



目录

初学者使用教程—2

进阶者使用教程—73

高阶者使用教程—127

初学者使用教程

01

快速编程入门

02

界面基本操作

03

程序的基本控制

04

动画的基本原理

05

游戏的基本控制

06

程序块的基本使用

AIScratch是什么?

AIScratch是一款适合广大青少年朋友学习的图形化编程软件，它不需要掌握复杂的程序代码，也对英语水平没有要求，像搭建积木一样简单，可以快速上手。



学习编程

作为学习编程的入门平台，
掌握编程的基本技术



学习物联网

可以学习物联网硬件编程，
制作出智能化家居



学习人工智能

学习人工智能的基本技术



制作动画

可以制作生动有趣的卡通动画和视频



制作游戏

可以制作精彩的游戏程序，
软件包含可游戏元素



制作课件

可以制作互动的课件，
方便课堂教学

快速开始——我的Hello World程序(1)

让我们快速地了解程序的运行过程，做一个简单的Hello World（你好世界）程序。



1

2

① 点击【控制】按钮，找到  ② 把这个程序块拖入到代码区（在程序块上方按下鼠标左键不放，移动鼠标到代码区空白位置，松开鼠标左键）

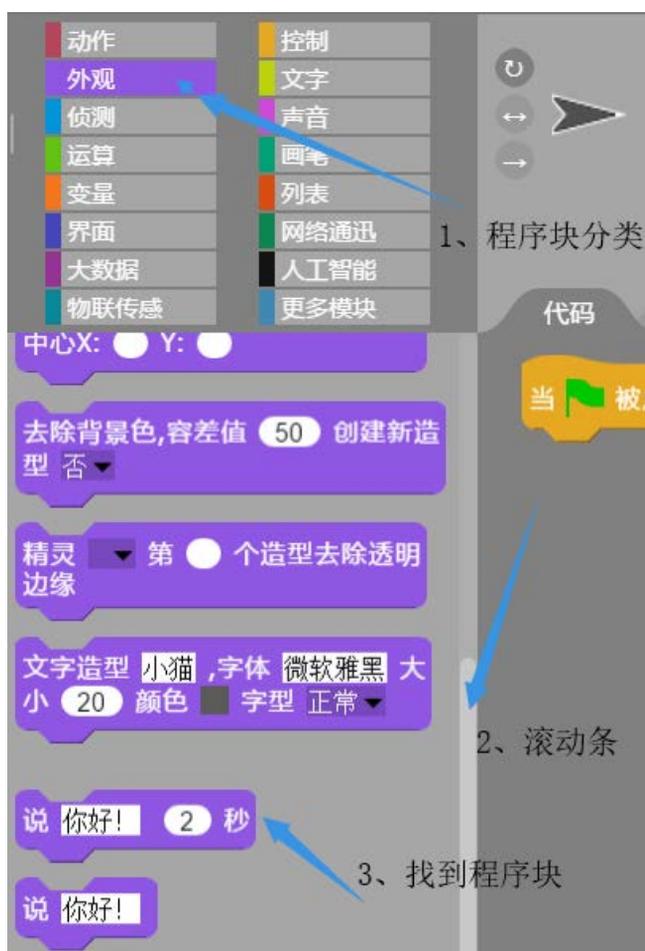


3

③ 按下shift键不放，点击程序块  出现白色的横线，键盘输入“说”，输入后出现两个程序块。

快速开始——我的Hello World程序(2)

找程序块的两种方法：1、点程序块分类按钮，在下方列表找到需要的程序块；2、按下shift键不放，点击鼠标左键，输入程序块的第一个字，下方列表中会出现需要的程序块。



第③步也可以这样找到程序块：点击程序块分类【外观】按钮，在下方出现许多到外观相关的程序块，下拉右侧的滚动条向下滚动（或向下滚动鼠标滚轮），直到找到。

④按下鼠标左键拖动 **说 你好!** 到代码区 **当 被点击** 的下方，当出现白色的横线时放下程序块。

⑤点击“你好！”，改成“Hello World!”

3

快速开始——我的Hello World程序(3)

运行程序代码的方法：1、直接点击程序块组合的任意位置，运行当前程序块；2、点击舞台控制区的运行按钮，运行全部程序代码；3、按F9运行全部程序代码。



6



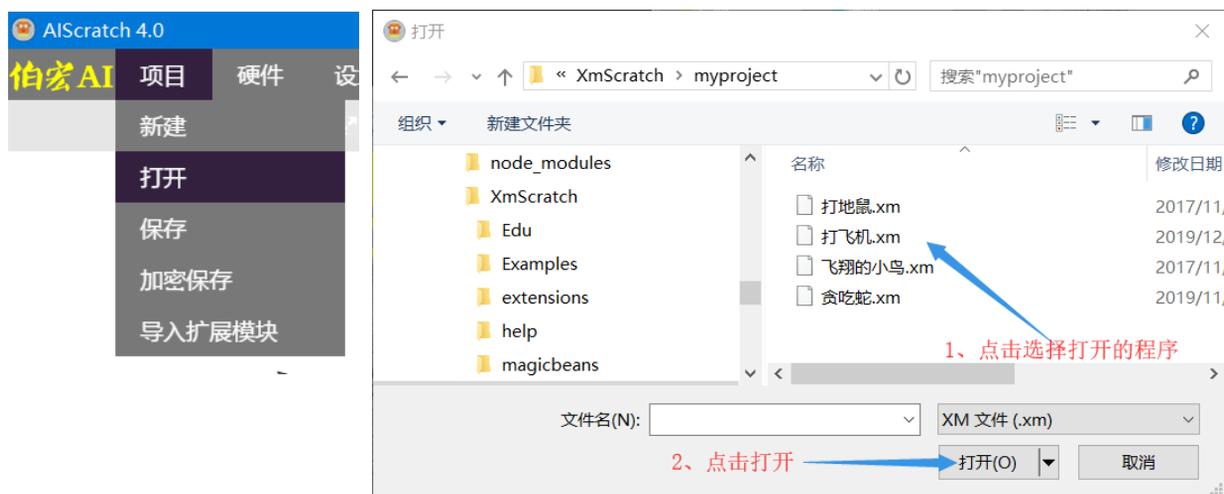
⑥点击程序块的任意位置，运行当前程序块代码；可以在舞台区观察运行结果。



舞台控制区的绿旗表示运行全部程序代码，而点击程序块运行程序时，只运行被点击部分的程序块代码，其他代码块不会运行。

快速开始——打开示例程序

AIScratch软件自带了很多程序案例，你可以打开后运行它，体验编程作品的运行效果。



点击顶部【项目】菜单，点击【打开】，在打开的对话框选择程序文件，点击对话框中的【打开】按钮。打开后按F9运行吧。

同样，我们可以把自己编辑好的项目，通过【保存】功能保存到本机电脑，下次使用【打开】功能导入。



界面的基本操作——认识界面元素 菜单

AIScratch软件界面分成菜单区、舞台区、精灵区、程序块区、精灵状态区、代码区、造型区、声音区，其中代码区、造型区、声音区共用一个界面区域，通过分页标签卡进行切换。

伯宏AI 项目 硬件 设置 会员中心 帮助 交流 系统通知:欢迎来到伯宏AI

新建

【项目】菜单，用于文件的保存与打开操作。

打开

【打开】打开一个保存在本地电脑的程序文件；

保存

【保存】保存当前正在编写的程序到本地电脑；

加密保存

【加密保存】与【保存】功能类似，代码仅限

导入扩展模块

编写者打开，此功能需要帐户登录后才能使用。

导出扩展模块

【导入扩展模块】与【导出扩展模块】用于保存自定义的程序块。

硬件 用于连接物联网硬件主机的功能。

设置 用于设置软件界面，切换scratch与arduino编辑界面。

会员中心 登录与注册会员，登录后可以查看与修改会员信息。

交流 包括技术社区、钉钉群、功能介绍及入门教程。

界面的基本操作——认识界面元素 舞台

程序运行的结果最终在舞台区输出，舞台区通过舞台控制条上的按钮进行运行控制。



 切换舞台区显示的大小； 切换舞台区最大与正常；

 运行程序； 暂停程序； 停止程序；

 舞台自动录制动画，当显示红色时，程序运行开始录制，程序停止结束录制，录制格式为GIF动画；

界面的基本操作——认识界面元素 造型(1)

舞台区的每个元素，我们叫做精灵，一个精灵可以有很多种外观，就像一个人可以穿很多衣服一样，这个外观我们叫做造型，精灵的造型其实就是图片，我们可以在造型编辑区添加造型、切换造型、编辑造型和删除造型。



打开绘图界面，绘制一个造型，绘制完成后加入造型。



在网络上查找一个指定的图片，并将找到的图片加入。



从造型素材库中选择一个图片添加为造型。



从你的电脑文件夹中选择一个图片文件添加为造型。

界面的基本操作——认识界面元素 造型(2)

造型的操作包括编辑、删除、复制、发送、合成全部造型为动画、删除造型图片的背景色。



界面的基本操作——认识界面元素 声音

精灵可以包含很多声音素材，包括音乐、讲话声、音效等，这些声音可以在声音编辑区进行添加、录制、预览播放等。



界面的基本操作——认识界面元素 精灵(1)

精灵，也称角色，是由造型（图片）、声音和代码（脚本）共同组成，每个精灵都有自己的造型列表、声音列表和代码，互不干扰。精灵之间通过广播消息的方式来相互通信，根据消息的不同可以运行不同的代码、切换不同的造型、播放不同的声音。



当点击精灵按钮时，代码编辑区、造型编辑区和声音编辑区会同时切换到精灵相对应的代码、造型和声音，同时舞台区的精灵也会切换到最上层。可以通过鼠标右键单击精灵按钮来显示精灵的其他操作功能，如显示和隐藏等。

界面的基本操作——认识界面元素 精灵(2)

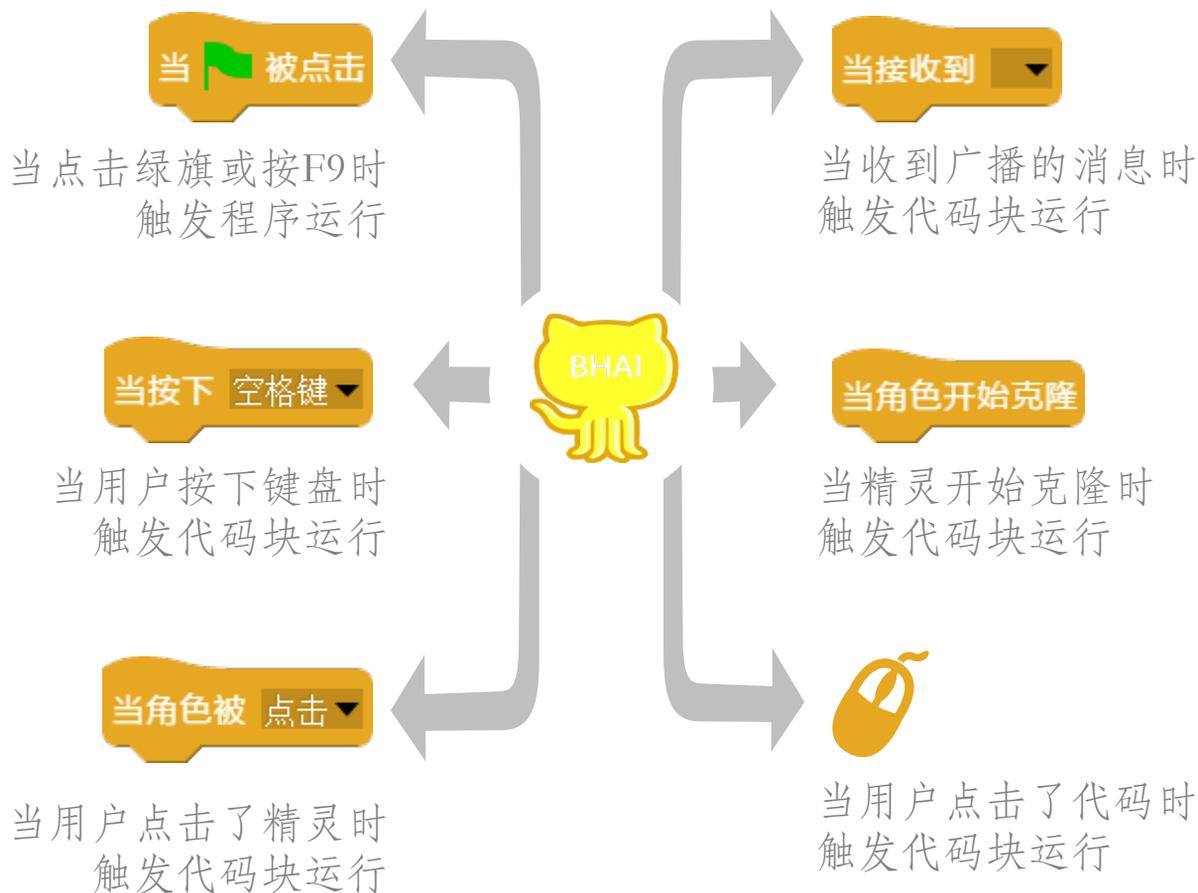
精灵当前的状态，比如位置信息、旋转方式等会显示在精灵状态区，精灵状态区的按钮也可以切换精灵的状态。



精灵状态区可以直接修改精灵的名称，也可以改变精灵的旋转模式（后面会详细讲解），同时还可以锁定精灵的位置不被鼠标拖动。关键在于X和Y的值，我们在后面讲坐标轴时会详细讲解。

程序的基本控制——开始运行(1)

程序的基本控制分为开始运行、暂停运行和结束运行。开始运行有全局运行、局部运行和程序块运行三种。



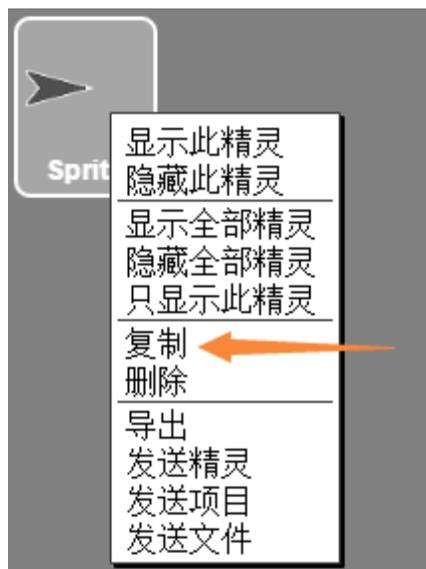
程序的基本控制——开始运行(2)

全局运行指所有精灵的代码全部开始运行，局部运行是指代码块（多个程序块组合成一个代码块）被触发运行，程序块运行是指单个程序块运行。下面我们用示例来演示一下。



1

① 我们在当前精灵的代码区拖入左侧的三个程序块，代码的意思是让当前精灵顺时针旋转。我们可以单击程序块的任意位置，这时代码块开始运行。



2

② 我们在精灵按钮上方鼠标右键单击，弹出快捷菜单，点击【复制】菜单项，此时会添加一个新的精灵按钮。复制的精灵会和原来的精灵一模一样，包括位置、代码、造型和声音，所以在舞台区我们只能看到一个精灵，我们可以在舞台区拖动其中一个精灵到另一个位置。

程序的基本控制——开始运行(3)

全局运行指所有精灵的代码全部开始运行，局部运行是指代码块（多个程序块组合成一个代码块）被触发运行，程序块运行是指单个程序块运行。下面我们用示例来演示一下。



当单击代码块的任意位置开始运行时，我们可以看到被我们点击的那个精灵的箭头一直在旋转，而另一个没有被点击的精灵是静止不动的。我们再次点击代码块时，程序停止了运行。

如果我们点击舞台控制区的绿旗或者按下F9键，我们可以看到两个箭头都在旋转，这时舞台全部的精灵代码都处于运行状态。



我们也可以直接点击程序块  让精灵旋转运行一次。

这个示例程序我们保存在软件里，你可以通过打开项目文件，导入运行，项目的名称为“开始运行”。

程序的基本控制——开始运行(4)

除了可以通过点击绿旗和点击程序块的方式运行代码外，还支持通过其他条件触发代码块的运行，键盘按键、鼠标操作、收到消息、克隆事件等都可以触发代码块的运行。

当按下 空格键

当按下 1

当按下 a

当按下 上移键

当按下某个键触发代码运行，比如按下空格键发射子弹，按下向左键精灵向左移动等，程序能够检测到0-9，a-z以及方向键和空格键的操作。

当角色被 点击

点击
鼠标按下
放下
鼠标进入
鼠标离开

程序还可以检测鼠标的操作：点击、按下、放下、进入和离开。可以根据鼠标的操作来运行相应的代码。

程序中可以通过 **广播 我的消息** 来发送消息给其他精灵，精灵根据消息的内容来运行相对应的代码。消息内容是自己命名新建的，点广播程序块右侧黑色的三角形就可以新建消息内容。你可以打开“触发运行”的项目文件，来体验效果。

程序的基本控制——暂停运行和结束运行

暂停运行是指暂时停止程序的运行，再次运行时精灵所有状态都是延续上次暂停时的状态。结束运行是指结束代码的运行状态，全部恢复到初始状态。精灵的克隆体在暂停时不会被清除，而在结束运行会被全部清除。

新建一个克隆 ▾

删除这个克隆

新建一个 ▾ 的克隆,名字为

删除名字为 的克隆

删除所有克隆

克隆是指复制一个和原精灵一模一样的精灵，和原精灵相比，克隆的精灵不会出现在精灵列表中，在程序结束时克隆体会被全部清除，也可以通过程序块主动删除克隆体，而原精灵只能被隐藏不能被删除。



除了舞台控制区的结束运行，也可以通过以下程序块结束程序：

停止 全部 ▾

停止 这个脚本 ▾

停止 这个程序块 ▾

停止 除此脚本之外的所有脚本 ▾

停止 该角色中的其它脚本 ▾

动画的基本原理——视觉暂留与帧动画

什么是视觉暂留现象？是指人在看到影像后，画面会在脑中保留0.1-0.4秒的时间，如果两个静止画面变换的时间小于0.1秒，人会觉画面是连续的，这个现象叫做视觉暂留。根据这个现象人们发明了帧动画——就是在0.1秒左右的时间切换不同的画面从而形成连续的影像动画，这种通过不同画面的快速切换来形成的动画，我们叫做帧动画。



四张鸽子的图片快速的切换就可以形成，一个连续飞翔的动画。

动画的基本原理——帧动画的运用(1)

通过快速切换精灵的不同造型可以形成帧动画，让精灵动起来，如行走、各种动作、飞行、各种动态效果。



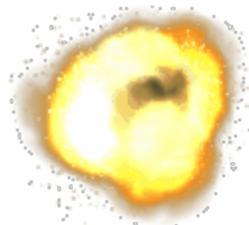
精灵行走



精灵动作



炫光效果



爆炸效果

AIScratch的精灵库有丰富的动画素材，都是帧动画的造型，可以轻松实现精灵动画。配合 **移动 10 步** 程序块就可以实现行走移动或飞行移动了。你可以打开“帧动画”项目文件，运行体验效果。

动画的基本原理——帧动画的运用(2)

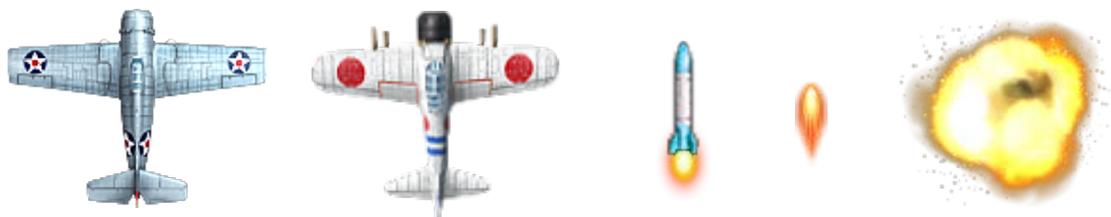
精灵运动是相对的，为了保持精灵运动时始终在舞台中央而不会走出舞台，我们通常移动的是背景。



让精灵在播放切换造型形成行走动画的同时，我们让背景图往相反的方向移动，这样形成更加逼真的动画效果。我们可以打开“精灵行走”的项目文件运行体验效果。

游戏的基本控制——动作控制

精灵的动作可以通过按键、触碰和消息来控制的。通过按键来控制行走、攻击等基本动作，通过触碰来改变精灵基本的状态，如被打、消灭等，通过消息来告知精灵当前的条件和环境状态。下面我们来设计一个飞机大战的游戏。



① 按F4打开精灵库，分别导入以上五个精灵素材。每个精灵都有几个造型可以生成帧动画，同时精灵还包含了相应的音效声音。

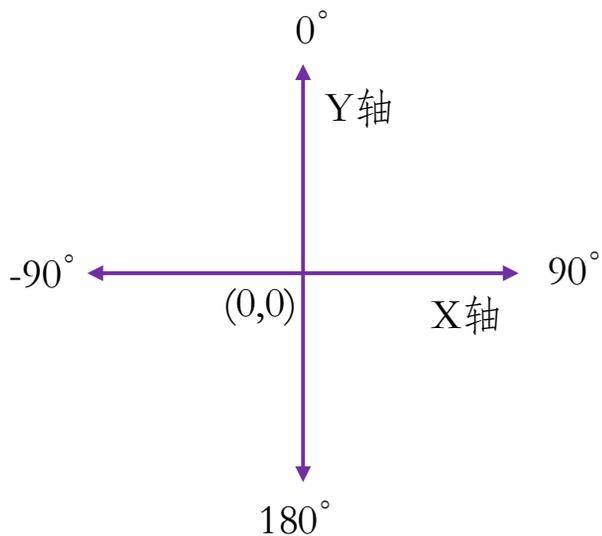


2

② 我们通过条件判断语句程序块来检测用户是否按下了左键或者右键，按下左键飞机向左移动，按下右键飞机向右移动。

游戏的基本控制——舞台坐标

精灵的移动是有方向性的，向左（ -90° ），向右（ 90° ），向上（ 0° ），向下（ 180° ）。舞台区其实存在一个坐标系，坐标就是舞台上每个点的位置信息。



舞台区默认的宽度为480个点，高度为360个点（一个点称为一个像素），在舞台区最中心的点，即X轴与Y轴相交的点的位置我们用坐标方式表示为（0，0），这个点也称为中心点或原点。

从原点（中心点）向左，X的值变小并且为负数，向右X的值变大并且为正数。同理，向上Y值变大为正，向下Y值变小为负。那么我们如何知道精灵当前位置的坐标呢？



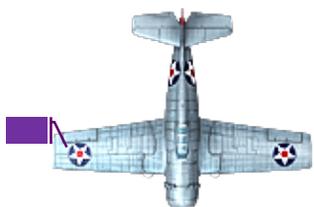
游戏的基本控制——移动方向

精灵的造型默认的方向是面向右边（ 90° ）方向。所以在移动精灵时，除了要改变面向的方向外，也要注意造型图片本身的方向要与面向的方向一致。



面向 90 ▾ 度

默认的方向是面向右边（ 90° ）移动。



面向 -90 ▾ 度

面向左边时，造型本身也会随之旋转。



面向 0 ▾ 度

面向上边时，造型的旋转效果。



面向 180 ▾ 度

面向下边时，造型的旋转效果。

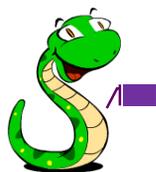
注意：紫色方块为移动方向。

游戏的基本控制——旋转控制

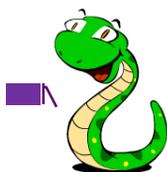
我们很多时候是不希望精灵的造型随移动的方向去旋转，这时我们应该怎么办呢？我们会使用到旋转控制的程序块。

设置旋转模式为 只能水平翻转 ▾

当精灵的旋转模式设置为只能水平翻转时，面向不同方向的效果（紫色方块为移动的方向）：



面向 90 ▾ 度



面向 -90 ▾ 度



面向 0 ▾ 度



面向 180 ▾ 度

设置旋转模式为 不能旋转 ▾

设置旋转模式为 可以旋转 ▾

当精灵的旋转模式设置为不能旋转时，不论面向什么方向移动，精灵的造型不会变化。如果要恢复旋转时，我们可以设置成可以旋转状态。精灵在旋转模式也可以在精灵状态区修改。



总结下要点：

- 1、精灵的移动有方向性，移动前要设置方向；
- 2、设置方向同时精灵也会旋转；
- 3、可以设置精灵的旋转模式；

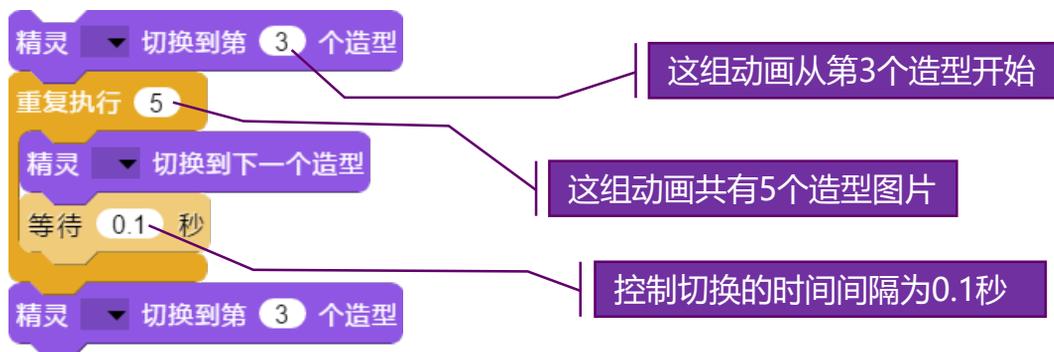
游戏的基本控制——造型切换

我们要形成帧动画，需要不断切换精灵的造型，并且要控制造型切换的时间，正常速度是间隔0.1秒切换一张。



在造型编辑区，我们可以直接点击相应造型图标来切换当前精灵的造型，在程序块代码中，我们可以通过 **切换到造型** 程序块选择造型的名字来切换造型，造型的名称是可以修改的。

播放帧动画就是快速地切换精灵的造型，这时我们并不需要而且也不知道造型的名称，我们可以通过 **精灵 切换到下一个造型** 切换到下一个造型；如果一个精灵包含了多组动画帧造型，这时我们可能要先指定第一个造型再切换到下一个造型，并指定切换的数量。



游戏的基本控制——碰撞检测

在游戏的精灵互动中，最常见的就是精灵的碰撞。碰撞是外界环境发生改变的标志，碰撞引起精灵的反应。



在飞机大战游戏中，战斗机被导弹击中，就是精灵的碰撞过程，如果导弹精灵碰撞到战斗机，战斗机精灵隐藏，爆炸精灵显示，并播放造型动画和爆炸音效。



导弹精灵的代码

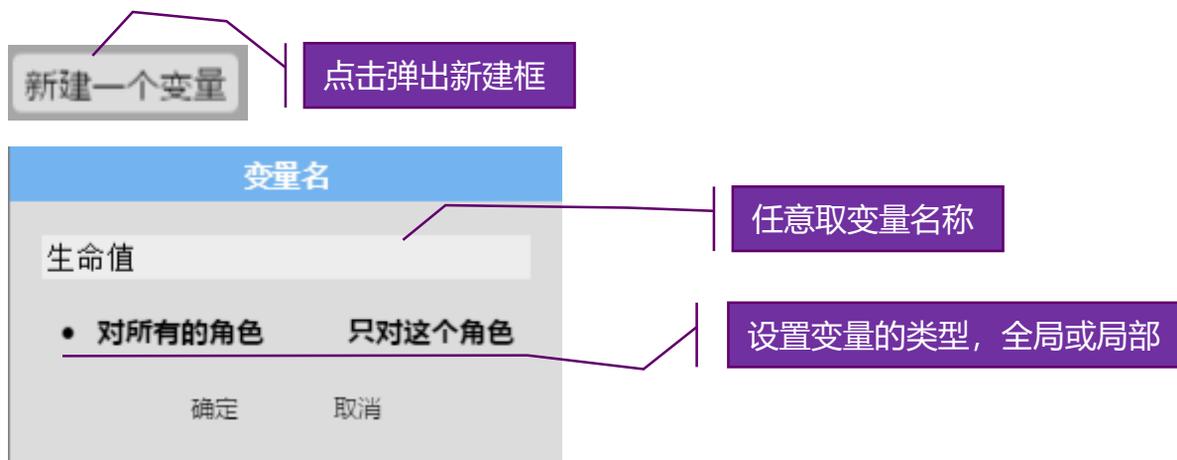


爆炸精灵的代码

游戏的基本控制——计数与变量

在设计游戏时，有很多数据是随游戏进度而变化的，比如生命值、子弹数量、难度系数、关卡等。这些数据被存储在变量中，变量是可随时修改值的量，我们修改变量值的过程，叫做赋值，变量的名称叫做变量名。

