个更低的整数。

数学函数



这个模块的功能是将给定的数字作为输入值代入指定的数学函数中，并返回该函数计算得到的结果。用户需要提供需要代入的数字以及数学函数的定义或描述。

变量模块

变量模块类别确实是一个广泛而重要的类别，它涵盖了与变量、列表（数组）以及用户自定义的“我的模块”相关的所有功能模块。这类模块在编程和自动化流程中发挥着至关重要的作用，允许用户存储、管理和操作数据，以及创建和复用自定义的功能。

变量模块

变量模块允许用户创建和存储数据值，这些数据值可以是数字、文本、布尔值等。通过变量模块，用户可以轻松地在程序的不同部分之间传递和共享数据。例如，一个变量模块可能允许用户设置一个变量的值，然后在程序的后续部分中检索和使用这个值。

列表（数组）模块

列表（或数组）模块是处理有序数据集合的强大工具。这些模块允许用户创建列表，并在列表中存储多个值。用户可以对列表进行各种操作，如添加、删除、修改元素，以及执行搜索、排序等复杂操作。列表模块在处理大量数据或需要组织数据以进行进一步分析的场景中特别有用。

“我的模块”

“我的模块”类别允许用户创建和保存自定义的模块。这些模块可以包含用户自己编写的代码和逻辑，用于执行特定的任务或实现特定的功能。通过创建和使用“我的模块”，用户可以简化复杂的编程任务，提高代码的可重用性和可维护性。此外，这也为用户提供了一个学习和实践编程技能的平台。

变量



这个模块是专门设计用来报告或显示变量当前值的。当用户在系统中创建或更新一个变量时，系统会自动生成一个与此变量相关联的“报告变量值”模块。这个模块会带有变量的给定名称，以使用户能够清晰地识别它所代表的变量。

将变量设置为



这个模块的功能是设置或修改指定变量的值为给定的值。无论该变量原本是一个字符串还是数字，此模块都能对其进行赋值操作。

使用此模块时，用户需要指定两个关键信息：要修改的变量名称和新的值。模块会接受这两个输入，并将变量的值更新为指定的新值。

将变量更改给定值



这个模块的功能是对指定变量中的当前值进行更改操作，而不是简单地设置一个新值。从描述来看，它执行的是将当前值增加一个特定的数量的操作。

当模块被调用并指定了一个变量和一个要更改的数量时，它会找到该变量的当前值，然后将这个值加上指定的更改量，得到一个新的值，并将这个新值赋给变量。因此，如果变量原本包含一个数字，该数字就会按照指定的更改量进行增加。

在你给出的例子中，如果一个变量包含值 **4**，并且使用“将变量更改 **3**”模块，那么这个变量的值将会变为 **7**（即 **4 + 3**）。这表明该模块是执行加法操作来增加变量的值。