我们通过点击程序块分类按钮 **变量** 打开与变量相关的程序块和操作，变量的操作包括新建、赋值、读值和删除。



创建后，会在列表中出现变量程序块，当变量前面的勾号选中时，在舞台上会出现 **生命值 0** 变量值的观察窗口，变量默认值为0。

✓ **生命值**

生命值 0

游戏的基本控制——变量类型

变量类型分成全局变量和局部变量，全局变量对所有精灵可见，任何精灵的程序代码中都可以修改和读取变量的值，局部变量仅对当前创建变量的精灵可见，其他精灵无法修改和读取变量的值。

在不同的精灵间传递内容，除了可以用通过广播消息的方式，也可以通过修改全局变量的值来实现。

将变量 生命值 的值设定为 0

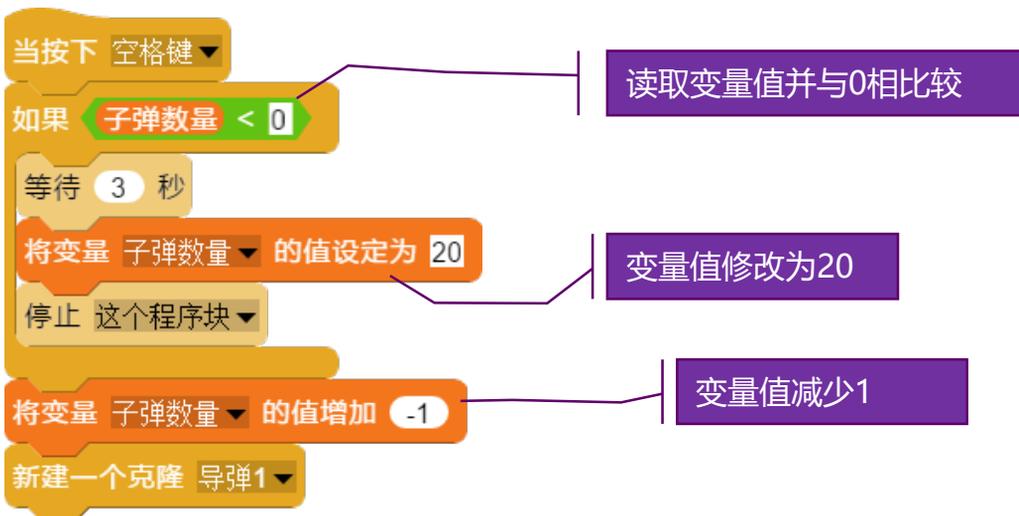
将变量 生命值 的值增加 1

修改变量的值（赋值）

变量也可以在程序代码中动态创建和动态删除。

新建 全局 变量 生命值 初始值 0

删除 全局 变量 生命值



游戏的基本控制——音效与声音

在设计游戏中音效是非常重要的，音效是播放声音来的实现的。每个精灵可以包含多个声音文件，播放声音有两种方式，一种是播放同时继续运行后面的代码，一种是播放完毕后再运行后面的程序代码。

前面我们已经讲过可以通过网络查找、电脑文件、声音素材库和录音话筒的方式获得声音文件，获得以后，我们通过以下程序块来播放声音：

播放声音 喷气战斗机 ▾

播放声音同时继续运行后面的代码

播放声音 喷气战斗机 ▾ 直到播放完毕

播放结束后再运行后续代码

我们还可以根据名称或索引播放其他精灵中的声音文件

播放精灵 导弹1 ▾ 第 1 个声音

播放精灵 爆炸1 ▾ 名为 的声音

在某些时候我们需要检测声音文件的一些状态，比如声音文件的长度和播放状态

精灵 ▾ 第 1 个声音长度(秒)

精灵 ▾ 第 1 个声音播放状态

精灵 ▾ 名为 的声音长度(秒)

精灵 ▾ 名为 的声音播放状态

游戏的基本控制——进度保存和读取

在设计游戏的过程中，我们有时候需要保存游戏的进度，把用户的在游戏过程中产生的变量值保存起来，在下次打开游戏时可以继续原来的进度。

软件提供了两种方式来保存游戏进度：一种是保存到本地文件；一种是保存到网络数据库（云存储）。

保存文本 **生命值** 到文件 `c:\a.txt`

保存单个变量值到文件c:\a.txt中

导入文本文件 `c:\a.txt`

从文件c:\a.txt中读取值并赋值给变量

将变量 **生命值** 的值设定为 **文本文件的内容**

我们也可以通过数组变量来保存和读取所有的变量值，关于数组变量，我们在后面的内容会详细讲解。

保存 **全局** 数组 **游戏进度** 到文件 `c:\a.txt`

从文件 `c:\a.txt` 新建 **全局** 数组 **游戏进度**

游戏的基本控制在“飞机大战”项目的代码中都体现出来，我们可以打开这个项目，学习、模仿和修改代码来体验游戏的效果。

程序块的基本使用——程序块分类(1)

在AIScratch4.0中提供了将近400个程序块指令，数量众多，所以根据程序块的功能进行了分类，除自定义的程序块之外，共分为15大类。

动作	控制
外观	文字
侦测	声音
运算	画笔
变量	列表
界面	网络通讯
大数据	人工智能
物联传感	更多模块

当我们点击程序块分类时，对应的程序块会出现在程序块分类的下方。我们也可以自己定义一个程序块分类，当我们自定义程序块时，如果输入不存在这个分类，则系统会自动新建这个分类。

我们还可以通过【界面】程序块  切换到相应的程序块列表。

除了可以通过点击程序块分类来查找程序块外，我们还可以使用关键字查找的方法找到程序块：按下shift键不放，鼠标左键单击代码编辑区的空白处，当出现闪烁的白色横线时，输入程序块的第一个字，即可以显示找到的程序块



程序块的基本使用——程序块分类(2)

在AIScratch4.0中提供了将近400个程序块指令，数量众多，所以根据程序块的功能进行了分类，除自定义的程序块之外，共分为15大类。

动作	控制
外观	文字
侦测	声音
运算	画笔
变量	列表
界面	网络通讯
大数据	人工智能
物联传感	更多模块

当我们点击程序块分类时，对应的程序块会出现在程序块分类的下方。我们也可以自己定义一个程序块分类，当我们自定义程序块时，如果输入不存在这个分类，则系统会自动新建这个分类。

我们还可以通过【界面】程序块  切换到相应的程序块列表。

除了可以通过点击程序块分类来查找程序块外，我们还可以使用关键字查找的方法找到程序块：按下shift键不放，鼠标左键单击代码编辑区的空白处，当出现闪烁的白色横线时，输入程序块的第一个字，即可以显示找到的程序块



程序块的基本使用——运行控制

控制程序块主要是用于控制程序的运行、条件分支运行、停止运行、调用子程序等运行流程，包含程序的开始、等待、暂停、结束、分支、调用、克隆等指令。

当  被点击

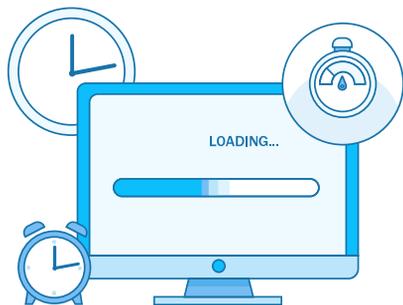
当按下  空格键

当角色被  点击

当接收到  消息

当角色开始克隆

开始运行的程序块都是带帽子的形状，每个开始运行的程序块都代表一种触发运行的条件，我们可以通过打开“触发运行”项目文件，来体验各种开始程序块的功能。



程序等待是指暂停当前运行的程序，等待运行条件满足后继续运行，等待的程序块并不会影响其他代码块的运行。

等待  1 秒

强行等待一秒钟后继续运行后续程序块。

直到  前都等待

当多边形框中的条件不成立时一直等待。

广播  并等待

在其他代码处理这条消息完成前一直等待。

程序块的基本使用——获取帮助

我们对每个程序块都提供了帮助信息和示范的案例，打开帮助和示范的案例就可以学习如何使用这个程序块。



我们在程序块的上方，单击鼠标右键弹出快捷菜单，点击【帮助】，显示出该程序块相应的帮助信息。



同样我们在程序块的上方，单击鼠标右键弹出快捷菜单后点击【项目演示】，则会自动打开示例的项目程序并运行。

程序块的基本使用——循环程序块

循环程序块主要用于代码需要重复运行的场景，包含无条件循环、次数循环和条件循环三个程序块。循环块可以相互嵌套，形成多重循环。我们可以打开项目演示运行示例。



①

①无条件重复执行循环体内的程序块，一般用于需要一直执行或一直检测的代码中。



②

②重复执行循环体内的程序块几次，执行次数到达后退出循环体。



③

③一直重复执行循环体内的程序块，直到多边形条件框内的条件满足后退出。



程序块的基本使用——条件分支程序块

条件分支程序块主要用于根据不同条件执行不同代码的场合，条件分支程序块分成单分支和双分支。条件分支可以嵌套以形成多个分支运行。



1

①如果条件框内条件成立，则执行条件分支块里的程序代码，不成立则直接跳到下一个程序块运行。



如果碰到边缘则移动到 (0, 0) 坐标位置。



2

②如果条件框内条件成立，则执行第一个条件分支块里的程序代码，否则执行第二个程序块代码。



如果碰到左边缘则移动的方向面向右边，否则移动的方向面向左边。

程序块的基本使用——程序控制的综合案例(1)

我们来看一个综合案例，一只蜜蜂在舞台上飞来飞去，撞到墙后头冒金星的动画。这里使用到循环块、条件分支块。



需要循环的部分：

- 1、重复播放飞行的声音；
- 2、重复翅膀飞行的造型切换；
- 3、重复向左或向右移动的动作；
- 4、撞墙后重复切换金星造型动画；

条件分支：

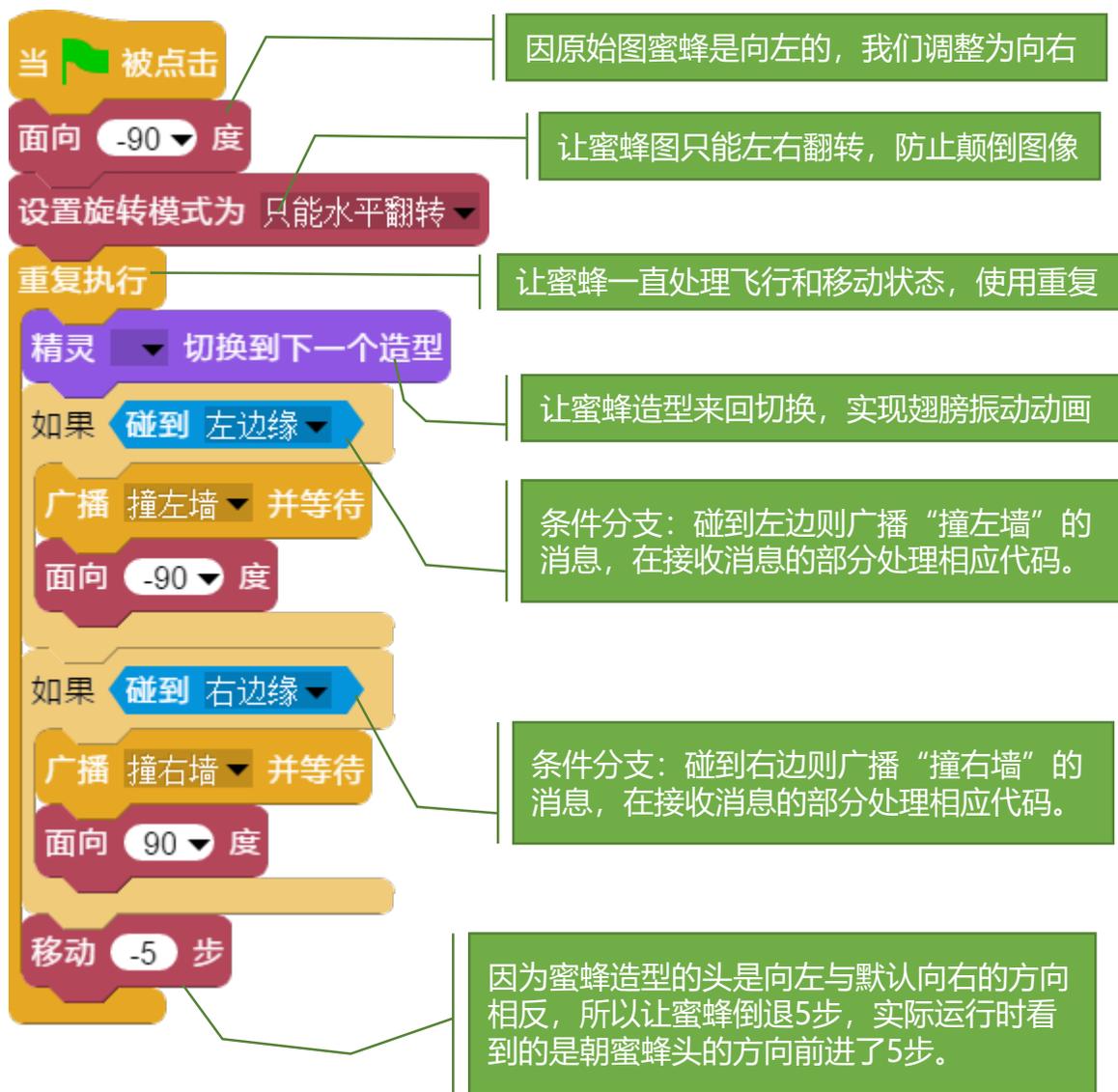
当撞到左墙后要播放撞墙的声音和动画，我们可以广播一个消息“撞左墙”来处理撞墙的程序代码；同理也可以处理撞到右墙时的情况；

我们可以通过打开



程序块的项目演示来体验这个案例的运行效果。

程序块的基本使用——程序控制的综合案例(2)



蜜蜂精灵的代码块说明

程序块的基本使用——停止运行

程序的停止可以是全部停止或局部停止。我们在《程序的基本控制》章节已经学习了停止的方法，现在重点介绍下各停止程序块的含义。

停止 全部 ▾

停止项目所有的代码运行，等同于按F10

停止 这个脚本 ▾

停止这个精灵所有的代码运行。

停止 这个程序块 ▾

停止这个代码块的运行。

上述程序块是代码终止块，在它下面无法继续添加程序块。

停止 除此脚本之外的所有脚本 ▾

上面程序块只运行当前精灵的当前代码，当前精灵的其他代码将停止运行，并且其他精灵的代码也全部停止。

停止 该角色中的其它脚本 ▾

上面的程序继续运行当前精灵的当前代码，当前精灵的其他代码停止运行，其他精灵的代码不受此影响。

程序块的基本使用——精灵克隆

精灵克隆是指将指定精灵复制一份，克隆体的造型、声音、代码、位置和大小与母体一模一样。通常我们通过其他精灵克隆当前精灵，以免造成无限循环的克隆。当无限循环克隆发生后，软件会耗尽全部资源导致程序卡顿甚至卡死。克隆体的数量，取决于电脑内存大小，所以尽可能删除无用的克隆体。

当角色开始克隆

当其他精灵开始克隆这个角色时，自动运行这个代码，原则上所有需要克隆的精灵都用这个程序块开始来写代码的，以保证克隆体的代码可以自动运行。

新建一个克隆

以指定精灵为母体，克隆一个新的精灵，精灵可以是自己也可以是其他精灵。新建的克隆体没有名字，在舞台停止运行时也会被全部删除。

新建一个 的克隆,名字为

新建一个带名字的克隆体。

删除这个克隆

删除名字为 的克隆

删除所有克隆

删除克隆体可以用以上三种程序块，用法大体相同。

程序块的基本使用——过程与函数

在程序代码中，有很多代码其实是相同的，我们之前用循环来完成重复性的代码，除此之外我们还有其他方法来实现代码的重复使用，这个就是过程和函数。过程是将一段长代码用一个名字来定义，以后要执行时只需要运行这个名字就可以了，而函数比较相似，只不过它除了运行重用的代码外，还会返回一个结果给调用它的代码。



过程和函数的定义是通过  来实现的，函数定义后也可以把它赋值给一个变量，在后面的代码通过变量取得这个过程或函数。

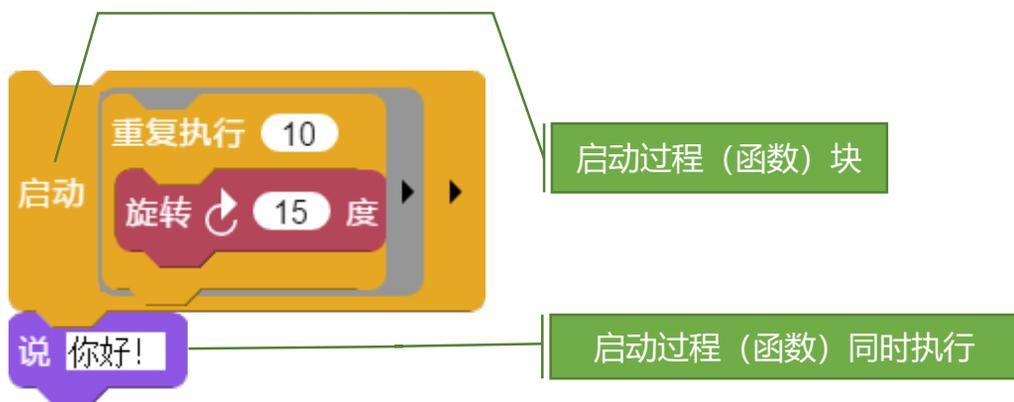
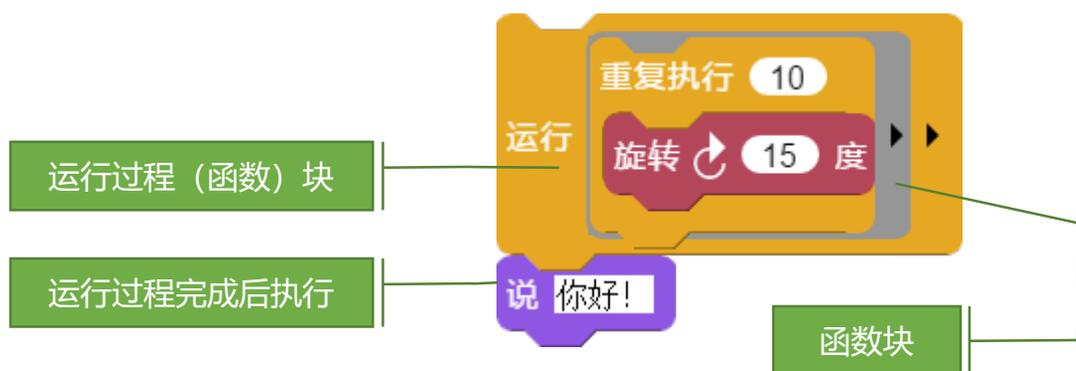


过程和函数与后续讲的子程序块相似，也可以带参数。

程序块的基本使用——过程（函数）的调用



通过这个程序块可以调用程序块过程或函数。



运行程序块与启动程序块功能相似，都可以调用过程(函数)块，但是启动程序块是用一个独立的代码调用过程（函数），所以它后继的程序代码会同时执行。

程序块的基本使用——移动动作

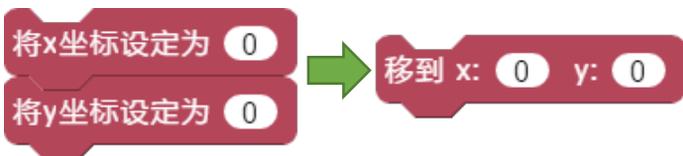
精灵在舞台上的动作主要是移动和旋转。我们前面讲过坐标轴的概念，移动其实就是精灵在坐标轴上位置的变化。

移动 10 步

精灵在舞台位置指定的方向（默认为正右方）移动指定的距离，默认为10个点。

精灵 移动 10 步

舞台默认的大小是宽480个点，高360个点，舞台坐标轴的原点在舞台中央，所以舞台的左边缘的X值是-240，右边缘X值是240。

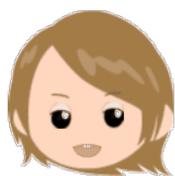


上面的左边的程序块作用等同于右边程序块的作用。

程序块的基本使用——跟随动作

精灵的跟随动作，就是一个精灵保持与另一个精灵的相对位置不变，当被跟随的精灵移动时，当前精灵也跟随移动。

左侧的图像是由三个精灵组成，分别是：



当头部移动时，眼睛和嘴巴是跟随移动的。

跟随精灵 头部 ▾ 移动

解除 眼睛 ▾ 的跟随移动状态

当前精灵跟随指定精灵

处于跟随状态的精灵会随被跟随的精灵一起移动和旋转，但在旋转时为了保持相对位置不变，还应该注意旋转中心位置的设定。除了跟随移动外，还有跟随缩放，当前精灵会与跟随的精灵一起放大和缩小，跟随缩放程序块在外观分类中。

跟随精灵 头部 ▾ 缩放

解除 眼睛 ▾ 的跟随缩放状态

程序块的基本使用——旋转动作与旋转中心

精灵的旋转动作，就是一个精灵围绕旋转中心点进行角色旋转的动作。默认的旋转中心是精灵造型图片的中心点。



不同的旋转中心的旋转的效果是完全不同的，我们可以通过编辑造型来设置造型图片的旋转中心位置，也可以通过程序块来指定造型图片的旋转中心的坐标。



在造型编辑区点击编辑按钮，打开画图对话框，点击旋转中心工具按钮，在图片上方拖动图标中心到旋转中心位置，点击确认。

设置精灵 猪 第 1 个造型旋转中心X: 0 Y: 0

通过【外观】程序块分类中的设置旋转中心程序块来设置指定精灵指定造型的旋转中心坐标，坐标轴的原点位置是图片的左上角，往右是正向X轴，往下是正向Y轴。

程序块的基本使用——外观·造型切换

精灵所有与造型相关的程序块都在外观分类中。造型的操作分成显示隐藏、切换、缩放、特效、层次和图形处理。



显示此精灵
隐藏此精灵
显示全部精灵
隐藏全部精灵
只显示此精灵

显示和隐藏程序块可以在代码中动态显示和隐藏精灵。在精灵操作区，鼠标右键单击精灵按钮也可以弹出相应的显示和隐藏菜单。

切换到造型

当前精灵切换到指定名称的造型。

精灵 切换到第 1 个造型

指定精灵切换到第N个造型，不指定则默认为当前精灵。

精灵 切换到上一个造型

精灵 切换到下一个造型

指定精灵切换到上一个造型，如果当前造型为第1个造型时，则切换到最后一个造型。如果造型是顺序排列的，重复切换到上一造型可以实现动画的倒播效果。精灵不指定则默认为当前精灵。切换到下一个造型的效果类型，如果当前造型是最后一个造型时，切换到第一造型。造型连续切换之间通常会等0.1秒左右。

程序块的基本使用——外观·造型缩放

造型的图片可以放大或缩小。在软件中支持两种图片类型，一种是位图，通常是PNG格式，一种是矢量图，通常是SVG格式。SVG矢量格式需要电脑系统支持HTML5，因此部分较旧的操作系统（或浏览器）不支持显示，如window xp系统。

将角色的大小设定为 200

将角色的大小设定为 50

这个是百分比，
200意思是200%，
即2倍大小，50即
50%，0.5倍大小。

将角色的大小增加 10

放大10%，减号表示缩小



位图放大后的效果



矢量放大后的效果

位图类型的造型在放大后会变模糊，格式为JPG、PNG、GIF的图片都是位图格式。GIF如果是动态格式，则软件会自动将每一帧分解成一个造型。

矢量类型其实是通过命令画出来的图，因此不管图片放大多少都不会变模糊，目前软件系统支持SVG格式的图片。

程序块的基本使用——外观·造型特效

造型的特效又称滤镜，是在图片上加上特殊的显示效果，系统支持亮度、鬼影、负片、漫画、复制、五彩纸屑六种特效。



将 亮度 ▾ 特效增加 50



将 鬼影 ▾ 特效增加 50



将 负片 ▾ 特效增加 50



将 漫画 ▾ 特效增加 50



将 复制 ▾ 特效增加 3



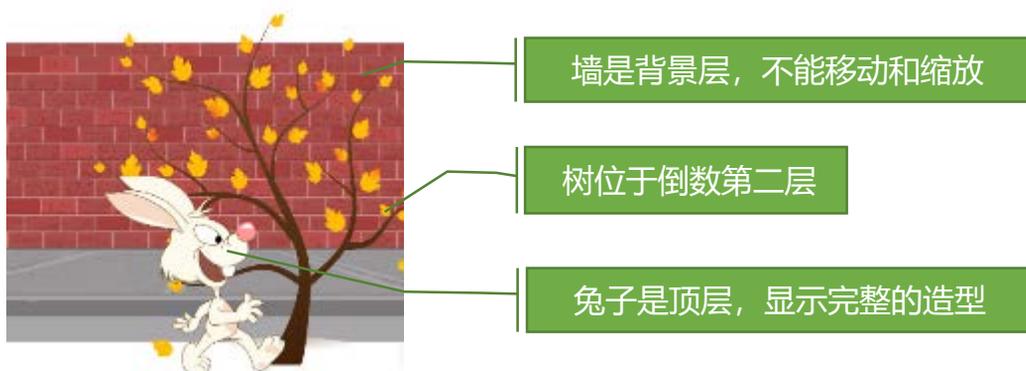
将 五彩纸屑 ▾ 特效增加 50



清除所有图形特效

程序块的基本使用——外观·造型层次

精灵在舞台上叠放的，因此上层的精灵会遮挡下一层的精灵，可以通过造型的层次来调整显示的效果。



移至最上层

下移 1 层

通过移至最上层可以让精灵造型完整显示而不被遮挡，舞台的背景层是最低层，背景层可以更换造型，但是不能移动和缩放大小，因此对于需要移动的背景图片，我们一般是添加成为一个精灵造型来处理，并把它放到最底层（下移999层）。

程序块的基本使用——外观·对话与思考

精灵在舞台上要显示文字对话，通常是通过说和思考程序块来完成的。如果要在舞台显示文字，则使用文字造型来实现。



说 睡觉是最幸福的事!

思考 嗯...让我想想

说和思考的作用一样都是显示对白，只是显示时的外框不同，说的外框显示的是对话形式的，而思考的外框显示的内心思考形式，它们都显示在造型图片的右上方。另外，说和思考都有另一种带时间的程序块，程序块中的数字是秒钟，在显示对白文字几秒后消失，如果不带秒钟数字，则显示的对白文字会一直显示，直到下一次使用这个程序块。

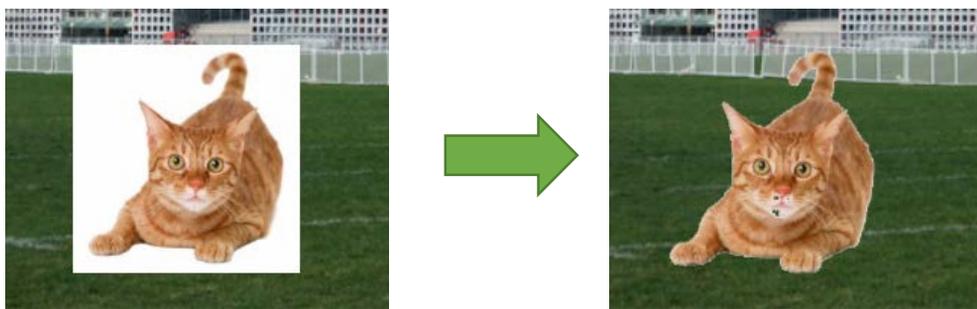
文字造型 文字造型, 字体 微软雅黑 大小 20 颜色 ■ 字型 加粗 ▼

文字造型

在很多时候要显示文字时，我们可以使用文字造型来生成造型图片，也可以在代码中动态生成文字。

程序块的基本使用——外观·图形处理

造型的图片经常要使用工具进行适当处理，以达到理想的效果，最主要的处理就是去除背景色和透明边缘。



去除背景色,容差值 50 创建新造型 是

我们经常得到的图片是带背景色的，很难溶入舞台的场景中，因此我们需要使用去除背景色的程序块。容差值是指背景色的范围，数值越大范围越大，创建新造型是指去除背景色后的图片是否添加成为一个新的造型，如果选否，会直接修改当前造型图片，而且无法撤消，使用时注意区别。

我们得到的背景透明的图片，有时候透明的边缘会非常大，这时我们要去除多余的透明边缘，可以使用下面的程序块。

精灵 第 个造型去除透明边缘

程序块的基本使用——文字显示(1)

在舞台上经常要显示和处理文字，系统提供了丰富的文字显示功能，包括设置字体、大小、颜色、边框、阴影和字形等。



文字程序块支持动态生成各种样式的文字，通过动态生成或文字不同的造型可以创建出文字的样式动画。

生成的文字支持两种阴影：清晰和模糊，清晰的阴影支持六个方向的生成；

文字字型包括正常、加粗和倾斜；

文字颜色除了点鼠标选取外，也支持十六进制的颜色值设置，以达到精确适配。

设置文字颜色值 #000

程序块的基本使用——文字显示(2)

针对文本的一些常见的处理，系统提供了强大的文字处理功能，包括转换、替换、截取、拼接、分隔、查找和判断等。

把文本 1.2 转换成数值,保留 0 小数

截取文本 小猫咪 ,从第 2 个开始,长度 1

把文本 小猫 中的 猫 替换成 狗

删除文本 小猫 两边的空格

将 你好|世界 ◀▶ 拼接

拆分 你好|世界 ,以 | 作为分隔符

返回列表,可赋值给列表变量

取 1 个字符,从字符串 世界 中

世界 的长度

小猫咪 包含 猫

文本 小猫咪 中第一次出现 猫 的位置

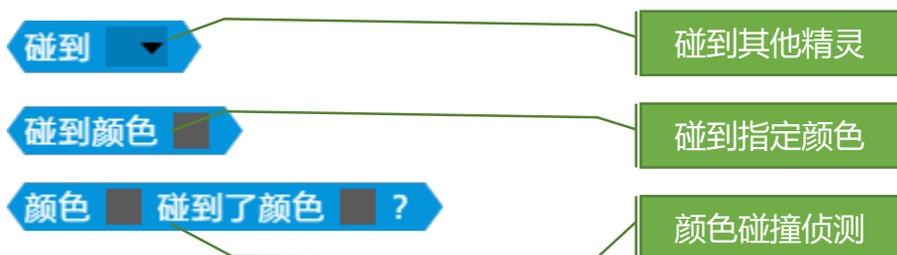
文本 小猫咪 最后一次出现 猫 的位置

字符 a 的Unicode编码值

Unicode编码值为 65 的字符

程序块的基本使用——侦测碰撞与按键

侦测主要用于获取一些环境参数和条件，以便程序做出相应的动作，包括碰撞侦测、位置侦测、按键侦测、时间侦测等。



鼠标的x坐标

鼠标的y坐标

按下鼠标?

侦测鼠标的位置和按键，
可以配合 **当角色被 点击**
使用。

当按下 空格键

可侦测按键，但与 **按键 是否按下?**

不同之处在于，前者是按下后松开才会被检测到，而后者是按下即使不松开也能被检测到。因此持续按键检测应使用后者来检测。

询问 你的名字? 并等待

用于获取用户的按键输入，以得到文本，输入的结果在 **回答** 中，我们在后面会讲到用户界面的设计，还可使用弹出对话框来实现获取输入。

程序块的基本使用——侦测其他

侦测主要用于获取一些环境参数和条件，以便程序做出相应的动作，包括碰撞侦测、位置侦测、按键侦测、时间侦测等。

到 鼠标指针 ▾ 的距离

侦测到鼠标指针或其他精灵的距离

计时器归零

计时器

时间侦测：计时器。

当前 年 ▾

当前 日 ▾

当前 日 ▾

当前 小时 ▾

当前 分钟 ▾

当前 秒 ▾

取得当前的年月日时分秒和毫秒，星期（从星期日开始）。

当前 毫秒数 ▾

当前 一周中第几天 ▾

我们通过示例来演示取得时间的方法，我们鼠标右击 **当前 日 ▾** 在弹出的菜单中点击“项目演示”打开示例代码。

重复执行

如果

当前 小时 ▾ 的长度 = 2

精灵 小时1 ▾ 切换到第

取 1 个字符，从字符串 当前 小时 ▾ 中 + 1 个造型

精灵 小时2 ▾ 切换到第

取 2 个字符，从字符串 当前 小时 ▾ 中 + 1 个造型

否则

精灵 小时1 ▾ 切换到第 1 个造型

精灵 小时2 ▾ 切换到第 当前 小时 ▾ + 1 个造型



程序块的基本使用——播放声音

前面我们已经讲过每个精灵都可以包含自己的声音，每个声音有自己名称，声音的播放可以根据名称，也可以根据序号。

播放声音

播放当前精灵中指定名称的声音，播放开始后，会继续运行下一个程序块。

播放声音

直到播放完毕

与上面的程序块功能相似，但在播放完指定声音之前一直等待。

播放声音

播放精灵 名为 的声音

播放精灵 名为 的声音到结束

停止播放精灵 名为 的声音

播放精灵 第 个声音

播放第 个声音直到播完

播放精灵 第 个声音到结束

停止播放精灵 第 个声音

上面左侧四个程序块是根据声音的名称来播放和停止，右侧四个程序是根据声音的序号来播放和停止。播放的声音都是从头开始，暂时无法从指定的时间播放或在指定时间停止，但可以检测声音播放的状态：没有播放、播放中、播放结束；

播放声音 生日快乐

重复执行直到

精灵 Sprite 名为 生日快乐 的声音播放状态 = 播放结束

精灵 Sprite 切换到下一个造型

程序块的基本使用——播放音符

除了可以播放声音外，系统还支持播放指定音频的声音，我们可以通过指定音频及时长来演奏音乐。

1=C 2/4

1 2 3 1 | 1 2 3 1 | 3 4 5 | 3 4 5 | 5 6 5 4

两只老虎，两只老虎，跑得快，跑得快。一只没有

3 1 | 5 6 5 4 3 1 | 2 5 1 | 2 5 1 ||

耳朵一只没有尾巴，真奇怪，真奇怪。

The image shows four Scratch 'Repeat' blocks, each set to repeat 2 times. Each block contains 'Play Note' blocks corresponding to the musical notation above:

- Block 1:** 4 'Play Note' blocks with notes 60, 62, 64, and 60, each with a duration of 0.5 beats.
- Block 2:** 3 'Play Note' blocks with notes 64, 65, and 67, with durations of 0.5, 0.5, and 1 beat respectively.
- Block 3:** 7 'Play Note' blocks with notes 67, 69, 67, 65, 64, and 60, with durations of 0.25, 0.25, 0.25, 0.25, 0.5, and 0.5 beats respectively.
- Block 4:** 3 'Play Note' blocks with notes 60, 55, and 60, with durations of 0.5, 0.5, and 1 beat respectively.

我们可以通过弹奏程序块的项目演示来欣赏这首《两只老虎》。

程序块的基本使用——数学运算与函数

基本的数学运算：加减乘除；比较运算：大于、等于、小于、不等于；逻辑运算：且、或、非；各类数学函数；

    四则运算 加减乘除

  求余数   取整，并对小数四舍五入

   三角函数

  取整函数

 绝对值函数

 平方根函数

  对数函数

  幂函数

    比较运算

     逻辑运算

  类似比较函数

 JavaScript自定义函数

程序块的基本使用——画笔与画布

通常我们的精灵造型表现为一张图片，但有时候我们也可以定义为一支画笔，用画笔可以画出各种图形。



我们在精灵造型编辑区，选中箭头造型后，当前精灵舞台上的造型会变成  的样式，此时我们可以使用画笔的程序块来绘画（其他造型下也可以绘画）。

绘画的图形与精灵造型是独立的，绘画的图形画在画布上，画布是介于背景与精灵造型之间的一个图层，在背景之上，在造型之下，在画布上绘出的图形无法移动和缩放，而且只能通过

清除所有画笔

删除，因为所有的画笔画的图形都是在画布上，所以删除时只能清除所有画笔，在新的绘画前要清除下画笔。

画笔动作包括落笔、抬笔和绘画三个，分别对应的程序块为：

落笔

抬笔

移动

10

步

而且移动（画线）的方向是由动作中控制移动方向的几个程序块来实现的。下面我们用一个示例来看下画笔的具体运用，在“落笔”程序块的上方点击鼠标右键，在弹出的对话框点击“项目演示”，打开演示项目，项目的代码如下：

程序块的基本使用——图章

在画布上可以使用一种特殊的绘画方式：图章。图章顾名思义，就是给画布盖个章，章子的图案就是精灵当前的造型。

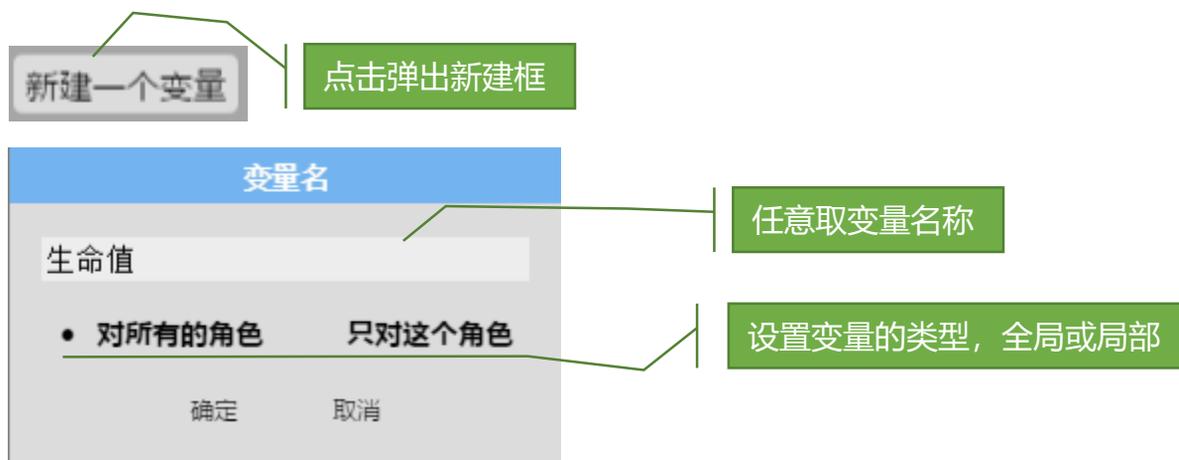


在上述代码中，我们绘制了一个六边形，六边形的粗细和颜色都是随机生成的，在画线的同时我们还使用了“图章”程序块，“图章”程序块的作用是将当前造型图片在画笔所在的位置复制一份到画布上，执行一次复制一个。

至此，我们知道了画笔的属性包括：颜色、粗细，画笔的动作包括：落笔、抬笔、画线，图章就是复制造型到画布的工具，画笔的动作不影响图章的使用。

程序块的基本使用——动态创建变量

在《游戏基本控制》章节我们讲解了变量的基本应用。这里我们来讲解一下变量的动态创建。



上述创建变量的方法，我们称为静态创建。我们如果想在代码执行的过程中创建一个自己命名的变量，我们称为动态创建。下面我们通过示例来了解下动态创建的过程。

保存文本 生命值 到文件 c:\1.txt

导入文本文件 c:\1.txt

新建 全局 变量 文本文件的内容 初始值 100

说 取得 局部 变量 文本文件的内容 的值 2 秒

设置 全局 变量 文本文件的内容 值为 10

说 取得 局部 变量 文本文件的内容 的值 2 秒

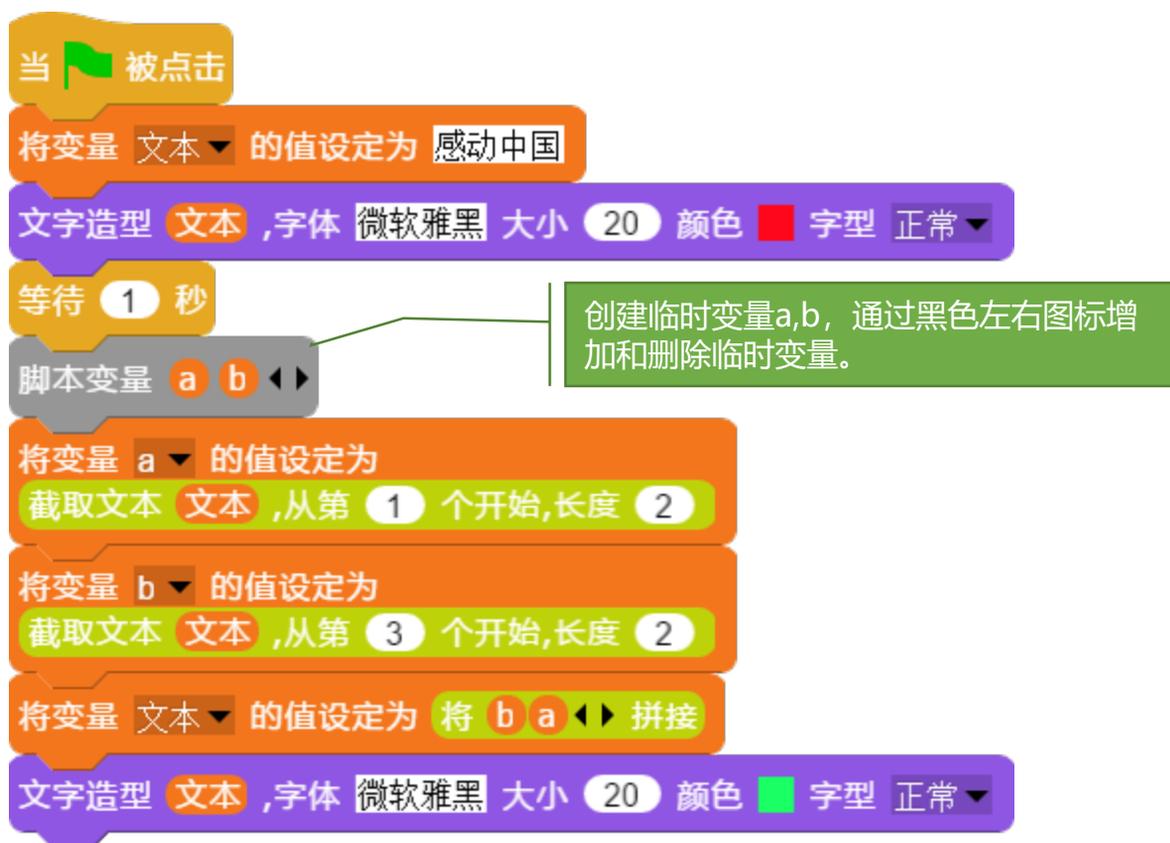
删除 局部 变量 文本文件的内容

说 取得 局部 变量 文本文件的内容 的值 2 秒

在这个示例中，我们先把变量名保存到一个文件，读取出文件，并把文件内容（“生命值”）做为变量名，在使用变量的过程中，我们都没有使用到变量名称。

程序块的基本使用——临时变量

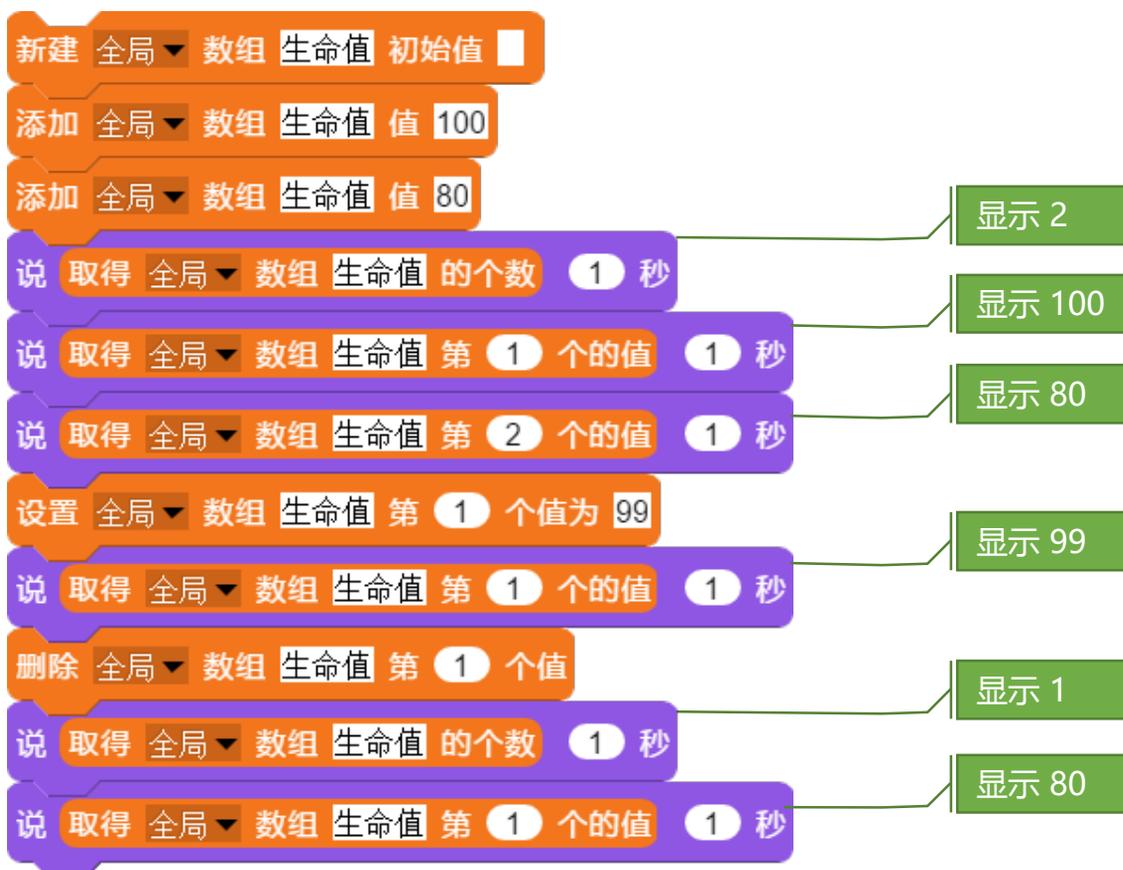
创建的临时变量不出现在变量列表中，无法显示和隐藏，但是可以像正常变量一样使用，可以同时创建多个临时变量。



上述代码中，我们的任务是把“感动中国”四个字的前两个字和后两个字的顺序互换，我们先创建了两个临时变量a和b，截取文本的前两个字赋值给a，截取文本后两个字给b，然后重新拼接b和a，并把拼接结果重新赋值给文本显示出来。

程序块的基本使用——数组变量

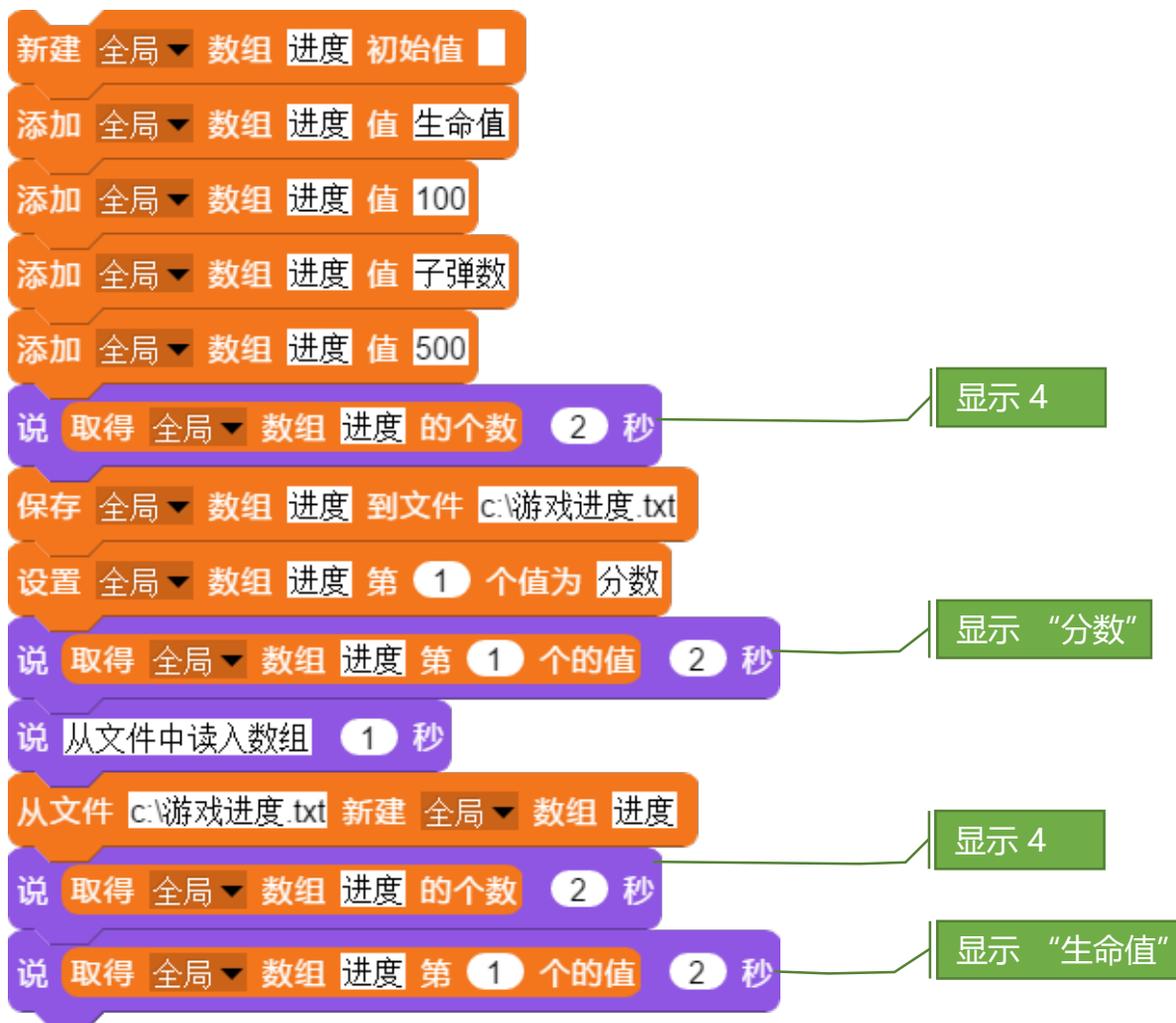
什么叫做数组？数组就是一组数字或文字组成的序列，可以看做是数字或文字的集合。在实际使用中，数组可以被赋予任意类型的值，比如一组数字、一组文字、一组代码、一组声音等等，都是被允许的。存储数组的变量，称为数组变量。



上述代码中，我们先是新建了一个名为“生命值”的数组变量，并添加了两个数据元素100和80，这时数组的个数为2，我们把第一个元素的值修改为99，最后我们删除了第一个元素，第二个元素变成第一个元素。

程序块的基本使用——数组变量与文件

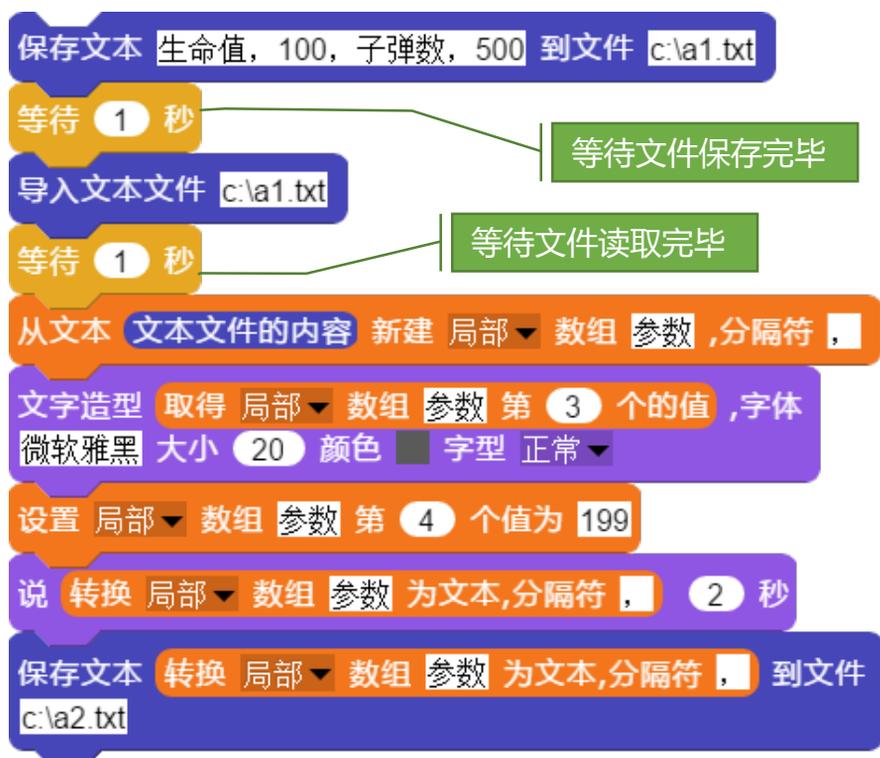
数组变量可以进行基本的文件操作：从文件读取数组值、保存数组值到文件中。



上述代码中，我们创建一个名为“进度”的数组变量，并把数组值保存到文件“C:\游戏进度.txt”中，随后我们修改了第一个元素的值，我们从文件中重新读入进度值，再次显示第一个元素的值。

程序块的基本使用——数组变量与文本

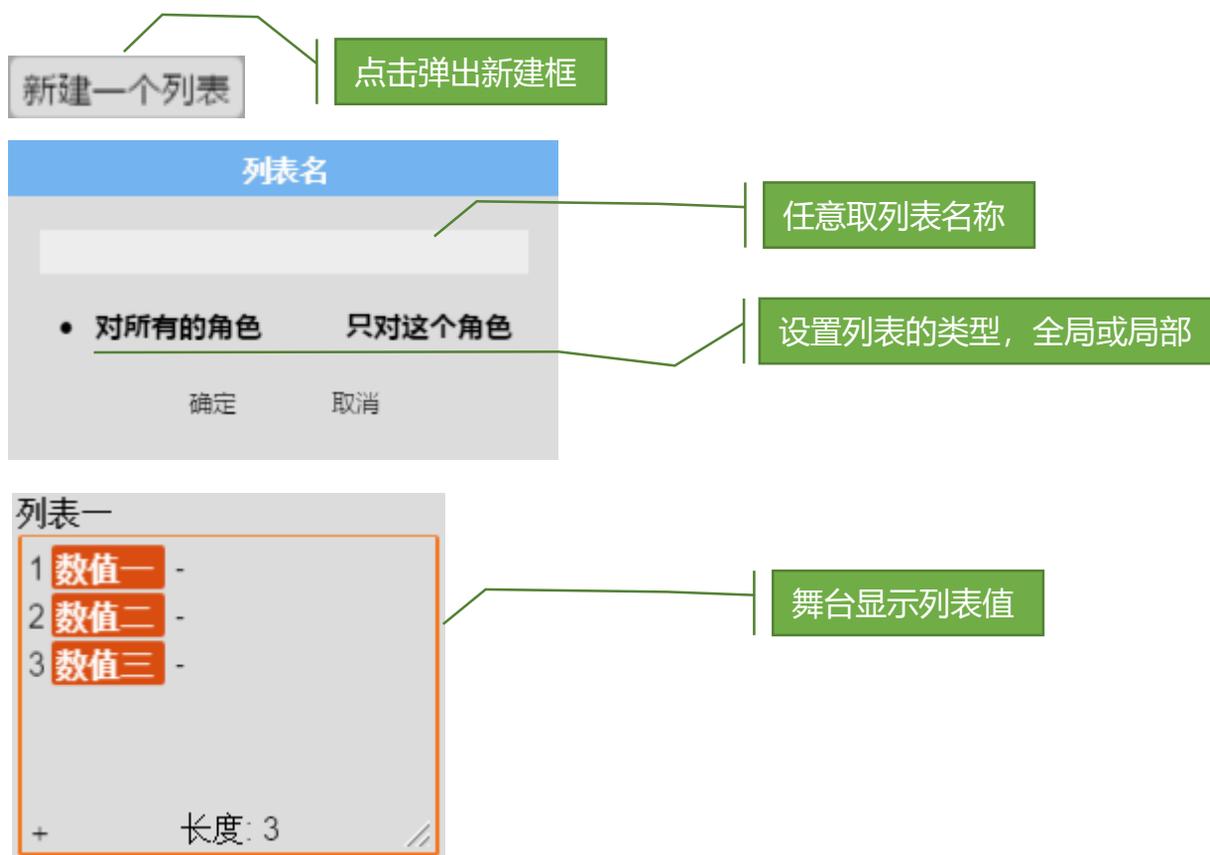
数组变量与文本之间可以相互转换，数组转成文本，可以指定每个值之间的分隔符。数组转换成文件后可以保存为文件。



上述代码中，我们先保存了一个文本到文件c:\a1.txt中，然后读出文件内容到文本中，通过文本转数组程序块创建了一个数组变量“参数”，我们显示数组中第三个元素的值，同时修改了第四个元素的值为199，重新把数组变量转成文本，并把文本保存到c:\a2.txt文件中。

程序块的基本使用——数组变量与列表

列表与数组变量相似，我们可以把数组变量看成是动态创建的列表，或者把列表看成是静态生成的数组。



我们创建的列表能够在舞台显示和隐藏，列表的操作与数组变量的操作类似，这里就不复述。

程序块的基本使用——数组与列表的遍历

遍历是指取从数组（或列表）的第一个值读取到最后一个值的过程。



等同



我们在【列表】程序块分类可以找到遍历程序块，遍历相当一个循环，从列表（或数组）第一个位置一直到最后一个位置，列表（或数组）的每个值可以通过变量“值”来获得。

下面的代码含义是说出所有精灵的名称，“所有精灵列表”是一个包含所有精灵的对象列表变量。



程序块的基本使用——变量知识总结

变量的作用是来存储数据的，包括单个变量、数组（列表）变量、对象变量，那么如何理解这些变量的区别呢？

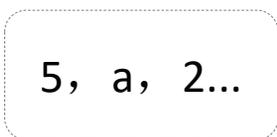
A=5



变量A

变量相当于一个有名称（变量名）的盒子，盒子可以装任何东西（变量值），单个变量只能装一个，而数组变量可以装很多。单个变量可以通过变量名直接取得里面的值，而数组（列表）变量要用变量名加上它的位置（索引）才能取得变量的值。

A=[5,a,2···]



数组变量A

将变量名称和变量值一起存储到一个变量中，这个变量就称为对象变量。在对象变量中取值既可以通过位置（索引）取得，也可以通过里面的变量名取得。

A={A:5,B:'F',C:[5,2,1]}

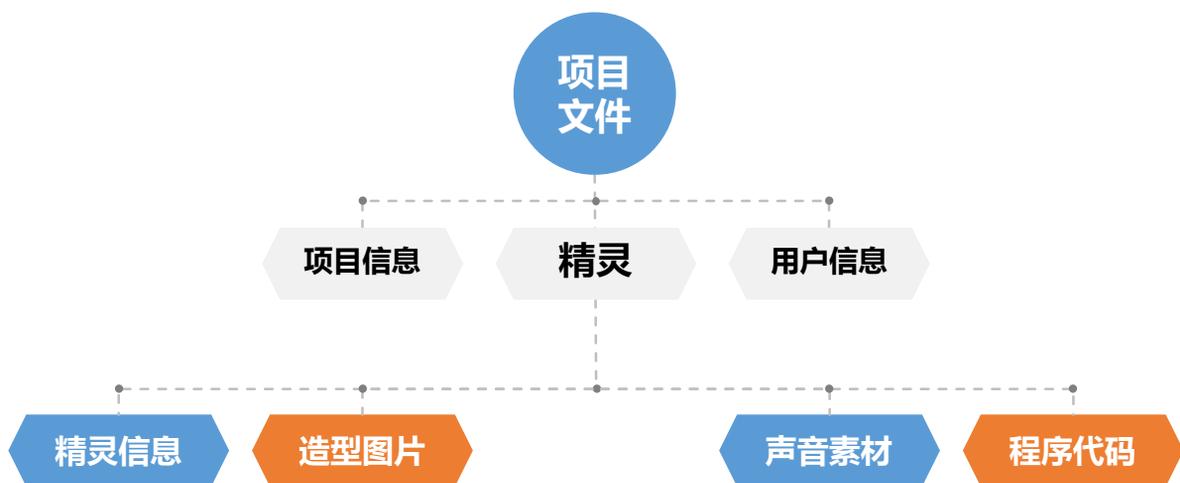


对象变量A

程序块的基本使用——文件导入导出

我们在实现操作中经常要保存信息到文件中，方便在下次使用，这就涉及到文件的操作。文件是组织电脑信息的一种形式，在PC电脑中所有信息都是以文件形式保存起来的。

在我们的编程过程中，我们需要保存的信息包括：项目、精灵、造型、声音、进度信息、用户输入、程序结果等等。



通常我们通过项目来保存程序的所有信息（运行结果和进度信息除外），但很多时候我们也需要单独保存一些信息或者一些中间产生的结果，系统也提供更多的文件功能，实现这些操作。文件操作分成用户鼠标点击执行和程序块代码执行两种方式，用户鼠标点击的方式我们这里就不在复述，用户自己可以在软件界面里点击体验效果。我们更多来了解程序块代码的实现方式。

程序块的基本使用——文件程序块

项目中所有的素材都可以用快捷菜单导入和导出，同时系统也提供了用程序块实现导入导出的方式。

导入项目文件 c:\1.xml

导入并运行项目文件 c:\1.xml

保存项目文件 c:\1.txt

使用上述代码可以导入项目和保存项目，在导入项目前一定要保存当前项目，否则将丢失当前项目。

导入图片为精灵 c:\1.png

导入精灵文件 c:\1.sprite

保存精灵 到文件 c:\1.sprite

使用上述代码可以导入精灵和保存精灵。

导入声音文件 c:\1.mp3 到精灵

保存精灵 第 1 个声音到文件 c:\1.mp3

使用上述代码可以导入声音和保存声音。

导入图片造型 c:\1.png 到精灵

保存精灵 第 1 个造型到文件 c:\1.png

使用上述代码可以导入造型和保存造型。

导入文本文件 c:\1.txt

文本文件的内容

保存文本 文本内容 到文件 c:\1.txt

使用上述代码可以导入文本文件和保存文本到文件。

上述都是从本机电脑导入和保存文件，其实所有的导入程序块的文件来源还支持网络文件，形如 <http://baidu.com/bac.jpg> 的网络地址文件也是可以导入到系统中的。

进阶者使用教程

01

素材的使用和管理

02

舞台与界面的控制

03

人工智能的应用

04

自定义程序块

05

网络通讯编程

06

智能硬件编程

素材的使用与管理——精灵素材

AIScratch的素材包括精灵、造型和声音，其中精灵素材同时包含造型图片和声音素材。

在系统中，精灵的素材都包含在素材库里，系统中有标准素材库、动漫素材库、教育素材库和自定义素材库。标准素材库含有通用的一些常用素材，动漫素材主要用于动漫制作和游戏制作，教育素材包含了中小学教材中常用的教学素材，自定义素材库主要用于用户自己创建个性化的素材库。



创建素材库



打开标准库



打开动漫库



小学教育库



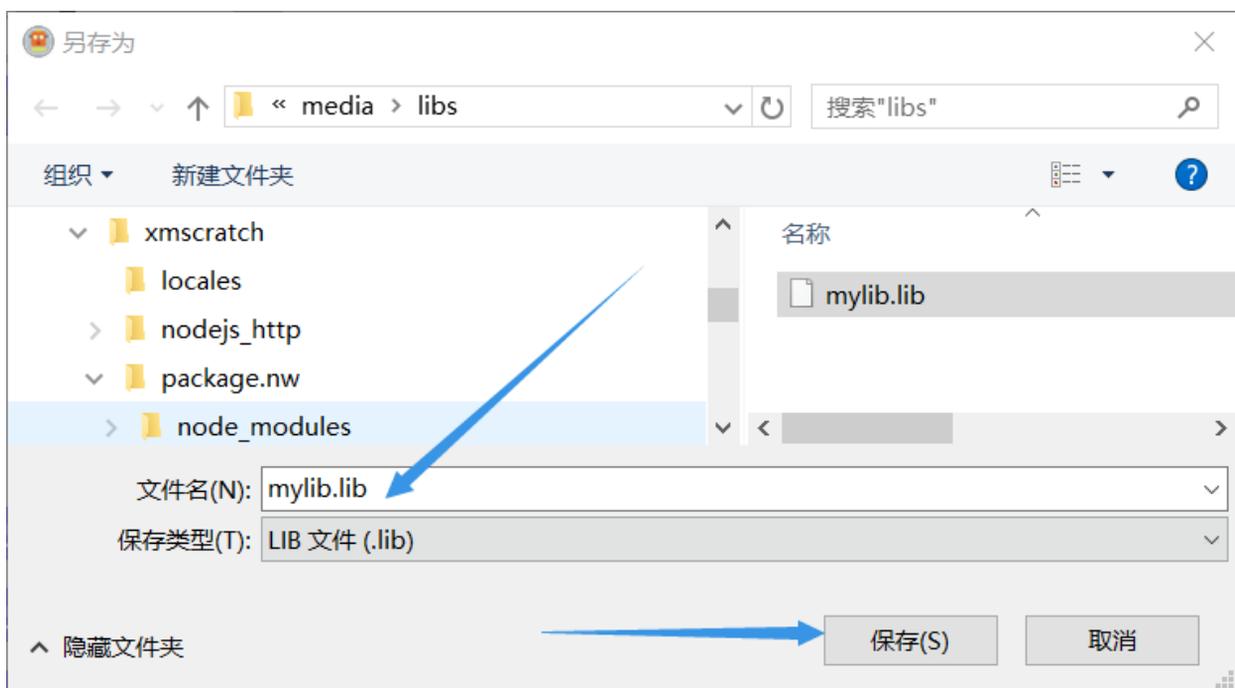
个人收藏库

下面我们来创建一个新的素材库，并且给它添加分类和精灵的造型素材和声音素材。我们点击上面“创建素材库”的按钮，弹出新建素材库的对话框，我们输入自己想的名称，单击确定。

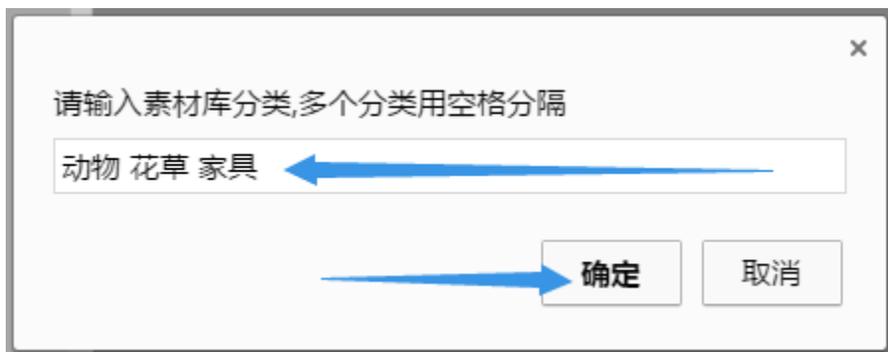


素材的使用与管理——创建素材库(1)

点确定后弹出保存对话框，我们输入文件名并点击保存（注意不要改变保存路径）。

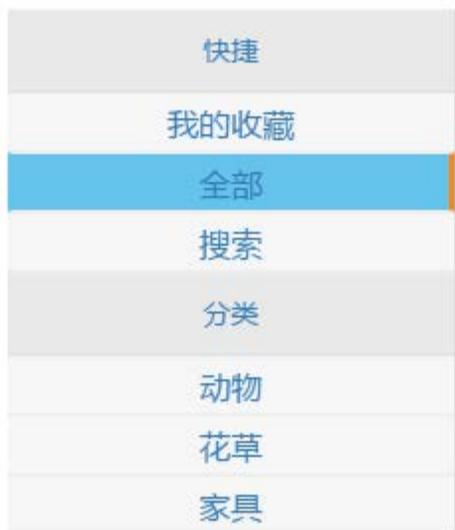


保存后在弹出的对话框输入分类名称，每个分类用空格分开，创建的素材库后续不支持分类的管理，因此要事先做好分类的规划。



素材的使用与管理——创建素材库(2)

我的素材



创建完成后会自动打开素材库，我们可以看到素材库已经有了一个默认的素材，在对话框的左侧就是素材分类，有两个大的分类项，分别是“快捷”和“分类”，在快捷分类中系统已经内置了三个分类，分别是“我的收藏”、“全部”、“搜索”，这三个分类用于快速找到素材。

那么如何添加自己的素材到素材库中呢，首先我们要制作好一个精灵，设置好精灵的名称、造型和声音，打开要添加的素材库，并且点击要添加的分类，点击素材库对话框左下角的加入按钮 则当前精灵立即被添加到素材库中，在下次打开后精灵素材就可以重复使用了。如果对素材库中的素材不满意我们还可以使用 来删除选中的精灵素材，同时对经常使用的素材进行收藏，收藏素材可以使用 按钮。

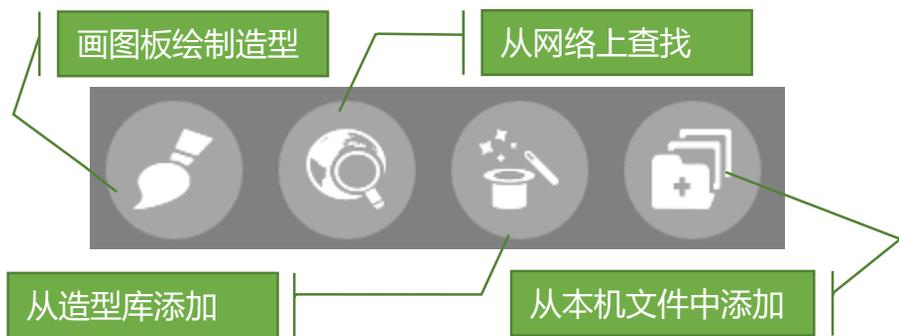
下面我们详细演示一下添加精灵的过程：

素材的使用与管理——添加精灵到素材库(1)



①在精灵状态区修改精灵的名称。这个将作为在素材库中素材的名称。

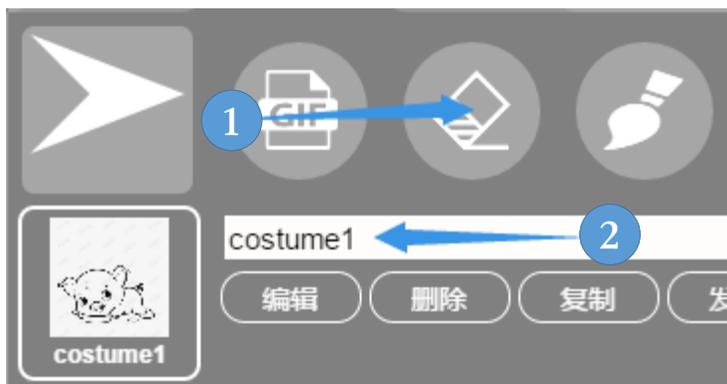
接下来我们添加精灵的造型，添加精灵造型有四种方法：1、从素材库中添加；2、用关键字从网络上随机查找；3、用画图板绘制；4、从本机电脑文件中添加；因为我们要添加到素材中，所以



我们不从素材库中添加，我们从网络上查找。我们点击从网络上查找造型的按钮，弹出查找对话框，输入查找关键字，如下图：



素材的使用与管理——添加精灵到素材库(2)



点击确定后，系统会将从网络上查找到的图片直接添加在造型列表中，但是有时候我们查到的图片是有背景色的。

我们可以通过第二个工具按钮（见①）来删除图片的背景色，点击按钮后会把删除背景色的图片添加到造型列表中，如果我们对删除背景色的结果满意，可以直接删除原图，删除后我们可以修改造型的名称（见②），这样我们完成了添加造型的步骤。

接下来，我们添加精灵的声音，同样的声音添加也可以从本机电脑、网络上和素材库添加，我们这里从网络上查找添加：



在弹出的查找对话框中输入查找关键字，查找的结果会自动添加到声音列表中。我们点击声音上的播放按钮试听声音效果，满意后修改声音素材的名称，完成添加声音素材的步骤。



素材的使用与管理——添加精灵到素材库(3)



分别点击上述按钮，找到我们刚才创建的自定义素材库打开，点击左侧的分类按钮，这是我们添加精灵的分类，再点击左下角

加入舞台精灵到库

按钮，此时我们看到素材已经添加进去了。



至此，我们已经完成了添加精灵到自定义素材库的任务。

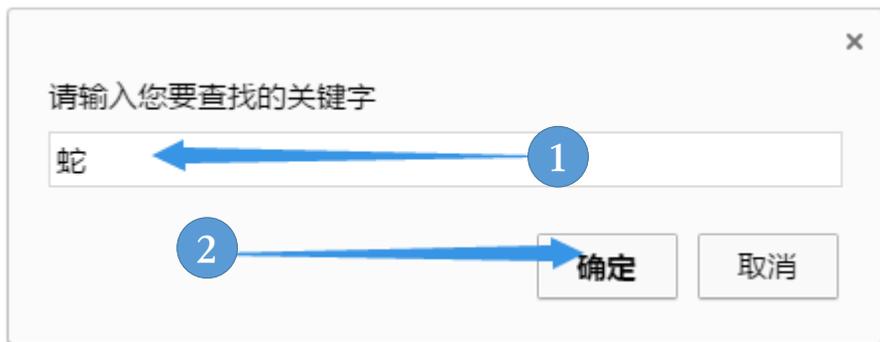
素材的使用与管理——查找与收藏

AIScratch的素材库包含了大量的素材，有时候仅靠分类难以找到，这时系统提供了查找和收藏素材的功能。



我们在素材库中选中要收藏的精灵，点击“收藏选中”的按钮。这时选中的精灵就添加到“我的收藏”分类中了（注意：登录会员与未登录会员的“我的收藏”内容是不同的）。点击左侧“我的收藏”分类查看结果。

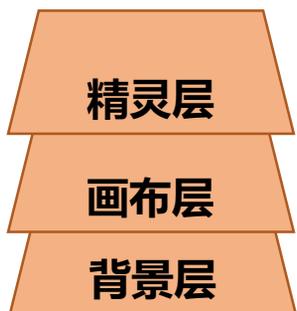
下面我们查找素材库的素材，点击左侧“搜索”分类，弹出搜索对话框，我们输入查找的关键字，点击确定，这时在右侧出现所有与关键字相关的素材，选中后点击确定就可以了。



舞台与界面的控制——舞台层次与尺寸

AIScratch舞台区是代码的运行结果，因此理解与控制舞台的显示非常重要。

一、舞台分层



舞台是分层显示的，位于上层的图形将位于下层的图形遮挡。从上往下分别是精灵层、画布层和背景层。其中精灵层的精灵也是分层显示。我们可以通过【外观】中的下移层程序块来修改精灵的层级。



二、舞台尺寸



舞台的大小是用像素点来计算的，舞台默认的宽度是480px（px表示像素点），高度为360px。舞台的大小是可以修改的。当舞台尺寸最大化时与屏幕的尺寸相同。



舞台与界面的控制——舞台缩放与全屏

舞台的控制分为：分层、修改尺寸、缩放、全屏、快照和录屏等操作，满足程序对舞台的基本操作。

三、舞台缩放



舞台缩放是为了显示的需要，放大显示或缩小显示，并不会真正改变舞台的尺寸大小，我们可以通过舞台右侧的缩放手柄来改变缩放比例，也可以通过舞台控制按钮缩放。

四、舞台全屏

在很多时候我们需要屏幕只显示舞台内容，其余内容全部隐藏，这时我们需要进入舞台的全屏模式，在全屏模式下，如果舞台尺寸小于屏幕尺寸时，舞台会自动缩放大小，以适应全屏显示，如果精灵缩放显示相差较大，显示时会比较模糊。因此在设计全屏的应用程序时，我们也应该将舞台尺寸最大化，以减少模糊情况的发生。在舞台全屏状态，我们可以按F10来退出全屏。

舞台进入全屏

舞台退出全屏

设置舞台最大尺寸

舞台与界面的控制——舞台快照与录屏

舞台快照是指对舞台显示内容的截屏操作，录屏是指对舞台显示的动画录制成GIF格式的动画。

五、舞台快照

舞台快照把当前舞台显示内容截图后添加到当前精灵的造型列表中，我们可以通过系统的导出功能或使用导出造型程序块来保存当前的快照，甚至可以保存数据库中。

舞台快照 名称

保存精灵 第 个造型到文件

六、舞台录屏

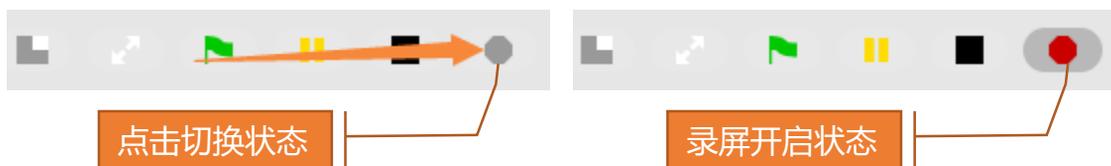
舞台录屏有两种方式：一是手动录屏；二是自动录屏。手动录屏是通过按F6功能键开始或结束录屏，结束录屏后会弹出GIF设置帧率对话框（帧率概率我们在前面动画章节讲过，就是每分钟显示图片的数量），默认为10帧/秒。



舞台与界面的控制——舞台刷新频率

设置好帧率后弹出保存文件对话框，我们选择好保存路径，点击确定后保存。

第二种是自动录屏，处于自动录屏状态时，点击绿旗或按F9运行程序时，系统自动开始录屏，当点击停止方块或按F10停止程序时，系统自动结束屏（这种录屏方式不适合无限循环的程序代码）。



系统自带的舞台录屏，只能录制舞台区，其他区域是无法录制的，而且因为保存格式是图片格式，所以存在无法保存声音和文件尺寸较大的缺陷。如果遇到长时间录制或者需要录制声音的情况，强烈建议使用第三方专业的录屏软件，比如Screen Recorder。

七、舞台刷新频率

舞台默认的刷新频率是60帧/秒，如果快速运动的精灵会出现卡顿现象，我们就要修改成更高的刷新频率。如果电脑系统配置很低可以修改成低一些的刷新频率。

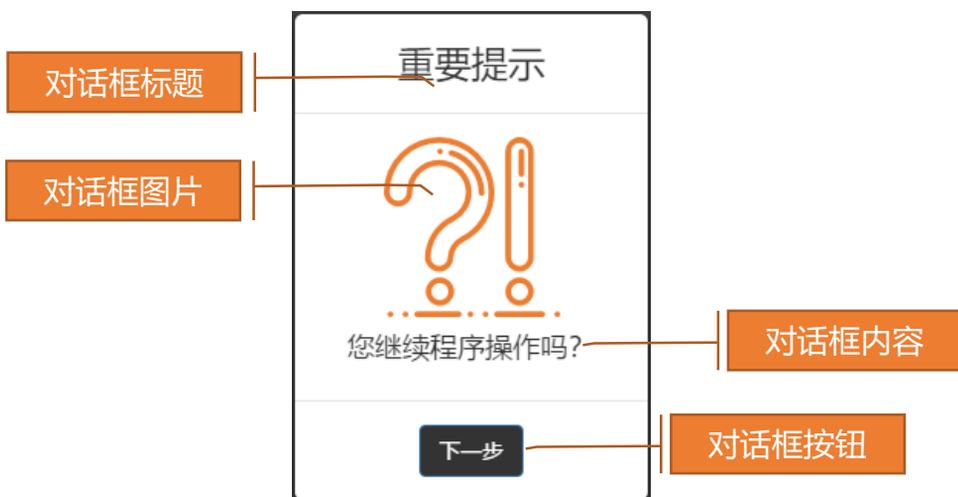
设置舞台刷新频率 60 帧/秒

舞台与界面的控制——消息对话框(1)

对话框是用弹出窗口的方式来提示用户信息和用户输入，是用户互动操作的一种方式。

1

显示对话框,标题 **重要提示** 内容 **您继续程序操作吗?** 按钮名称 **下一步** 图片 **c:\1.png**



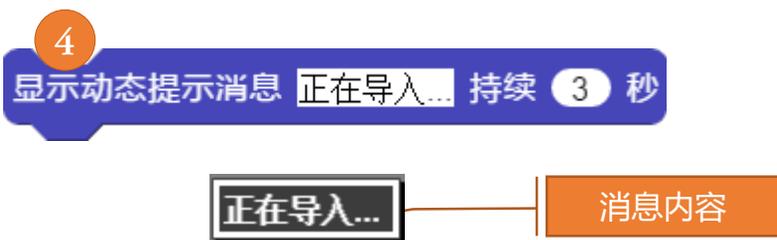
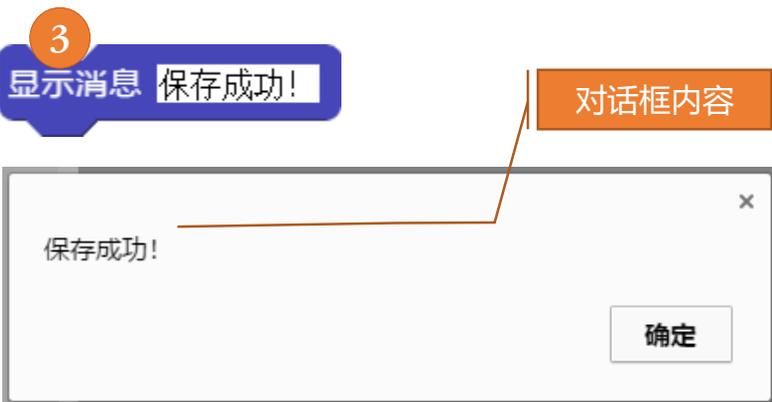
2

显示选择对话框,标题 **提示** 内容 **你确定吗?** 按钮1名 **确定** 按钮2名 **取消**

选择对话框用户选择的结果



舞台与界面的控制——消息对话框(2)



人工智能的应用——什么是人工智能

人工智能，英文简称AI，简单来说就是让机器拥有人类能力和智慧的一种技术。比如模仿人类的运动、视觉和听觉等。AIScratch支持人工智能的一些基本应用，同时也支持人工智能功能的扩展。

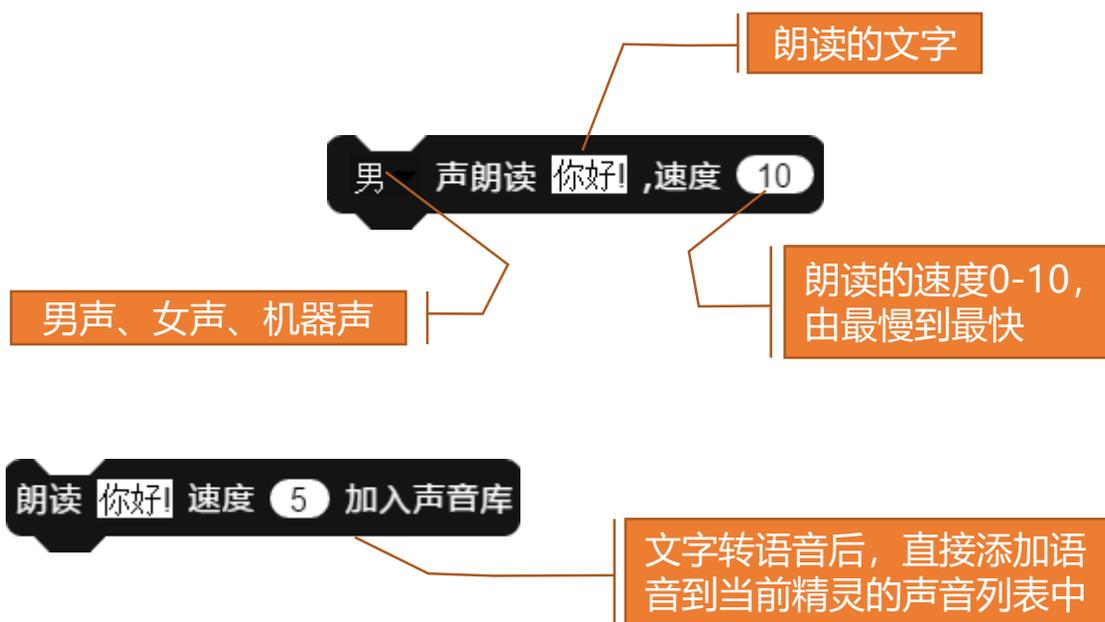


人工智能图谱

AIScratch提供了人工智能视觉、听觉、大数据和机器硬件传感等多种功能模块，这些模块都需要您在连接网络情况下实现。

人工智能的应用——朗读文字

朗读文字就是系统将文字转换成语音的过程。转换的结果还可以直接保存在精灵的声音列表中。



使用人工智能模块都需要在连接外网后才能运行，我们可以鼠标右键单击朗读程序块，在弹出的菜单中点击【项目演示】来听一听机器朗读的效果。如果需要朗读英语，我们还可以在此之前将文字翻译成英语后再朗读。



人工智能的应用——语音识别

语音识别就是让程序听懂人话，语音通过电脑的话筒录音或者声音文件提供给网络服务器，识别后返回结果（文字）。

我们先通过系统的录音功能来录制一段语音到精灵的声音库中，然后通过语音识别程序块来转换成文字，切换到声音编辑区后我们点击录音按钮①，在弹出的录音框中点话筒图标②开始录音，再次点击③结束录音，这时我们看到声音列表中新增了一个名为“recoding”的声音文件。



识别声音 recoding 为文字

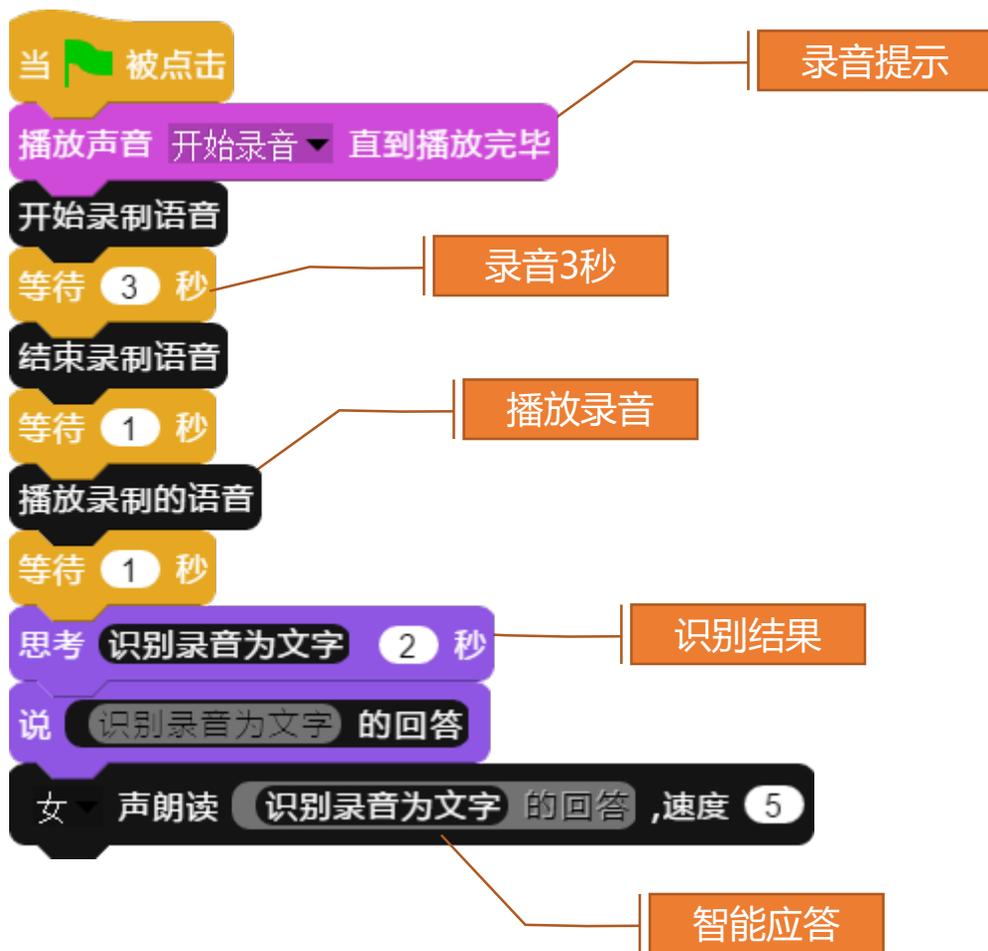
识别声音文字结果

使用左侧的识别声音程序块，就可以识别刚才录制好的语音，通过“识别声音文字结果”来获取转换好的文字。

系统还提供了快捷键来操作录音的过程，按F12开始录音，再次按F12结束录音，结束录音后系统自动识别文字，并将结果保存在“识别声音文字结果”的程序块和系统的粘贴板之中，供我们后面调用。

人工智能的应用——语音识别

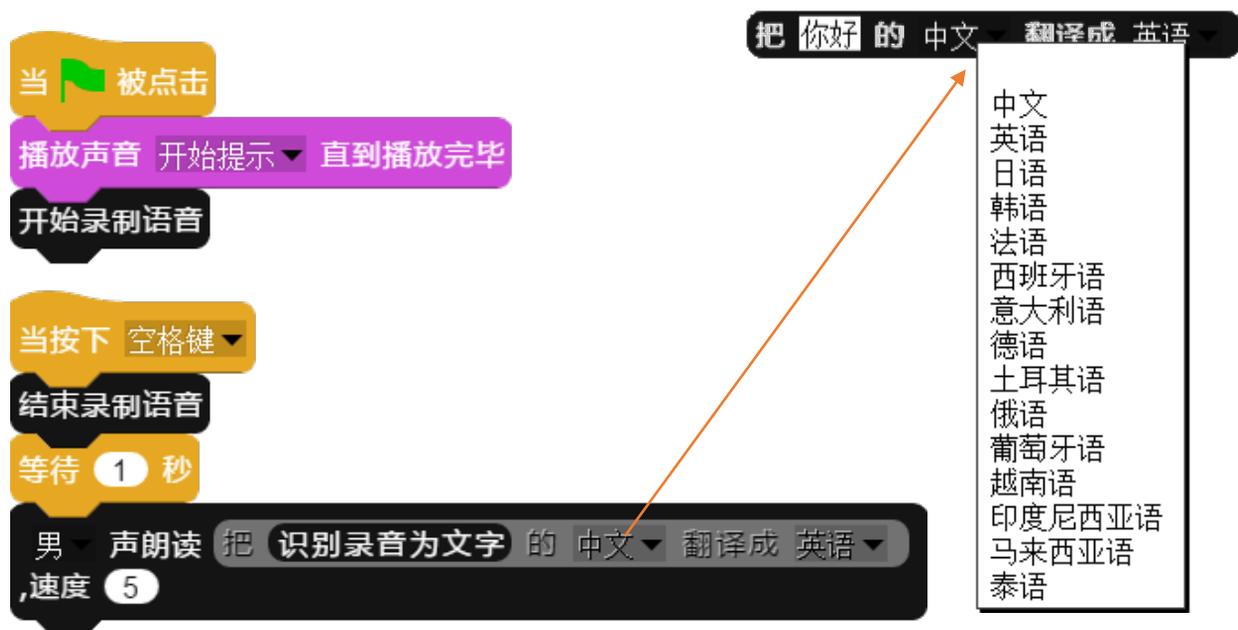
我们还可以通过程序块代码方式来控制录音的过程。通过识别录音的内容来做相应的程序处理，实现程序的一些语音交互功能。



很多语音指令系统就是这样完成的，AIScratch4.0中也内置了一个简单语音指令系统，我们可以用语音来指挥系统做一些简单的事情，使用这个语音系统只需要按F8键就可以开始。

人工智能的应用——文字翻译

文字翻译是将各个国家的文字相互转换的工具，系统目前支持15种语言文字的相互转换。



上述代码实现了一个简单的英语翻译机的功能：程序开始时录音，按空格键停止录音，把录音转换成文字后翻译成英文，并把翻译结果用语音朗读出来。

人工智能的应用——图文识别

图文识别功能包括识别图片上的文字和图片上的内容两种。图片上的文字又有印刷体和手写体，系统都支持识别。

识别当前图(或链接)文字

识别当前图(或链接)手写文字

文字识别结果

我们可以在识别程序块中直接填入图片的网址链接，也可以把图片导入成为精灵的造型后识别，由于图片上传需要时间，因此在识别程序块后，应该等待几秒钟的时间。最后识别的文字结果会保存在“文字识别结果”程序块中。



内容识别就是认出图片的物体，并返回名称。内容识别包括：通用识别、动物识别、植物识别、水果识别、车型识别、车牌识别、菜品识别。最后系统还提供了身份证识别功能。

人工智能的应用——大数据

AIScratch4.0提供了一些基本的公共的大数据采集功能，包括天气、图片、音乐、音效等。这些图片和声音数据来源于网络，由于版权不清晰，因此除了可以在教学中任意使用外，注意不要用于商业场合，以避免商业风险。



上述代码是通过“当前天气”和“当前温度”来获取你本地的天气情况，你的地理位置是系统通过你上网的IP地址进行智能判断得出。获取天气情况后拼接播报的文字，再用语音功能朗读出来。



查找图片有两个程序块，一个是普通查找，一个是查找透明背景的图片，我们可以在查找关键字上加上PNG可以找到精准的透明图。

自定义程序块——新建命令程序块(1)

我们经常要重复使用一些程序块代码，我们可以把这些重复使用的程序块代码组合成一个新的程序块，这个程序块称为自定义程序块。自定义程序块能够让代码更加简洁清晰。

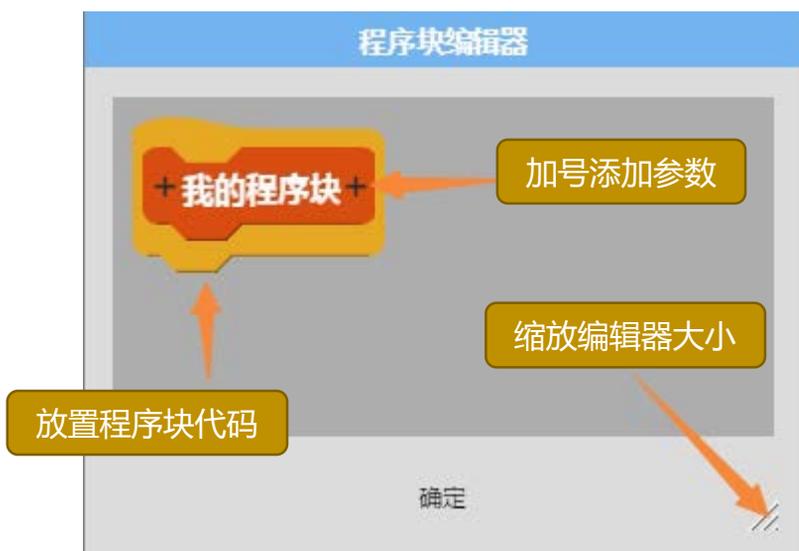


- ① 点击【更多模块】按钮；
- ② 点击【新建程序块】按钮，弹出新建程序块对话框；
- ③ 点击自定义程序块的分类；
- ④ 输入自定义程序块的名称；
- ⑤ 选择自定义程序块的类型；
- ⑥ 选择自定义程序块的作用范围；
- ⑦ 点击【确定】按钮。

第⑤步中，自定义程序块有三个类型，分别是命令型、报告型、条件型，命令型是直接运行一段代码没有返回值，报告型会返回一个结果值（任意类型的值），而条件型返回成立或不成立的结果（布尔类型值）。

自定义程序块——新建命令程序块(2)

自定义程序块的内部其实相当于一个代码编辑区，可以将所有已有的程序块拖入编辑器组合成一段代码，删除程序块方法类似：把程序块拖出编辑器就可以了。



点确定后，会弹出程序块编辑器，程序块编辑器与代码编辑区一样，也是可以放置程序块来组合代码的，只需要从程序块列表中拖入程序块到编辑器就可以编辑代码了。

下面我们制作一个让精灵旋转指定圈数的命令程序块，点击“我的程序块”文字，在弹出的对话框中修改“我的程序块”为“旋转圈数”，单击确定，回到编辑器状态后，我们看到程序块的名称已经修改为“旋转圈数”了，我们点击右侧加号，弹出“创建参数”的对话框，如下图：

自定义程序块——新建命令程序块(3)

程序块参数是指程序块外部给出的初始值，程序块会让初始值参与到程序的运行中，从而影响程序的运行结果。



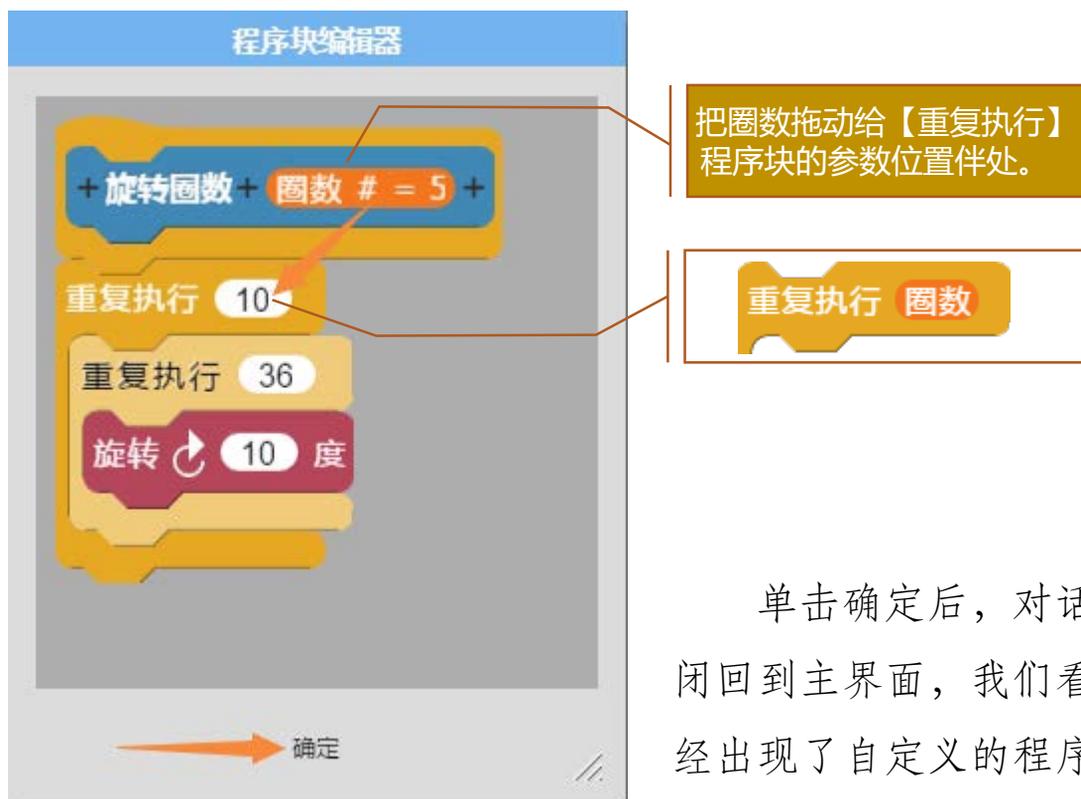
在“创建参数”对话框中，我们输入参数名称为“圈数”，选择参数的类型为“数字”，选择“固定单个参数”并输入默认值为“5”，单击确定，这时返回编辑器，程序块标题变成下图：



自定义程序块——新建命令程序块(4)

程序块参数是指程序块外部给出的初始值，程序块会让初始值参与到程序的运行中，从而影响程序的运行结果。

我们从程序块列表中拖入程序块，如下图：



单击确定后，对话框关闭回到主界面，我们看到已经出现了自定义的程序块

（注：我们创建程序块时选择了分类【更多模块】）。

这时我们就可以使用自己定义的命令块了。你可以修改不同的参数来体验运行的效果。



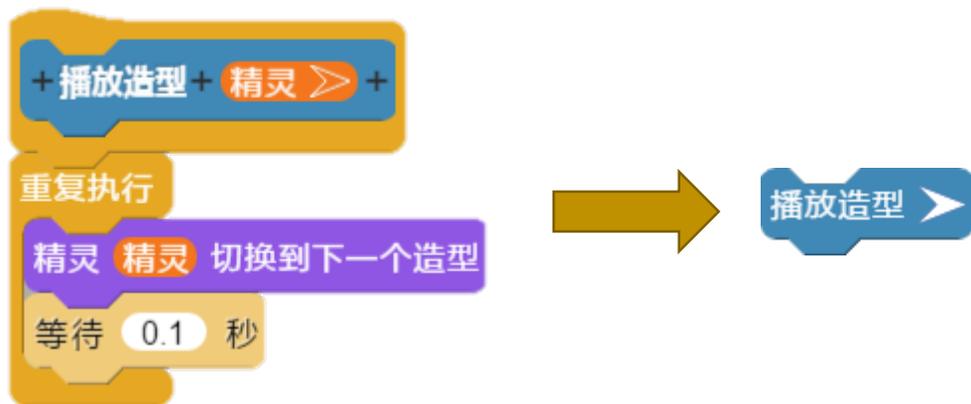
自定义程序块——参数的类型(1) 对象型

参数有非常多的类型，下面我们重点来讲解不同类型的含义和作用。



参数类型设定，其实就是这个程序块接收什么样的值，值的类型很多，包括：对象、文本、列表、数字、真假（布尔）、任意值，以及一些特殊类型如：命令、报告、条件、嵌套命令等。

①对象类型是指精灵（角色），程序块接收精灵对象为参数，通常这个参数的值就是调用这个程序块的精灵本身。



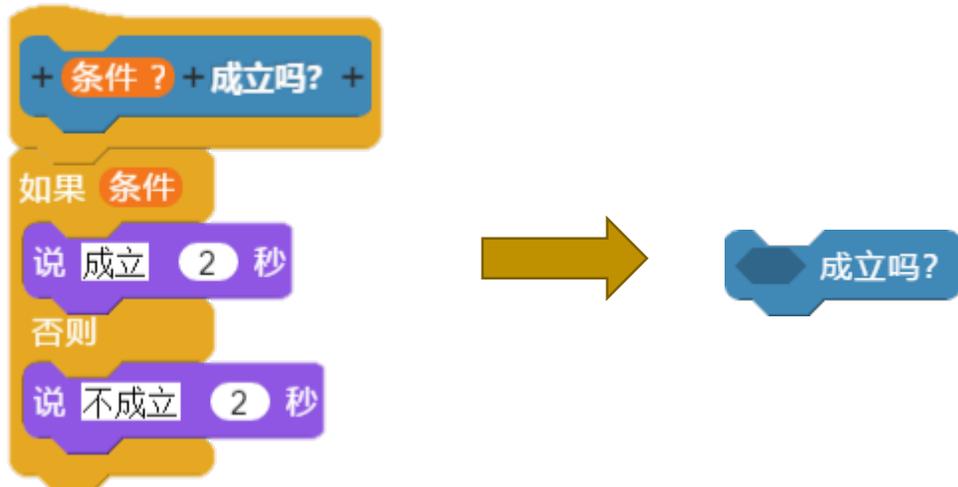
自定义程序块——参数的类型(2) 真假型



文本类型是指接收的参数为文字，上面这个程序块运行结果如下：

文本类型

②真假类型（又称布尔类型），是接收真值（True）和假值（False），根据真值（成立）或假值（不成立）来改变程序块运行的结果。



自定义程序块——参数的类型(3) 列表型

③列表类型是指把列表变量作参数代入程序代码中运行，以下代码的作用是统计列表中含某个字符串的个数：



列表内容

列表 国家 包含 中国 个数

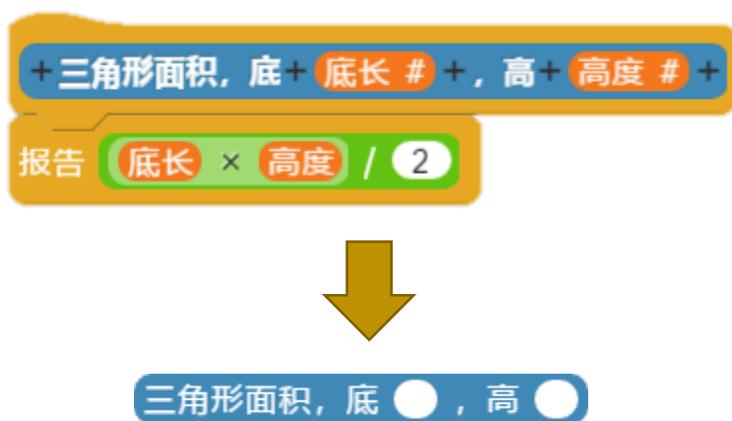
运行结果：2（直接显示在舞台上），如果是报告型程序块，则在最后用 报告 b 来返回结果；

自定义程序块——参数的类型（4）数字型

④数字类型是指把数字作参数代入程序代码中运行，以下代码的作用是求三角形的面积，这里我们创建一个报告型的程序块：



与创建命令型自定义程序块的过程相同，不过在最后我们要返回一个结果值，结果值在报告程序块中填写。



我们可以看到报告程序块与命令程序块的外观是不一样的，而且在程序代码中也只能做为参数来调用，不能直接做为程序代码块。



报告程序块把运行结果作为参数赋值给命令程序块参与运行

自定义程序块——参数的类型（5）任意型

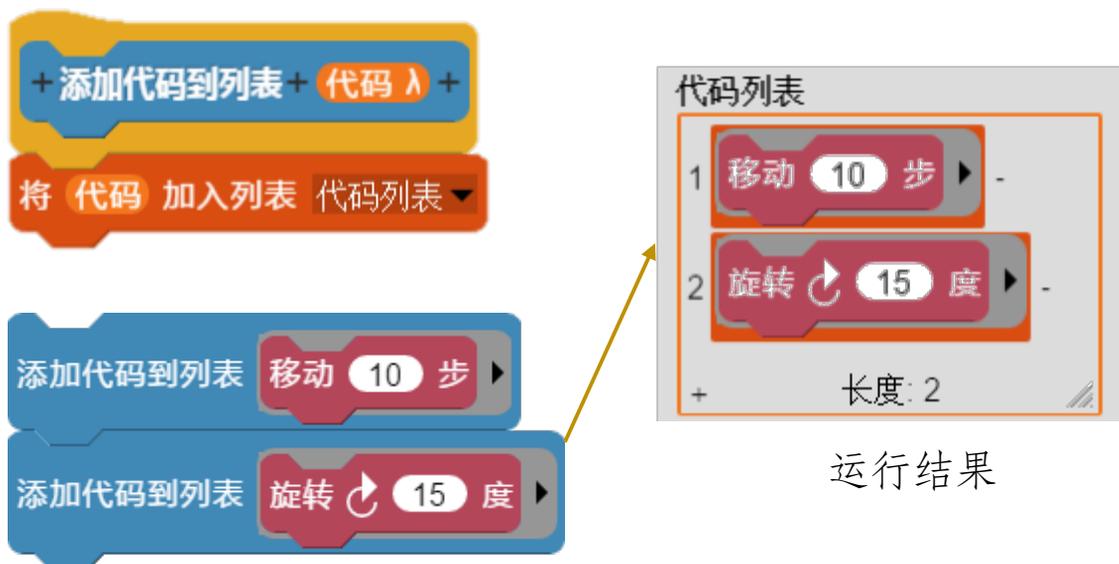
⑤所有类型（任意类型）是指对参数没有任何要求，接收任意类型的值作为参数，下面程序块作用是判断参数是不是数字类型，并返回一个布尔类型值（成立或不成立）：



布尔类型（真假类型）的值：

真=成立=True；假=不成立=False；

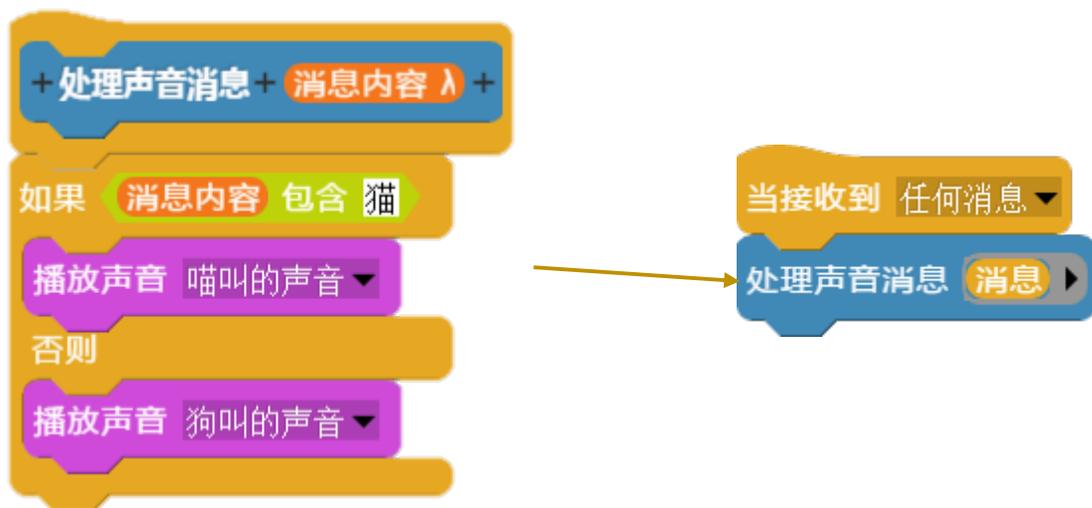
⑥命令（内置）类型参数可以直接将一个程序块或多个程序块作为参数，在自定义程序块内部进行调用。



运行结果

自定义程序块——参数的类型（6）报告型

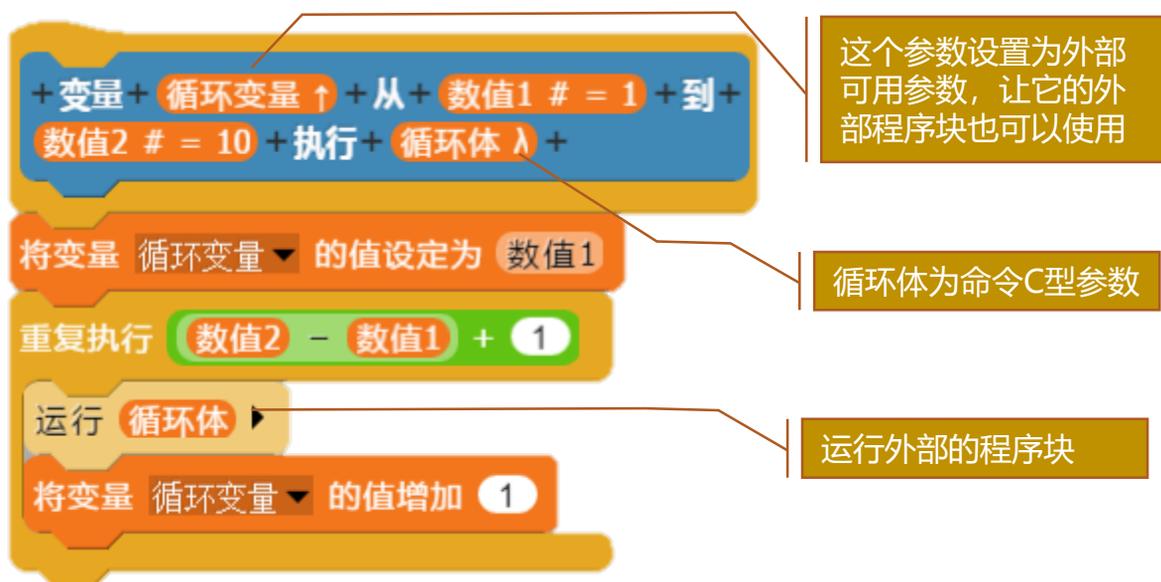
⑦报告型参数，是把报告程序块返回结果作为参数，程序块运行时，会先运行报告程序块代码，并把报告程序块的结果作为参数代入自定义程序块中，从而影响程序块结果。下面的程序块代码实现根据收到消息的内容，来播放相应的声音：



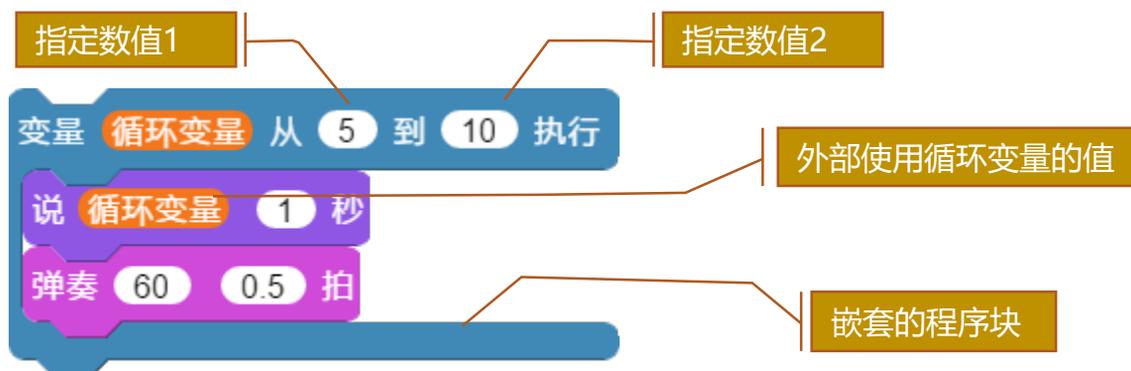
⑧条件型参数，程序块运行时，会先运行条件程序块代码，并把条件程序块的结果作为参数代入自定义程序块中，根据成立或不成立来影响内部运行的分支。左边的程序块代码根据条件来播放声音：

自定义程序块——参数的类型 (7) 命令型

⑨命令C形参数，类似于命令内置参数，不过是以C形结构提供给用户。下面的程序块代码实现一个指定数值的循环：



下面是自定义程序块的使用代码，代码作用是从指定数值1循环运行嵌套程序块，直到指定数值2结束。



自定义程序块——参数的设置

参数的设置：固定单个参数、数量可变参数、外部可用参数

- 固定单个参数 默认值:
- 数量可变参数 (参数是列表数据类型)
- 外部可用参数 (外部可以使用此参数变量)

①固定单个参数是指参数的数量只有一个，不能动态增加参数的数量，默认值是用户缺省

时的值。②数量可变参数是指参数数量可以外部动态增加，通过点击向右的三角形增加一个参数，通过点击向左的三角形减少一个参数。此时的参数相当于一个列表，可以用列表程序块来操作使用。③外部可用参数是指参数在外部的程序块也可以使用。

The image shows a Scratch script for a '求和' (Sum) function. At the top, there is a '求和' block with a '数值列表...' (Number List) parameter. Below it is a '脚本变量' (Script Variable) block for '和' (Sum). The script then sets '和' to 0 and '序号' (Index) to 1. It enters a loop '如果 列表 数值列表 的长度 > 0' (If list length > 0). Inside the loop, it repeats until '序号 = 列表 数值列表 的长度 + 1'. In each iteration, it sets '和' to '和 + 第 序号 项在列表 数值列表 中的值' (Sum + value of item at index in list) and increments '序号' by 1. Finally, it reports '和'.

Annotations on the right side of the image:

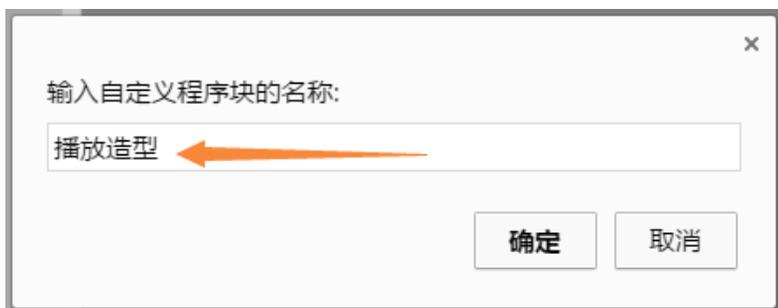
- 数量可变参数, 求10+2+5的和 (Variable parameter, sum of 10+2+5)
- 添加一个参数 (Add a parameter)
- 删除最后一个参数 (Delete the last parameter)

自定义程序块——快速自定义程序块(1)

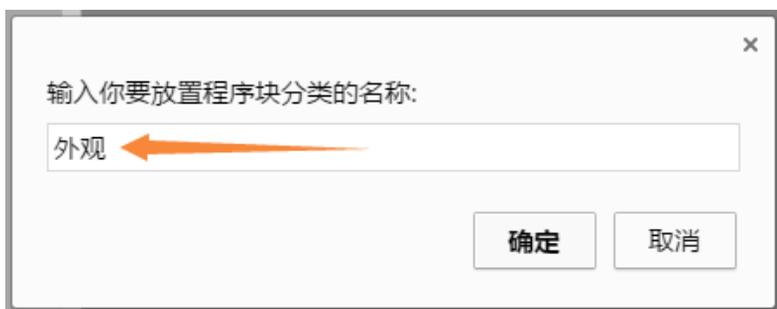
在实际编辑中，我们时常要将一大段已经写的代码合并成一个独立的程序块，这时我们可以使用快捷的创建方法。



在已经编辑好的程序块代码的上方单击鼠标右键，点击“自定义为程序块”菜单项，弹出输入自定义程序块名称的对话框，输入自己定义的名称，如“播放造型”。



点击确定后弹出输入程序块分类的对话框，我们可以输入程序块放置在哪个分类中，在本例中我们放置在“外观”分类中，如果你输入的分类不存在系统中，系统将自动创建一个新的分类。



自定义程序块——快速自定义程序块(2)

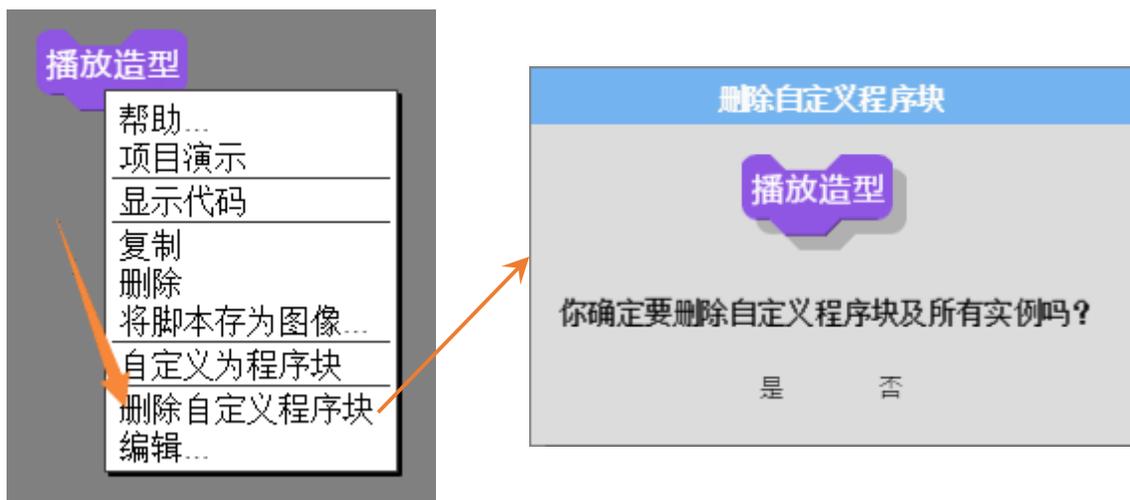
快速自定义程序块可以设置编辑密码，如果设置了密码，在下次编辑时需要提供正确密码后才能进入编辑器编辑。



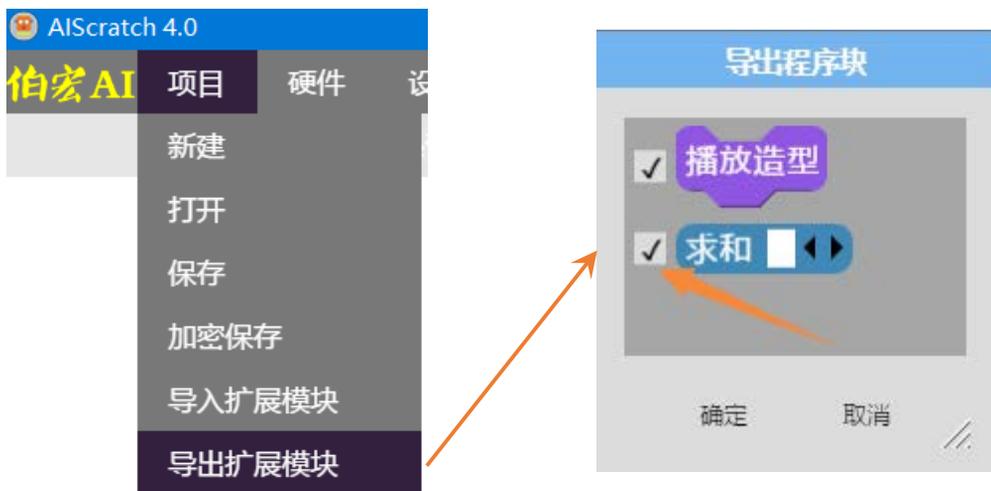
密码留空的话，表示不设置密码，如果设置了密码，没有密码将无法查看也无法编辑这个自定义的程序块，因此要牢记密码，而运行这个程序块则不需要提供密码。



自定义程序块——删除与保存自定义程序块



①删除：在自定义程序块上方单击鼠标右键，弹出快捷菜单，在菜单中点击“删除自定义程序块”，弹出对话框选择“是”。

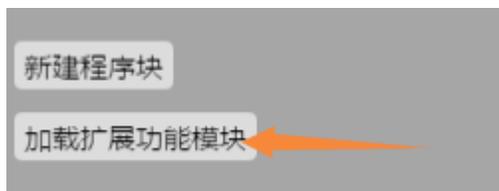


②保存：点击主界面【项目】菜单项，点击【导出扩展模块】，在弹出的对话框中，勾选要保存的自定义程序块，点击确定，在弹出的保存对话框输入保存文件的名称，点击保存（正常情况下，保存的文件都将以“.ext”为扩展文件名）。

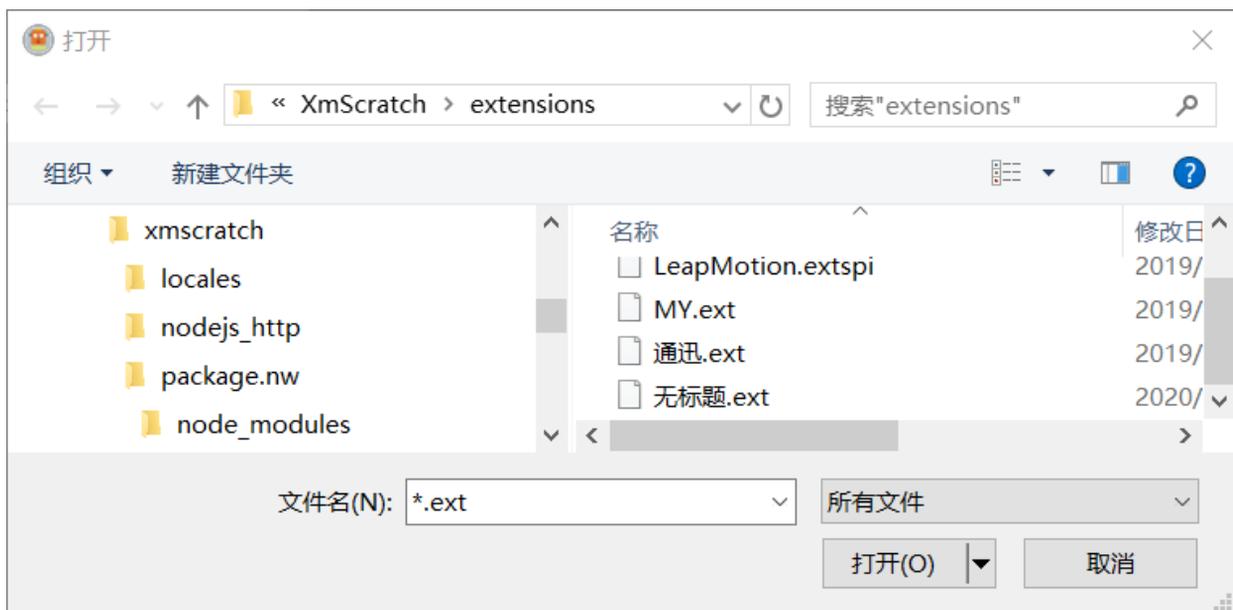
自定义程序块——导入程序块库文件



或者



③导入：点击主界面【项目】菜单项，点击【导入扩展模块】（或者在“更多模块”中点击【加载扩展功能模块】的按钮），在弹出的打开对话框中，我们选择自己的扩展功能文件后打开。



网络通讯编程——基础知识(1)

在实际编程应用中，我们时常要进行网络通讯。网络通讯分为互联网通讯与局域网通讯两种。

互联网与局域网技术上基本相通，本质上都是实现电脑、手机及网络设备之间的信息交换作用，主要区别在它们应用范围不同，互联网是面向全球的互联互通应用，全球电脑、手机及互联网设备通过服务器和互联网络连接在一起。而局域网面向的是局部用户的一个网络，比如某某校园网、网吧内网、家庭内网等。

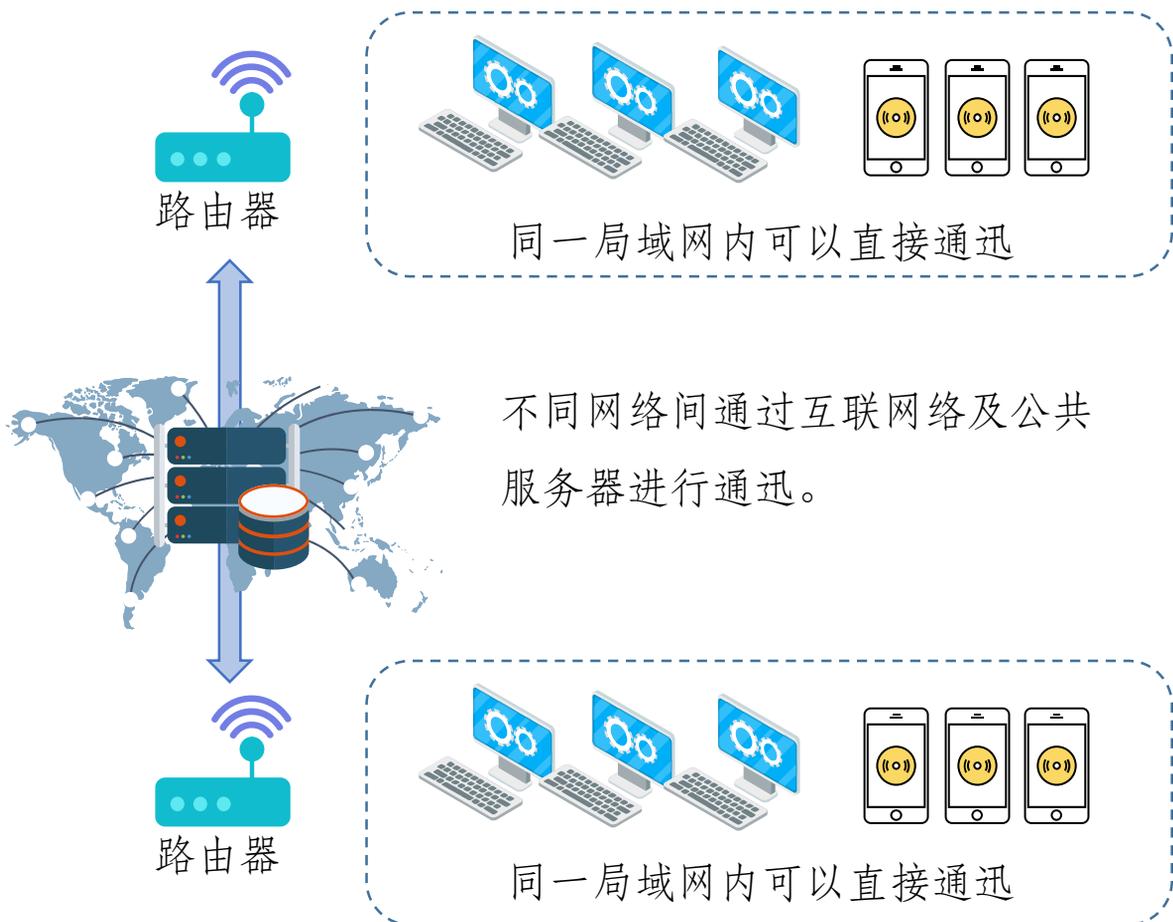
如果我们从日常生活来简单判断它们，可以通过以下示意图来了解互联网与局域网的关系。



网络通讯编程——基础知识(2)

网络通讯的过程就是网络设备（电脑、手机等）之间传递信息的过程，硬件方面需要有线网络或无线网络相互连接，软件方面需要双方共同遵守的网络通讯协议（协议就像两个谈话的人需要一个共同的语言一样，是沟通理解的基础）。

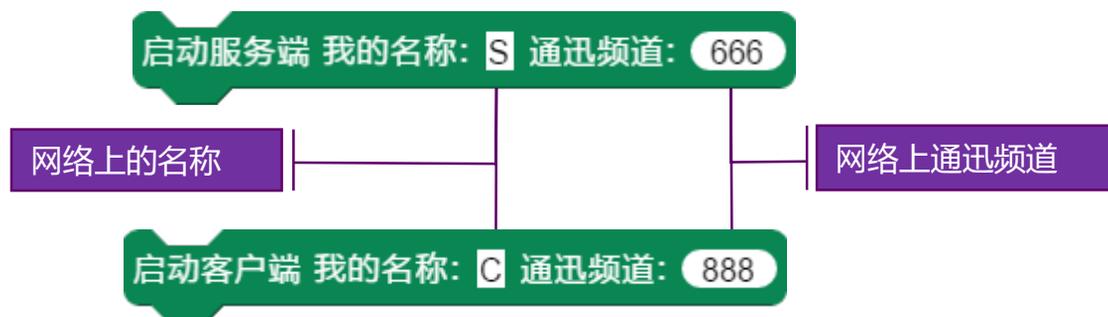
通常在同一局域网的两台电脑是可以通讯的，但在不同两个局域网的电脑是不能直接通讯的，必须通过公共互联网上的服务来进行通讯。



网络通讯编程——局域网通讯(1)

由于互联网通讯需要借助公共网络服务器，因此我们重点讲解在AIScratch中局域网网络通讯的编程技术。

首先应该知道的在本系统是我们在一台电脑发出信息，其实每台电脑都可以收到信息的，但是为了给指定的电脑发送信息，我们事先会给自己的电脑定义一个名称和一个通讯的频道，网络上的信息只有电脑名称和通讯频道相同时，信息才会被接收处理，否则会直接忽略掉。

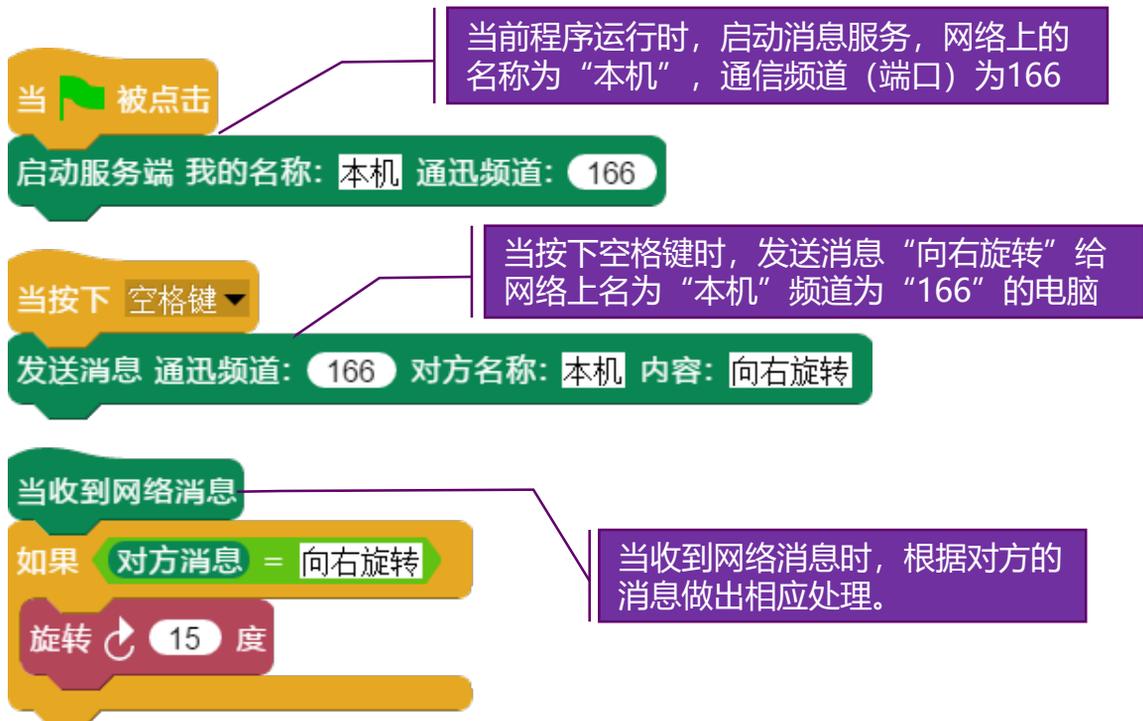


在两台电脑通讯前（需要保证上网络已经互通）我们需要启动消息服务功能，系统提供了两个程序块来启动消息功能，启动消息服务端和启动消息客户端，功能上两个程序块是一样的，主要是为了方便在同一台电脑模拟两台电脑的通讯，相当于一台接收，一台发送，在实际有多个电脑通讯中，只需要一启动其中一个程序块的消息服务就可以了。在上述程序块中已经默认了名称和频道，实际使用可以自己定义名称和频道，确保相互通讯的电脑处于同一个频道之中。

网络通讯编程——局域网通讯(2)

局域网通讯使用系统统一的消息机制，和我们前面提到广播消息是一样的，接收的消息都是存储在消息列表中的。

下面这个程序代码分成三个部分，作用是给自己的电脑发送一个网络消息“向右旋转”，如果接收到“向右旋转”则执行相应程序块。



上面这个程序代码是在一台电脑上发送和接收网络消息，如果我们在两个电脑上通讯，我们可改成以下的程序代码：

网络通讯编程——局域网通讯(3)

电脑1的程序代码：启动消息服务时广播“电脑1启动”的消息给所有网络上的电脑（此时对方名称为空值），当按下空格键时发送消息“旋转”给名称为“C”的电脑，当收到“移动”网络消息时移动当前精灵：

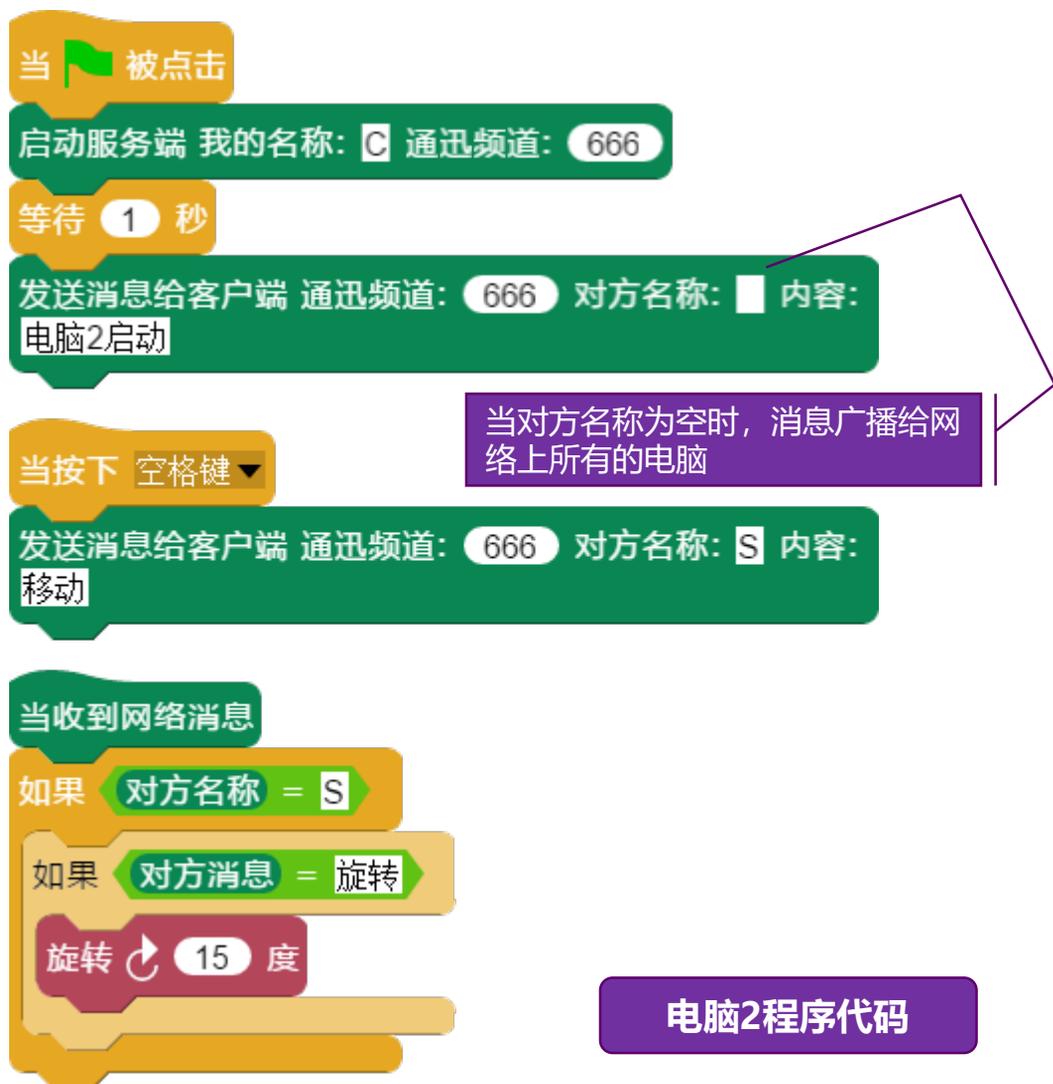


当对方名称为空时，消息广播给网络上所有的电脑

电脑1程序代码

网络通讯编程——局域网通讯(4)

电脑2的程序代码：启动消息服务时广播“电脑2启动”的消息给所有网络上的电脑（此时对方名称为空值），当按下空格键时发送消息“移动”给名称为“S”的电脑，当收到“旋转”网络消息时旋转当前精灵：



网络通讯编程——发送邮件(1)

发送邮件 (Email) 是典型的互联网通讯应用, 系统提供对 QQ 邮箱的信件发送 (没有 QQ 号的可以自己注册一个)。

发送邮件前需要知道自己 QQ 的账号及邮箱密码, 注意: 这里的邮箱密码不是 QQ 密码, 是在 QQ 邮箱网页设置里生成的授权码, 我们通过浏览器打开网址 <http://mail.qq.com>, 用自己的 QQ 账号和密码登录邮箱后, 点击①【设置】-②【帐户】, 如下图:



POP3/Exchange/CardDAV/CalDAV服务

POP3/SMTP服务 (如何使用 Foxmail 等软件收发邮件?)	已关闭 开启
IMAP/SMTP服务 (什么是 IMAP, 它又是如何设置?)	已关闭 开启
Exchange服务 (什么是Exchange, 它又是如何设置?)	已关闭 开启
CardDAV/CalDAV服务 (什么是CardDAV/CalDAV, 它又是如何设置?)	已关闭 开启
(POP3/IMAP/SMTP/CardDAV/CalDAV服务均支持SSL连接。如何设置?)	

温馨提示: 在第三方登录QQ邮箱, 可能存在邮件泄露风险, 甚至危害Apple ID安全, 建议使用QQ邮箱继续获取授权码登录第三方客户端邮箱 。生成授权码

网络通讯编程——发送邮件(2)

③下拉找到【POP3/IMAP/SMTP/Exchange/CardDAV/CalDAV服务】并点击【开启】POP3/SMTP服务，此时会弹出对话框④，我们用自己密保手机发送“配置邮件客户端”到指定号码后，点击⑤【我已发送】，此时弹出发送邮件的授权码（⑥记下备用）。



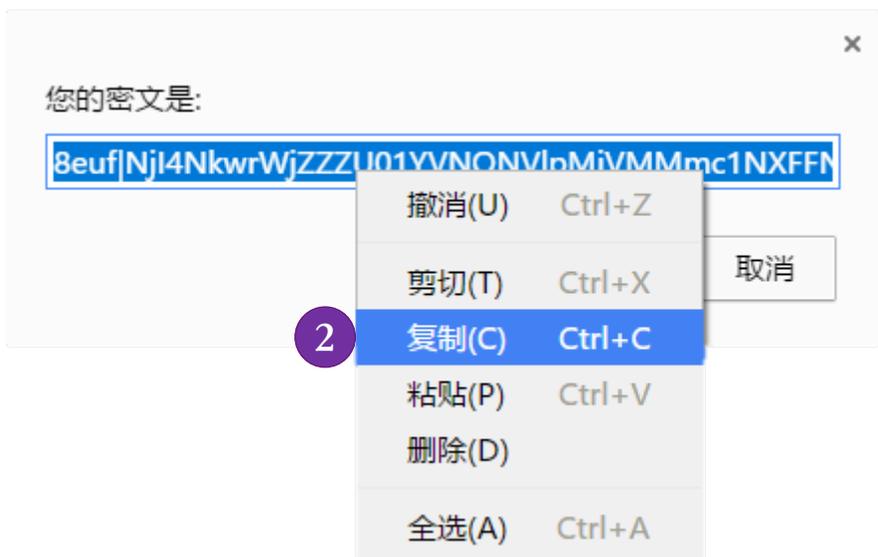
最后要记得点击【保存更改】。

网络通讯编程——发送邮件(3)

拿到邮件密码（授权码）后，为了在程序块中不显示授权码明文，我们可以先加密授权码：

1 密码转密文 这里复制你的授权码填入

点击弹出生成的密文，复制到剪切板



2

发送邮件 对方邮箱: nkxhl@163.com 标题: Hello! 内容:
我是伯宏AI 我的QQ邮箱 86012999@qq.com 密码密文
nbczdmybcfupbhfc

3

粘贴密文到这里

我的QQ邮箱账号

点击发送邮件程序块，邮件就发送出去了，打开邮箱的收件箱查看结果吧！

网络通讯编程——发送钉钉群消息(1)

我们可以通过钉钉群的群机器人发送消息到指定的聊天群。

首先我们要创建钉钉的群机器人，得到群机器人的链接，在电脑上登录钉钉后，双击打开钉钉群，点击①【群设置】-②【智能群助手】，点击③【添加机器人】。



网络通讯编程——发送钉钉群消息(2)

点击④【自定义】后，在弹出的对话框，选择⑤【添加】。



网络通讯编程——发送钉钉群消息(3)

在添加机器人对话框，填写基本的信息：

添加机器人

机器人名字： 6

* 添加到群组：

* 安全设置 ? 自定义关键词 7

[说明文档](#)

8

[+](#) 添加 (最多添加 10 个)

加签

IP地址 (段)

9 我已阅读并同意 [《自定义机器人服务及免责条款》](#)

10

这里自定义的关键词是句号，发送内容带上句号才会被发送出去。更多设置查看说明文档

网络通讯编程——发送钉钉群消息(4)

最后完成添加机器人流程，得到群机器人的链接地址：

添加机器人



1. 添加机器人 ✓

2. 设置webhook，点击设置说明查看如何配置以使机器人生效

Webhook:

* 请保管好此 Webhook 地址，不要公布在外部网站上，泄露有安全风险

使用 Webhook 地址，向钉钉群推送消息

群机器人链接地址

发送钉钉群消息 机器人链接:

发送内容:
测试内容。

发送内容记得加上 “。”

硬件智能编程——连接硬件(1)

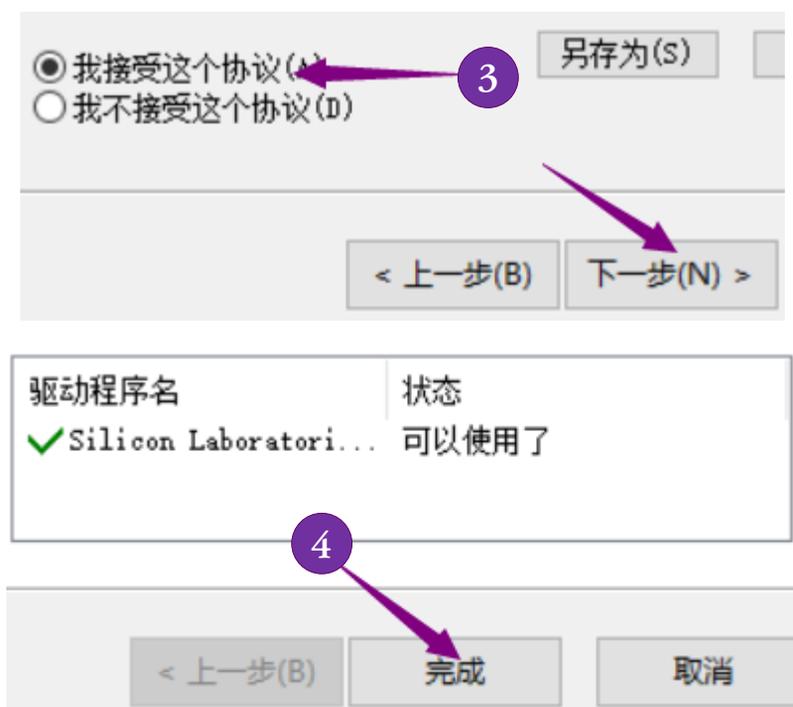
AIScratch4.0支持连接Arduino硬件主控器和对应电子传感（如需要硬件可以在交流群购买），硬件支持USB和蓝牙连接。

在连接硬件以前我们需要安装设备的驱动（大部分硬件都是串口通讯的，但现在电脑基本上是USB接口，因此需要USB转串口的设备驱动），我们点击【硬件】菜单，点击①【安装驱动】。



硬件智能编程——连接硬件(2)

在弹出的对话框③选择接受协议，点击下一步，直接到完成，这个安装驱动的过程，一台电脑只需要安装一次即可。



安装好驱动后，打开主控器电源，用USB线将主控板与电脑的USB端口连接起来，点击【硬件】菜单，点击①【连接串口】。



硬件智能编程——连接硬件(3)

在弹出的对话框中点击②【扫描】，如果设备正确连接，扫描后可以在设备列表中看到COM的字样，③选择设备并点击④【连接】。



另一种连接方式是使用蓝牙连接设备，这个蓝牙连接好处是没有线缆的束缚，使用起来更加方便（两种连接方式两选一即可）。



硬件智能编程——连接硬件(4)

在弹出的对话框中点击②【扫描】，扫描完成后③选择主控器对应的编码型号，④点击【连接】，连接完成后我们就可以使用硬件的所有程序块了。



关于更多的硬件设备使用方法，可以参看硬件说明和软件中的帮助系统。更多主控板的定制连接，我们会在专题中讲解。

在教学中教师的课件有对本硬件更为详细的教程。

高阶者使用教程

01

教学网络与演示

02

数据库的操作

03

网络数据采集

04

Web网页服务器

05

外部硬件的开发扩展

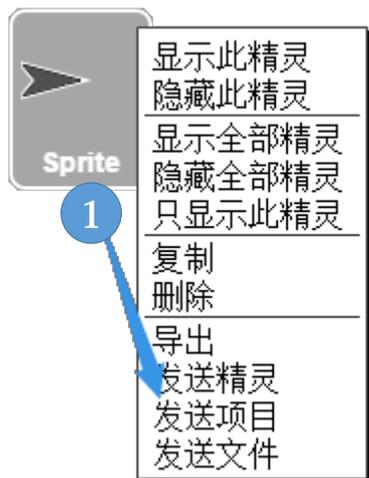
06

综合案例

教学辅助功能——教学网络管理

AIScratch提供了校园网络化教学的一些辅助功能，如资源的网络传输、舞台的录制、教学互动演示的制作。

在校园网络化教学中，我们经常要不同电脑间传递素材，比如声音、造型、代码、文件和项目，为此系统中内置了网络传输功能，既可以根据电脑名称来发送素材给对方，也可以广播素材给所有人，在校园版中还支持一些简单的远程控制。



在精灵按钮上方鼠标右键点击，在弹出的菜单，选择发送功能：发送精灵、发送项目、发送文件。输入对方名称是指对方的电脑名称。



教学辅助功能——素材网络传输

素材网络传输是指在局域网内传输声音、造型、代码、精灵、文件和项目。

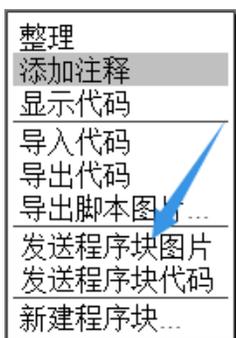
在网络传输前应该确保网传的电脑处于同一局域网络上，并且电脑间没有被路由器和防火墙隔离，部分家庭WIFI路由器默认的设置是设备间相互隔离的，需要新设置。AIScratch启动时会自动开启一个端口为13721的UDP消息服务，并利用这个服务来进行消息的网络传输。



发送造型



发送声音

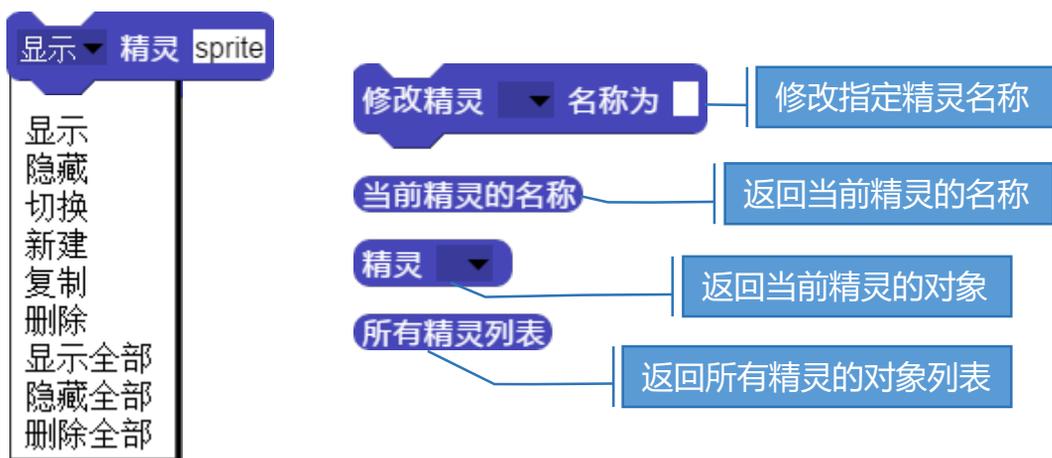


在造型编辑区、声音编辑区都有发送按钮可以发送造型和声音，在代码编辑区右键单击弹出的菜单，可以发送程序块图片和程序块代码。

教学辅助功能——教学演示制作（精灵控制）

在编程教学中我们经常要进行教学演示，我们通常的做法是录制屏幕视频，但系统还提供了我们更直观的方法来演示。

在系统的【界面】程序块分类中提供了舞台控制、精灵控制、程序块控制、对话框控制等用于系统在编辑状态下进行操作的命令程序块。教学演示程序的制作主要用到了精灵控制和程序块控制程序块。我们可以打开项目演示来学习它们的使用。

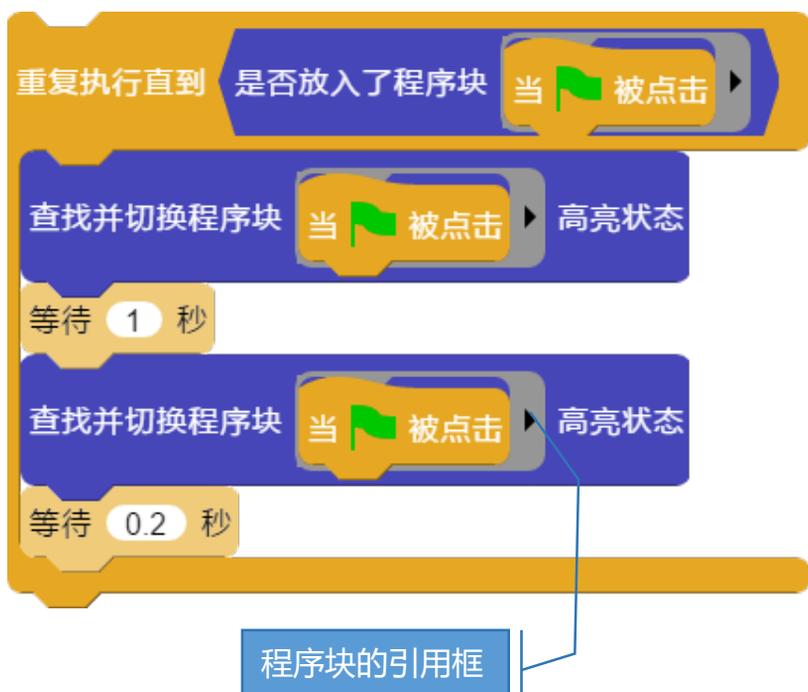


精灵的控制主要包括显示（显示全部）、隐藏（隐藏全部）、切换（设置当前精灵）、新建、复制、删除（删除全部）、修改精灵名称。所有的精灵列表程序块，返回精灵的对象可以用【列表】中遍历程序块来获取每个精灵对象及其属性。



教学辅助功能——教学演示制作（程序块控制）

程序块的控制主要用于教学演示的互动，包括切换程序块分类、强调程序块、检测代码编辑区用户的操作。



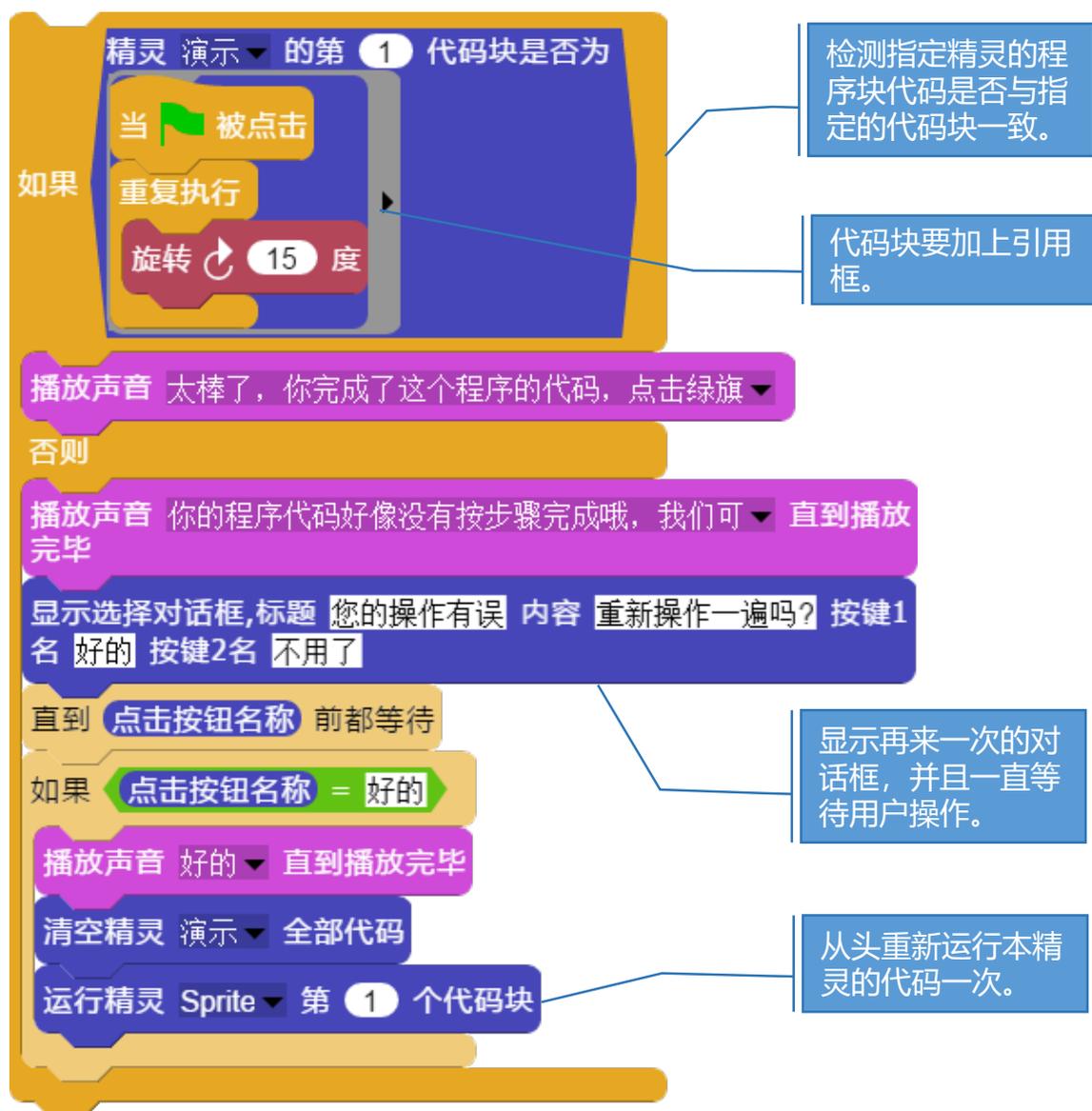
左侧的代码实现等待用户点击【控制】程序块分类按钮，在等待的同时不断闪烁

【控制】程序块分类按钮，提示用户点击的位置。

左侧的代码实现等待用户点击指定程序块，在等待同时不断闪烁指定程序块，提示用户点击的位置。这里使用到程序块的引用框，可以在单个程序块的右键菜单中找到这个功能。加框后可以引用这个程序块。

教学辅助功能——教学演示制作（用户操作检测）

检测用户操作既可以使用是否放入了程序块的检测功能，也可以使用代码块内容的对比来检测用户是否操作正确。



教学辅助功能——教学演示制作（代码控制）

界面程序块分类的功能简单来说就是原来用户需要通过鼠标操作的功能，通过程序块也能实现。同时可以监控用户操作。

设置当前精灵为

用于切换当前操作的精灵对象。

切换到 代码 编辑区

用于切换代码、造型和声音编辑区。

切换 程序块分类

用于切换、显示、隐藏、新建、删除分类。

添加代码块 到精灵 第 个代码的第 个块后面

添加代码。

替换代码块 到精灵 第 个代码的第 块

替换代码。

清空精灵 全部代码

用于清空指定精灵的全部代码。

查找并切换程序块 高亮状态

用于查找程序块，并高亮显示。

清除用户操作状态

用于清除用户最近的操作状态。

是否点击了程序块

用于判断是否点击了指定程序块。

是否放入了程序块

用于判断是否放入了指定程序块。

精灵 的第 代码块是否为

对比指定精灵指定代码。

最后点击程序块的名称

用于返回最后点击程序块的名称。

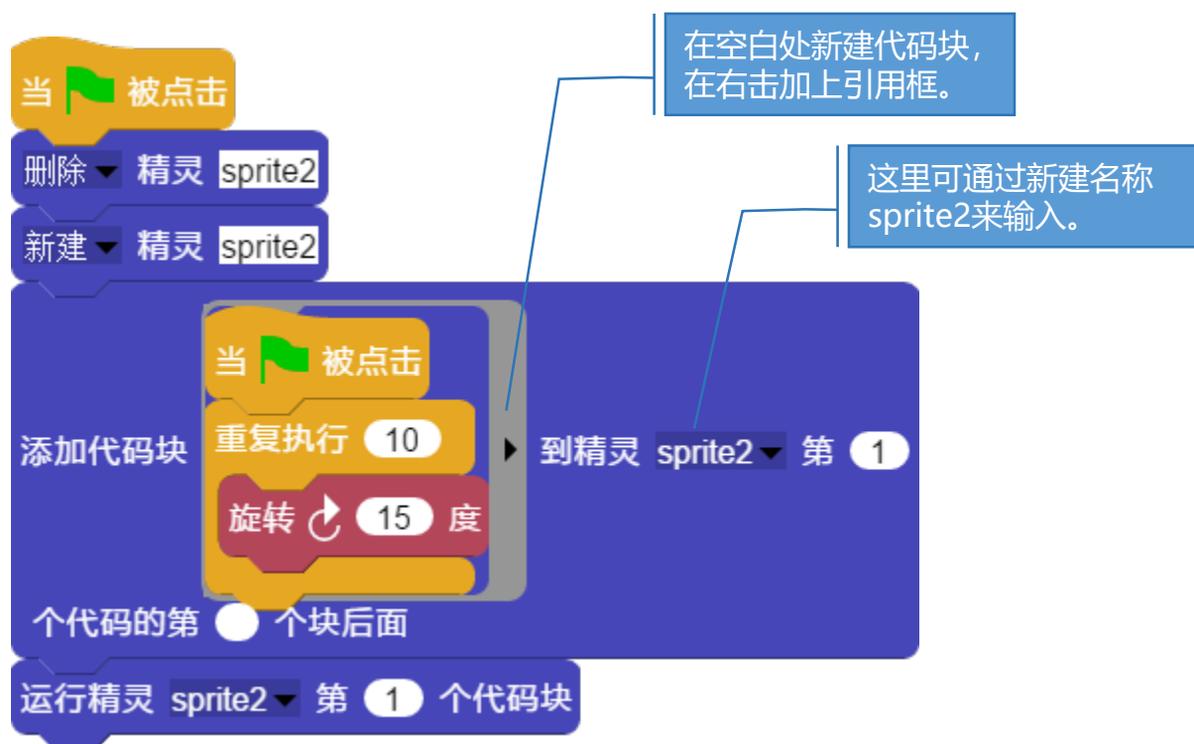
最后放入程序的程序块名称

用于返回最后放入代码的程序块名称。

教学辅助功能——教学演示制作（简单案例）

我们通过一个示例来讲解如何实现演示互动的。我们在给学员讲代码时可以使用这些界面控制的功能。

首先，我们新建一个项目，把默认的精灵sprite复制一份，复制的方法：在精灵上方右击弹出菜单，点击复制。我们用第一个精灵(sprite)来写代码控制第二个精灵(sprite2)的代码来进行教学互动演示（这个过程也可以通过新建精灵的程序块完成）。为避免教学演示的精灵显示在舞台上，我们要把精灵sprite的造型隐藏起来，在精灵sprite上方右键单击，在弹出的菜单选择隐藏。我们在精灵sprite代码编辑区写入以下代码：



数据库的操作——认识MySQL数据库

数据库 (Database) 是按照数据结构来组织、存储和管理数据的仓库。我们也可以将数据存储于文件中，但是在文件中读写数据速度相对较慢。比较常用的数据库就是MySQL。

在我们开始学习MySQL数据库前，让我们先了解一些术语：

数据库：数据库是一些关联表的集合。

数据表：表是数据的矩阵。在一个数据库中的表看起来像一个简单的电子表格。

列：一列(数据元素)包含了相同类型的数据。

行：一行(记录)是一组相关的数据，例如一条用户的数据。

主键：主键是唯一的。你可以使用主键来查询数据。

索引：使用索引可快速访问数据库表中的特定信息。索引是对数据库表中一列或多列的值进行排序的一种结构。类似于书籍的目录。

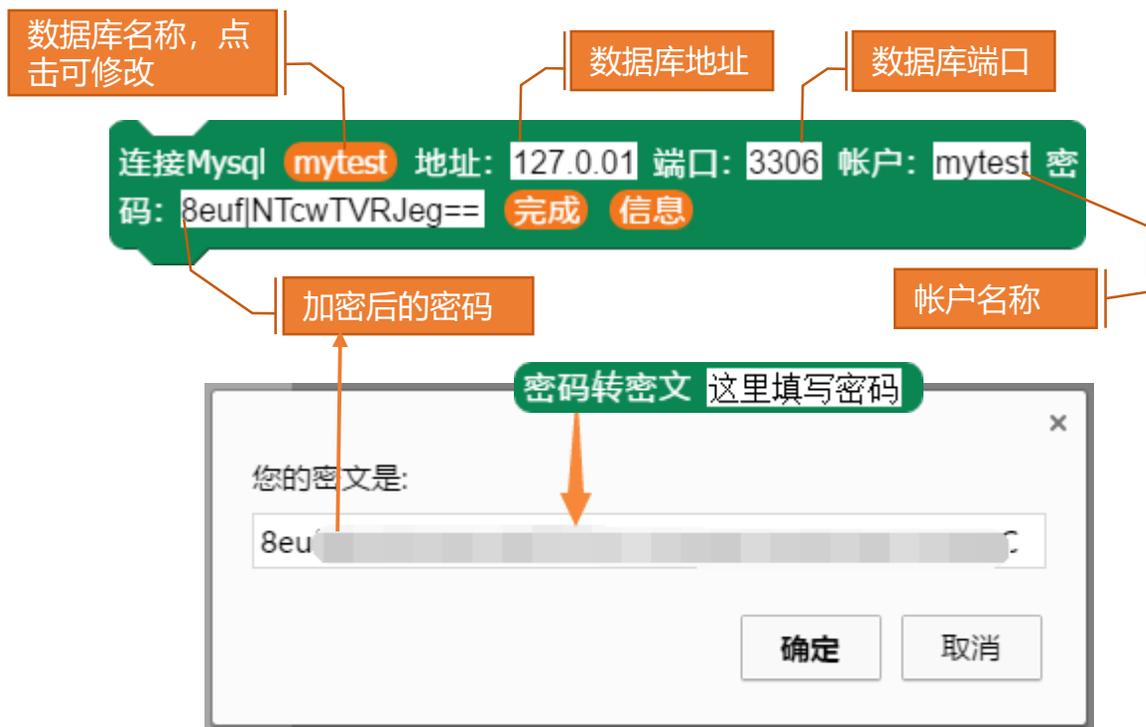
MySQL是一种关联数据库管理系统，它将数据保存在不同的表中。MySQL是开源的，所以不需要付费；MySQL支持大型的数据库，可以处理拥有上千万条记录的大型数据库；MySQL使用标准的SQL数据语言形式。

数据库的操作——连接数据库

数据库可以很好地用来存储程序中的数据，数据的操作包括增加、删除、修改、查询等操作。AIScratch提供了很好的MYSQL数据库操作和数据操作。

在使用AIScratch的MYSQL数据库功能之前，你必须要有可以连接使用的MYSQL数据库，如果没有，你可自己在本机上安装MYSQL系统，并新建好数据库，具体搭建过程我们可以在网络上查找相关教程，这里我们主要讲解AIScratch的数据库连接和使用。

第一步是连接数据库，操作前先需要知道数据库的名称、IP地址、端口、用户名称及密码（我们使用“密码转密文”程序块来加密明文）。



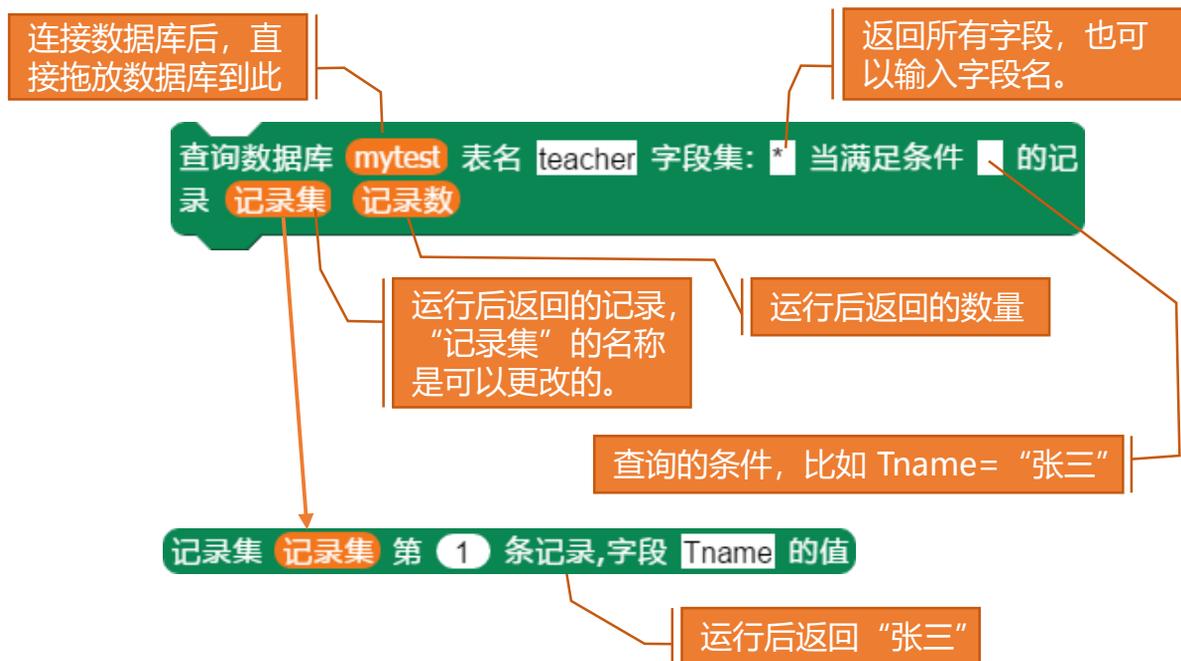
数据库的操作——查询数据

连接数据库后，我们就可以根据条件进行数据记录的查询，执行后就会返回查询表。

在上述的数据库mytest中，我们存有一张教师信息的数据表teacher，表的结构和记录如下：

Tno	Tname	Tsex	Tbirthday	Prof	Depart
教工编号 (主码)	教工姓名	教工性别	教工出生年月	职称	教工所在部门
1	张三	男	1979-03-02 00:00:00	中级	高一组

在teacher数据表中共有六个字段：Tno、Tname、Tsex、Tbirthday、Prof、Depart。下面的程序块取得数据库mytest中表名为teacher的所有记录：



数据库的操作——遍历数据

我们通过查询程序块得到记录集，对每个记录集的操作我们可以使用遍历数据程序块来完成。

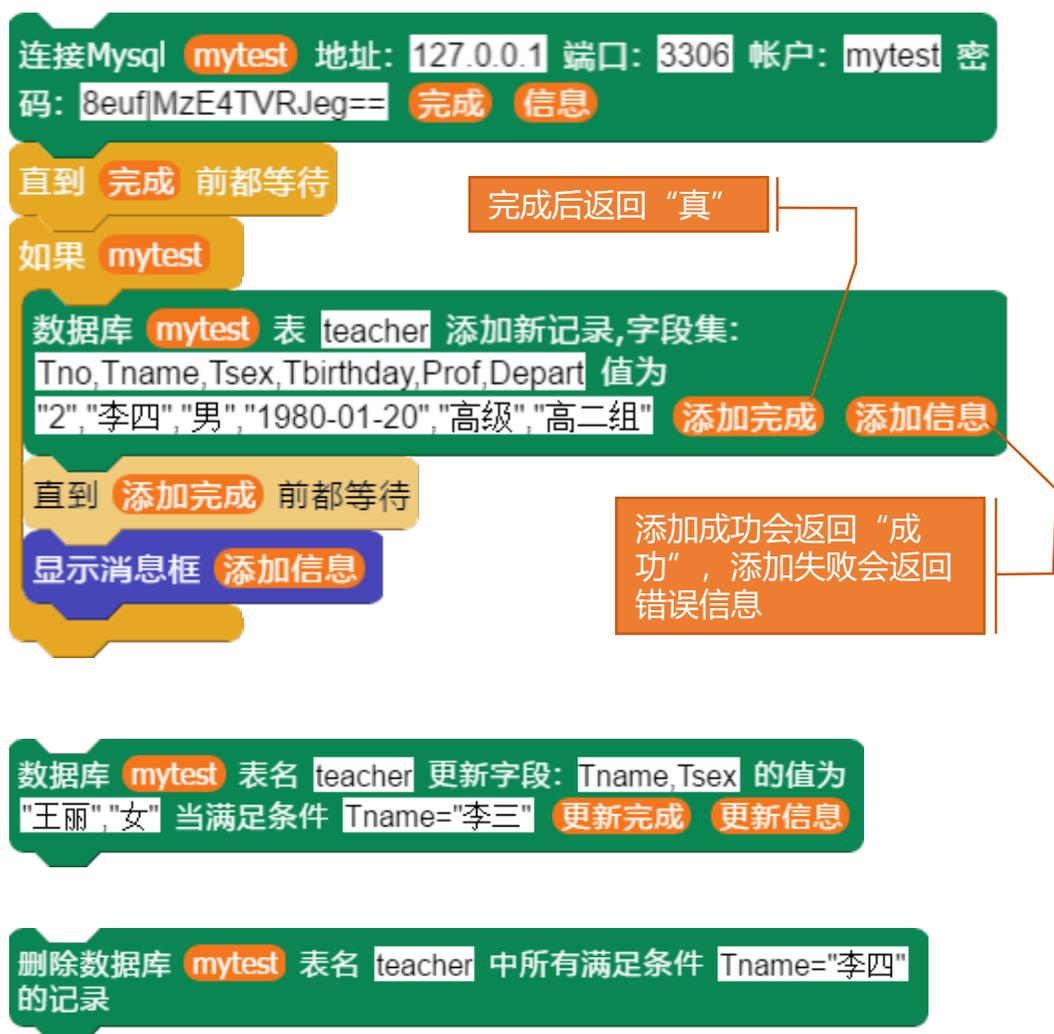


在上面的代码中，我们先连接数据库mytest，在连接完成以前一直等待，如果连接成功则查询数据表teacher的所有记录，查询结果保存在“记录集”变量中（“记录集”是一个变量的名称，可以根据需要点击名称进行修改），记录数量放在“记录数”变量中，把“记录集”拖放到“遍历记录”的程序块中，每条记录保存在名为“记录”的变量中，记录的序号保存在“序号”的变量中。如果连接失败，则显示错误信息。

数据库的操作——添加、删除和更新记录

数据记录的添加、删除和更新记录。

在成功连接数据后，可以使用“添加新记录”的程序块，输入相应的字段名称，多个字段名用逗号分隔，字段值与字段名一一对应，并加上引号，执行完成后“添加完成”变量的值为真，如果添加成功，“添加信息”变量的值为“成功”，否则返回错误信息。



数据库的操作——执行SQL语句

因为MYSQL是支持SQL语句的，所以系统提供了执行程序块。

下面的代码执行了一个查询语句“select * from teacher”从数据库mytest的teacher表中返回所有的记录，如果得到结果把所有的Tname(姓名)字段的值通过遍历的方式一个一个显示出来。



执行SQL语句返回的结果，其实是一个JSON的对象（JSON是一种具有结构化的字符串，后面内容会详细讲解）。我们可以通过JSON转换程序块（这个程序块在【大数据】分类中）显示一下这个“执行结果”JSON对象变量的内容：

显示消息框 JSON 执行结果 转文本

网络数据采集——基础知识

人工智能的很多应用都需要用到网页数据的采集，AIScratch提供了强大的数据采集功能。

我们在浏览器上通常见到的网页代码基本都是HTML格式，它的主要作用就是对网页进行排版和结构搭建，分类，控制网页中的信息。它的主要结构就是由<标签字></标签字>成对组成的文本语句构成的，形如<html></html>、<body></body>、<title></title>等，一组标签的内容我们称之为一个元素（DOM），元素是可以嵌套的，如<html><title>我的网页</title></html>，被嵌套的元素我们称之为子元素。我们网页上的所见的所有内容都是由这些HTML代码包裹起来的形成的，我们可以在浏览器的网页上单击右键来查看网页源代码来看到它们。这种结构化的语句我们称之为超文本。超文本的每个标签都有特定的含义比如IMG表示是一个图片，BR表示换行，TABLE表示是一个表格，A表示是一个链接。在标签的后面可以加上一些属性值，如 在这个标签里加了一个属性scr，它的值为http://xxx.com/abc.jpg，这个属性其实就是图片的文件地址，我们可以通过这个地址下载这个图片。HTML有很多标签，每个标签的含义这里就不去详细讲述，更多可以参看：https://www.w3school.com.cn/html/html_elements.asp

网络数据采集——获取网页代码

AIScratch是通过获取和分析网页的HTML的元素来采集网络数据的，只要我们能在网页看到的，一般都可以采集下来。

下面的代码先新建了两个列表变量，“链接名称”用于保存采集到的链接文本，“链接地址”用于保存采集的链接网址，使用“获取网页代码”的程序块来获取指定网站的网页源码，网页源码保存在Html变量中，因为访问网页需要一定的时间，所以在获得Html代码前一直等待，我们把Html变量拖放到“遍历网页代码”程序块中，网页中链接的标签为A，遍历的每个链接（标签为A的元素）会保存在“元素”变量中。



网络数据采集——获取元素属性

网页代码中的所有对象都是以元素（DOM）形式存在，网页上显示的东西都可以通过元素的属性来获取。

元素 元素 的 文本 ▾

代码(html)
 文本(text)
 链接(href)
 网址链接(href)
 值(val)
 图片(img.src)
 属性名(attributes)
 子元素
 父元素
 子元素个数
 新建属性名

代码 Html 中标签为 A 元素的 文本 ▾

上面的两个程序块都可以获取元素的属性，属性包括元素的代码、文本、链接、等，不同标签的元素拥有不同的属性，如果元素存在其他属性，我们还可以通过新建属性名来添加。

下面的代码获取百度首页的图片地址，并且把导入为当前精灵的造型：

获取网页 `https://www.baidu.com/` 的代码 Html

直到 Html 前都等待

新建“图片”变量用来保存图片地址

脚本变量 图片 ▾

将变量 图片 ▾ 的值设定为

取得IMG（图片标签）
 图片地址

代码 Html 中标签为 IMG 元素的 图片 ▾

导入图片造型 将 `https:` 图片 ◀▶ 拼接 到精灵 Sprite ▾

网络数据采集——元素标签（选择器）

要在网页代码中找到想要的元素，必须知道元素的标签，获得标签的方式很多，可使用超文本标签，也可以使用选择器。

元素可以通过超文本标签来获取，常用的HTML元素标签（更详细的内容可以在百度百科上搜索“HTML标签”）：

<html>	定义 HTML 文档。
<title>	定义文档的标题。
<body>	定义文档的主体。
<p>	定义段落。
<form>	定义供用户输入的 HTML 表单。
<input>	定义输入控件。
<textarea>	定义多行的文本输入控件。
<select>	定义选择列表（下拉列表）。
	定义图像。
<a>	定义跳转或链接。
	定义无序列表。
	定义有序列表。
	定义列表的项目。
<table>	定义表格。
<tbody>	定义表格中的主体内容。
<tr>	定义表格中的行。
<td>	定义表格中的单元。
<video>	定义视频。

网络数据采集——选择元素

我们可以通过标签加上属性和索引来更精准地选择元素，而且系统还提供了通过关键字的来选择元素。

代码 **Html** 中标签为 `input` 元素的 值 ▾

取得第一个输入框的值

代码 **Html** 中标签为 `input.c1` 元素的 值 ▾

取得属性class为c1的第一个输入框的值，在标签后面加“.xxx”表示class属性为xxx的所有元素。

代码 **Html** 中标签为 `input.eq(2)` 元素的 值 ▾

取得第三个输入框的值，在标签后面加上“:eq(n)”表示取得标签的第n+1个元素，eq参数缺省时就是第一个元素。

“input.c1:eq(2)”表示所有class属性值为c1标签为input的第3个元素。

所有的标签和属性我们都可以查看网页代码来获得，但是很多时候我们希望通过看到的网页文字来选择和定位元素会更加直接，系统提供了这个功能程序块。

获取 `https://www.baidu.com/` 的网页代码 **Html**

代码 **Html** 中关键字为 `登录` 的 对应元素 对应标签

显示消息框 元素 对应元素 的 网址 ▾

网络数据采集——采集数据案例(1)

我们通过一个网页采集案例来讲解采集数据的过程，下面的代码实现采集百度实时热点排行榜的内容。

我们先打开百度实时热点排行榜的网页（网址可能会随时间的推移而变化）：<http://top.baidu.com/buzz?b=1&fr=topindex>



我们找到内容的关键字“瑞德西韦临床试验”来获取元素的标签，我们得到标签“A.list-title:eq(0)”，再通过标签“A.list-title”来定位所有相同标签的内容，代码后面两个程序块是用来验证取的标签是否能准确地定位元素的。

获取 `http://top.baidu.com/buzz?b=1&fr=topindex` 的网页代码 **Html**

代码 **Html** 中关键字为 `瑞德西韦临床试验` 的 **对应元素**
对应标签

显示输入框 提示 默认值 **对应标签** 输入值

代码 **Html** 中标签为 **对应标签** 第 **1** 个 **元素** **总数**

显示消息框 **元素** **元素** 的 **文本**

网络数据采集——采集数据案例(2)

下面的代码块通过标签“A.list-title”找到所有的实时热点的标题了，遍历网页代码中所有包含“A.list-title”的元素，取得元素的文本后，将内容朗读出来。在“遍历代码”程序块中，我们是通过“新建标签关键字”来输入“A.list-title”。



我们取得网页内容后可以根据自己的需要来处理这些数据，可以结合我们前面讲的数据库知识把它保存到数据库中，从而与其他网页或程序进行数据交换和互动。



网络数据采集——提交数据 (Ajax)

很多情况我们需要提交数据给网页服务器，网页服务器通过我们提交的数据来生成结果并返回给我们。

提交网页有两种方式：一种是GET，一种是POST。两种方式的区分简单来说就是GET把数据附加在网址的后面，以“？”号分隔网址和数据，数据以“名称=值”的形式填入，一次性提交给服务器；而POST方式是分两次访问服务器，第一次请求，请求同意后再把数据提交给服务器。GET的方式由于是以网址形式提交，所以长度有限，不适合长数据提交。下面两个代码块，第一个是以GET方式提交数据，第二个是POST方式提交，返回的结果都是一样的。

提交网页

`https://api.heweather.net/s6/weather/now?key=4&location=auto_ip&lang=null`

方式 `get` 参数名 `key` 参数值 `4` 返回结果

直到 返回结果 前都等待

显示消息框 `JSON` 返回结果 转文本

提交网页 `https://api.heweather.net/s6/weather/now` 方式

`post` 参数名 `key` `location` `lang` 参数值

`4` `auto_ip` `null` 返回结果

直到 返回结果 前都等待

显示消息框 `JSON` 返回结果 转文本

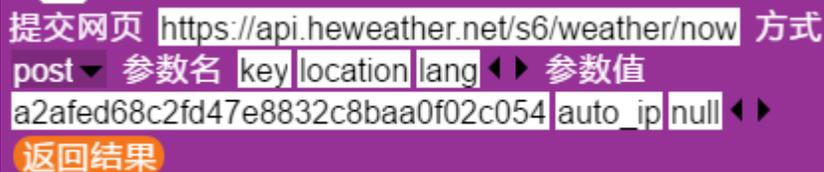
网络数据采集——处理JSON数据 (1)

通常服务器会返回一段用“{}”包裹的结构化的数据给我们，我们把这种结构化的数据称为JSON数据。

JSON数据是由指定格式的字符串构成的，格式如下：

```
{ “名称1” : “数据值1” , “名称2” : “数据值2” }
```

用来保存JSON数据的变量我们称为JSON对象，JSON对象中的值是如何存取的呢？我们通过下面的程序块得到一个JSON数据，从数据中，我们可以看到JSON数据是嵌套在一起的，就是JSON数据中又包含另一个JSON数据，而且数据中还有用“[]”包裹的JSON数据，我们称为JSON数组。



```
提交网页 https://api.heweather.net/s6/weather/now 方式  
post 参数名 key location lang 参数值  
a2afed68c2fd47e8832c8baa0f02c054 auto_ip null  
返回结果
```

```
{"HeWeather6":[{"basic":{"cid":"CN101240701","location":"赣州",  
"parent_city":"赣州","admin_area":"江西","cnty":"中国","lat":"25",  
"lon":"15","tz":"+8.00"},"update":{"loc":"2020-02-08 ","utc":"2020-02-  
08"},"status":"ok","now":{"cond_code":"104","cond_txt":"阴","hum":  
"86","pres":"1013","tmp":"8","wind_dir":"西北风","wind_sc":"2"}}]}
```

网络数据采集——处理JSON数据 (2)

系统在【大数据】中提供了JSON的处理程序块，共有四个：



我们可以通过 **返回结果 . HeWeather6** 得到下面的数据：

```
[{"basic":{"cid":"CN101240701","location":"赣州","parent_city":"赣州",
"admin_area":"江西","cnty":"中国","lat":"25","lon":"15","tz":"+8.00"},
"update":{"loc":"2020-02-08 ","utc":"2020-02-08"},"status":"ok",
"now":{"cond_code":"104","cond_txt":"阴","hum":"86","pres":"1013",
"tmp":"8","wind_dir":"西北风","wind_sc":"2"}}]
```

我们通过 **返回结果 . HeWeather6 . 0 . basic . location** 得到数组中第一个 {} 中的“basic”数据中“location”的值：“赣州”。（注意：数组第1个数据是从0开始的，第2个数据索引是1，以此类推）。如果我们要得到当前风向的数据，可以通过以下代码：

返回结果 . HeWeather6 . 0 . now . wind_dir

上述程序块都是通过 **[]** 和 **[0]** 的嵌套组成的，我们可以通过运行JSON程序块的项目演示来查看运行效果。

Web网页服务器——基础知识

现在很多信息服务和物联网控制都是通过网页来的实现的，AIScratch也提供了网页服务器的基本功能。

我们在电脑、手机、电视和联网设备上的浏览器都是支持访问以HTTP打头地址的网页内容，以HTTP打头的字符串（形如 `http://www.baidu.com/logo.jpg?abc=123&k=1`）我们称之为网址，其中 `http` 是通讯协议（相当于电脑间通信的统一的语言，除了 `http` 还有 `https`、`FTP` 等） `www.baidu.com` 我们称为主机域名，是访问某个主机的地址，它会在访问时转换成IP地址（形如 `127.0.0.1`），后面的 `logo.jpg` 是访问的文件名称，“？”后面的是参数，多个参数用“&”分隔，例子中有两个参数，参数名称分别为 `abc` 和 `k`，参数的值分别为 `123` 和 `1`。

网络上的电脑传递信息都要开放一个端口（相当于收音机或电视机的频道），而网页的默认端口为 `80`（若是 `https` 打头则是 `443`）通常情况网页端口是可以不写的，如果要显式表达出来就是加在主机名称后面，并用“:”分隔，如 `http://www.baidu.com:80/`，我们也可以用其他端口来提供网页服务，这时端口号就不能省略了。在系统中，我们可以用新建WEB服务器的程序块，来建立自己的网页服务程序，别人只需要通过网址来访问你的网页了。通常我们的本机电脑的主机IP地址是：`127.0.0.1`（或 `http://localhost`），如果在别的电脑上访问你可以使用局域网的IP或者外网的IP地址。

Web网页服务器——建立服务器

现在很多信息服务和物联网控制都是通过网页来的实现的，AIScratch也提供了网页服务器的基本功能。

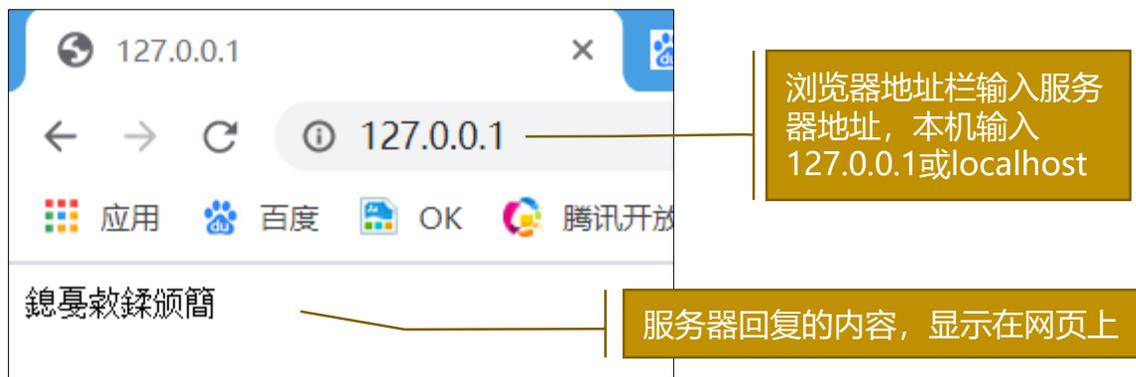


要提供给用户网页信息服务我们先要建立一个WEB网页服务，服务需要使用到一个端口号，默认的网页端口号为：80，在建立服务后，如果没有用户访问，“请求1”和“回复1”两个变量的值是空的，不能使用，只有当用户访问后两个值才能使用，因此所有处理的代码块是放到 **当端口 80 收到网络请求** 程序块下面的。我们可以根据用户提供的网址和参数来进行相应的程序处理，并在处理后返回信息给用户的浏览器。网页服务建立后，我们做一个简单的网页回复处理：



Web网页服务器——访问服务器

客户可以通过任何一个上网设备来访问服务器，访问时输入服务器的IP地址就可以了。



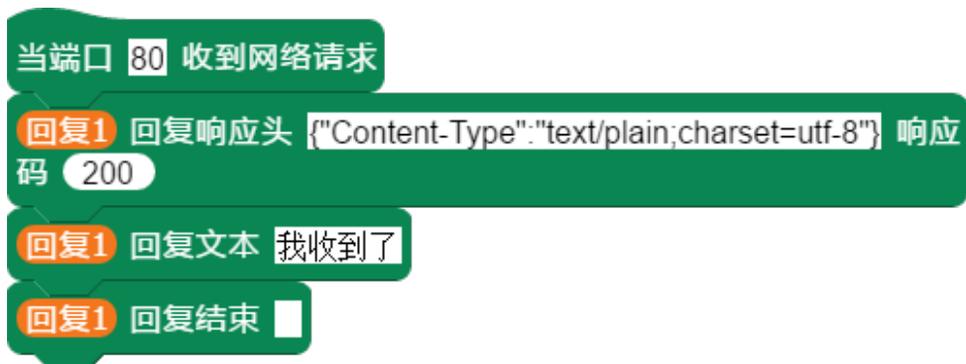
我们在浏览器输入服务器的IP地址，如果是本机我们可以输入127.0.0.1或localhost，如果客户机与服务器处于同一内网，我们可以输入客户机的IP地址（windows系统的查看方式：在客户机上点击开始—运行—输入cmd并运行—在cmd命令行输入ipconfig/all）如果客户机与服务器处于外网（互联网）则使用外网IP访问。如果客户机与服务器不是同一台电脑，不管是外网还是内网，如果我们无法访问服务器，我们就要查看服务器的防火墙是否开放了服务程序指定的端口号（本例中是80），如果是外网，我们通过百度搜索关键字“IP”来得到本机的所在网络的外部IP地址，同时如果需要访问本机服务器，还需要在外网路由器上设置端口映射，把服务端口映射到本机服务器的内部IP地址上，否则外网电脑是无法访问的。

Web网页服务器——回复响应头

服务处理程序可以对客户的访问进行相应的程序处理，还可以对客户的访问进行回复，在回复之前需要设置响应头。



我们刚刚这个例子回复用户“我收到了”但是在网页端却出现了乱码，这是因为我们没有告诉浏览器我们的回复文本的编码格式，我们可以在回复文本前回复一些文档的辅助信息，这些信息我们称之为响应头(Header)，响应头就是告诉浏览器一些背景信息，比如文档的类型格式、长度、编码、来源等。下面的程序块告诉浏览器我们将回复一个文本并且编码格式是utf8的：



我们再次在浏览器访问127.0.0.1，我们看到了正常的文本了。

Web网页服务器——设置响应头

响应头还可以单独设置，下面两个程序块的作用是一样的：

回复1 设置响应(请求)头 Content-Type 值
text/plain;charset=utf-8

回复1 回复响应头 ["Content-Type":"text/plain;charset=utf-8"] 响应
码 200

服务端
内容类型(Content-Type)
内容长度(Content-Length)
语言(Content-language)
压缩编码(Content-Encoding)
跳转(Location)
最后修改时间(Last-Modified)
缓存控制(Cache-Control)
过期(Expires)
连接(Connection)
响应时间(Date)
服务端名称(Server)
刷新(Refresh)
下载打开(Content-Disposition)
客户端缓存(Set-Cookie)
分块传输(Transfer-Encoding)
客户端
接收类型(Accept)
接收语言(Accept-Language)
主机(Host)
来源(Referer)
浏览器(User-Agent)
压缩编码(Accept-Encoding)
缓存时间(If-Modified-Since)
缓存(Cookie)
请求时间(Date)
新建属性名

程序块可以设置响应头和请求头的全部属性，这些属性的具体含义我们可以在百度搜索关键字“响应头header详解”来获得更多的相关知识，我们这里不详细展开讲解，可以参看<https://www.jianshu.com/p/ee2e569a30c4>。

服务端的属性对应的是“回复”变量，客户端的属性对应的是“请求”变量，使用时注意区分。如果有自定义的属性或新增的属性，我们还可以通过“新建属性名”来添加新的属性。

通过响应头来告知客户端浏览器一些回复的基本信息，通过请求头来告知服务器一些请求的基本信息。

响应头和请求头统称报文。

Web网页服务器——获取请求信息

客户在浏览器输入网址后，服务程序会得到一个请求变量，我们根据请求变量中的报文来获取客户端的一些基本信息。

请求 请求1 的

- 请求头(headers)
- 请求方式(method)
- 请求对象(socket)
- 请求链接(href)
- 请求主机(host)
- 请求路径(pathname)
- 请求参数(query)
- 请求端本地地址(localAddress)
- 请求端本地端口(localPort)
- 请求端远程地址(remoteAddress)
- 请求端本地端口(remotePort)

我们可以通过“请求”程序块来获得客户端提交过来的请求头信息，并根据这些信息来做相应的程序处理和回复。在这些信息中，有些不是文本格式，而是对象格式，我们可以通过【大数据】中的结构化数据程序块来处理它们。

请求 请求1 的 headers

[object Object]

我们看到“请求”程序块返回的“headers”属性是object格式的，这时我们通过 JSON 转文本 来查看它返回的内容。

JSON 请求 请求1 的 headers 转文本

```
{"user-agent": "Mozilla/4.0 (compatible; MSIE 9.0;
```

还可以通过 `headers.host` 来取得属性的值。下面的程序块获取了客户端的IP地址：

请求 请求1 的 headers . host

192.168.1.1

Web网页服务器——处理客户请求

我们根据请求的链接所提供的信息来处理客户端的请求，并且回复客户相应的信息。



以上是我们获取请求信息各个部分的程序块代码，如果还需要获得更多的信息我们可以通过解析请求头的信息来获得。下面我们看一个例子，用网址来控制精灵的动作。在浏览器地址栏输入“`http://127.0.0.1/旋转?d=右转&s=90`”，d代表方向，s表示度数。



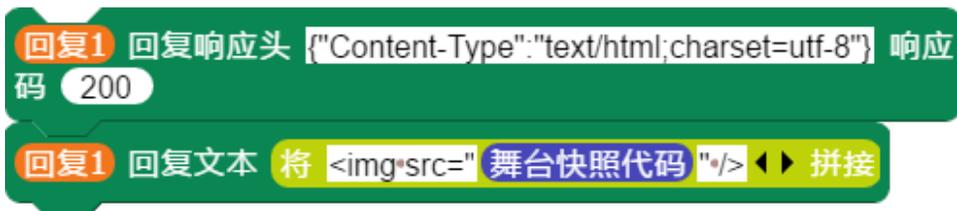
Web网页服务器——回复文本

我们回复客户端请求的内容可以是文本、网页代码、脚本和文件，回复类型不同客户端浏览器处理方式也不同。

如果我们简单地回复用户文本，我们可以使用以下程序块：



我们使用默认的响应头来告诉客户端浏览器，我们将发送的是一个文本，并且文本编码是utf-8格式的，指定编码格式主要是防止客户端显示中文是乱码。我们可以回复多条文本，在“回复结束”程序块执行前，回复的文本是不会发送给客户端的，客户端浏览器一直处于等待状态，当执行“回复结束”时服务程序会发送响应头和所有的文本给客户端，并结束此次请求。如果我们想告诉客户端浏览器发送的文本是网页代码，我们可以把响应头设置成 { “Content-Type” : “text/html;charset=utf-8” }，浏览器收到文本后会解析成HTML显示在网页里。下面的代码把舞台快照发送给客户端，并在浏览器显示出来：



Web网页服务器——回复文件

回复文本需要指定文件类型，系统可以自动根据扩展名来判断文件类型，同时我们也可以显式指定文件的类型。

下面的程序块给客户端发送一个图片文件：

当端口 80 收到网络请求

回复1 回复文件 c:\1.jpg 类型 自动

代码块在收到客户请求后，回复了c:\1.jpg这个图片文件给客户端。文件类型对应就是响应头中“Content-Type”的值，即使是同一个文件，如果我们设置的类型不同，客户端浏览器处理的方式也不一样，下面我们举个例子来说明，我们在记事本编辑一个文本文件，内容是“Hello World!”，我们把它保存到C盘下，并命名为“abc.txt”，我们使用下面的四种文件类型设置来观察显示效果（可以右键打开“回复文件”程序块的项目演示）：

回复1 回复文件 c:\abc.txt 类型 自动 Hello World!

回复1 回复文件 c:\abc.txt 类型 文本 Hello World!

回复1 回复文件 c:\abc.txt 类型 网页 **Hello World!**

回复1 回复文件 c:\abc.txt 类型 下载  另存为

我们可以用程序块 **窗口打开** <http://127.0.0.1> 来打开内置的浏览器。

Web网页服务器——应用场景

通过WEB服务我们可以对接各种应用场景，与服务程序产生互动，比如用手机控制精灵或硬件、与网站互动、与远程设备互动、语音控制与程序的互动。

我们要让客户端访问服务程序，必须有一个可以被访问的IP地址，通常内网IP地址只能被处于同一内网的设备访问，如果要让外网客户访问，我们必须有一个固定的外网IP，但是通常我们家用宽带的IP都是动态的，就是IP地址会经常变动，所以无法使用。我们的解决方法是使用一些内网穿透的软件，比如花生壳（下载地址<https://hsk.oray.com/download/>）内网穿透软件会给我们一个固定的网址来访问我们的内网服务器，这样任何外网设备和浏览器都可以使用服务程序，花生壳的使用方法可以参看官网帮助

（<http://service.oray.com/question/8146.html>）或参看软件帮助系统。

有了外网的地址我们就可以与外部设备和用户连接了，使用任何电脑和手机的浏览器都可以，还可以使用各大手机品牌的语音助手来实现语音控制。下面我们举例来说明手机如何访问我们的程序服务，在访问前我们确保已经安装好了内网穿透软件，并且已经获得了外网访问的固定地址了，例子中我们只用内网IP来测试服务程序，与外网访问的效果是一致的。