（亲子共读）

第一部分
欢迎来到编程世界

欢迎来到编程世界，我们将一同通过Scratch图形化编程工具来探索世界、发现世界、创造世界。编程的学习不是刻板、枯燥的，相反，它趣味横生，能够玩中学、学中玩、做中学、学中做、用中学、学中用、创中学、学中创。在编程的世界里，我们可以无拘无束地表达自我，无论是漫无边际的梦境，还是遥不可及的梦想，统统都可以实现。在编程的世界里，不需要担心犯错，因为排错本身就是一件趣事，Debug让我们思考，让我们成长。

我们构思，我们分析，我们拆解，我们编程，我们探索，我们尝试，我们改进，我们总结，我们收获，我们分享，我们喜悦。打开计算机，现在出发！

Scratch

第1章

如何学习编程

1.1 少儿编程到底学什么

很多家长朋友让孩子学习编程，是为了培养逻辑思维能力，担心人工智能时代下逻辑思维能力差的孩子会被淘汰。

毫无疑问，编程是一门非常适合用来锻炼逻辑思维的学科。因为编写程序的过程就是不断思考和运用逻辑的过程，所以说编程对逻辑思维的培养非常有效和重要。

但是，这些也恰巧是计算机最擅长的，未来在大数据和人工智能面前也是最容易被替代的。

那么，什么是不容易被替代的呢？

人类的品质，探索的精神，创新的意识，文明的艺术，更高的思维方式。

我认为少儿编程的重点不在于学习编程软件的使用和编程语言，而在于训练思维。让孩子在愉悦的学习环境中尽情地表达内心想法，通过项目制的学习方法掌握分析问题、解决问题的能力。

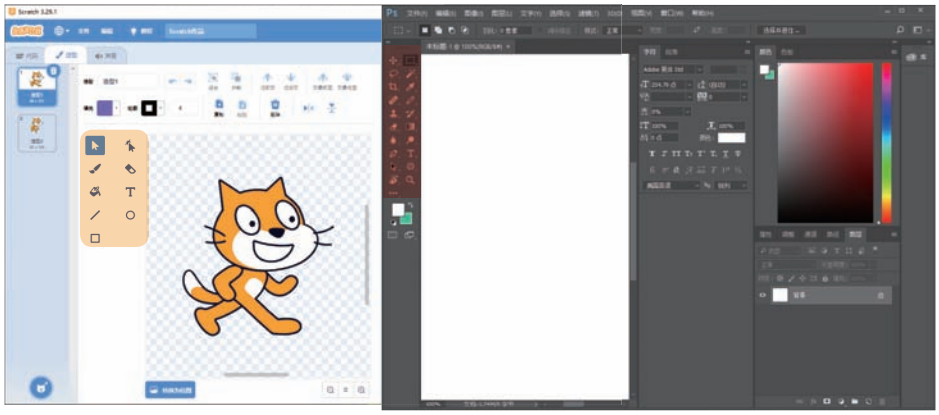
以下将从3个方面阐述我认为的少儿编程。

1. 学习什么 ➲ 2. 培养什么 ➲ 3. 训练什么

1. 学习什么

1）学习软件的使用方法

未来，我们将会越来越多地使用各种软件来更高效地完成工作和解决问题，例如Word、Excel、PPT，以及图像处理工具、视频剪辑软件、ChatGPT等。而它们的一些操作在很多地方是相通的。

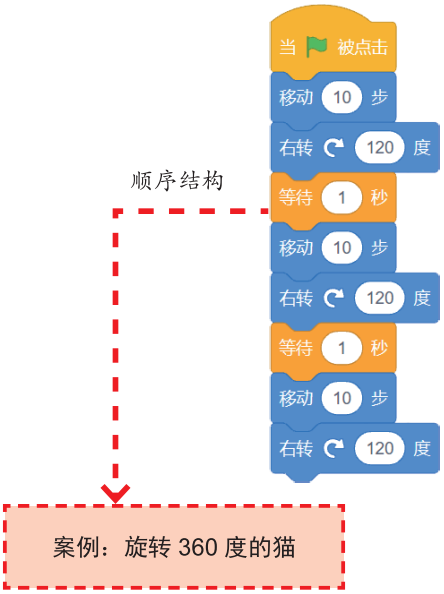


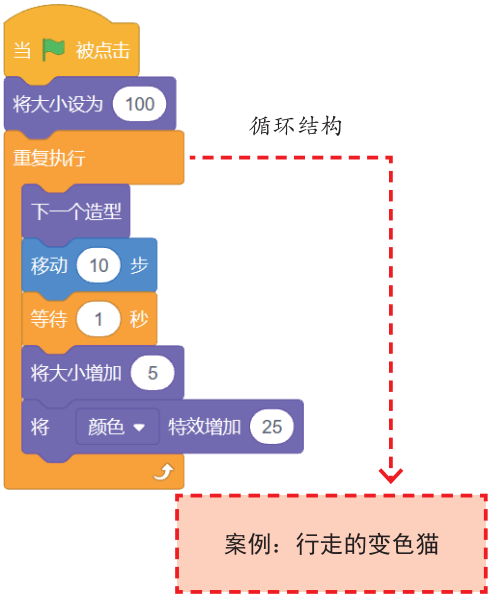
Scratch 软件界面　　Photoshop 软件界面

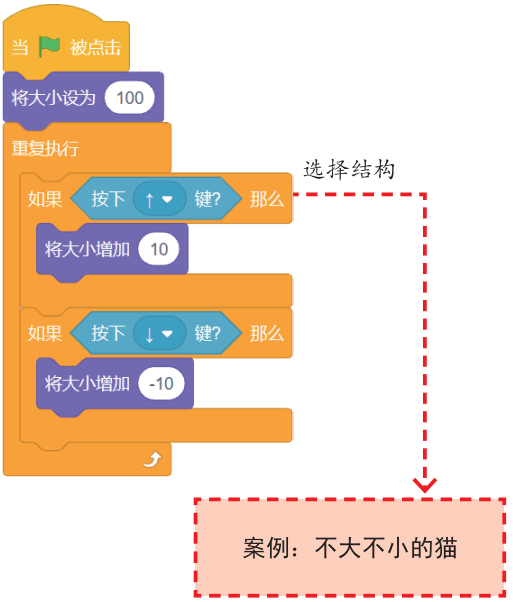
Scratch 的软件界面与 PhotoShop 的软件界面就有几分相似。

2）学习编程知识

编程知识的学习只是少儿编程学习中的一小部分。学习流程图，可以掌握描绘编程应用逻辑的能力；学习编程语言语法，可以掌握代码编写的能力。学习顺序结构、循环结构、选择结构、变量、数组、函数等一系列的编程知识和编程技巧，通过运用这些编程知识和编程技巧，可以设计简单的程序，帮助我们解决一些重复性或流程化的工作。







3）学习跨学科应用

将数学、物理、化学等学科知识运用到一个个实实在在的项目上，融合相关知识来模拟现实场景、解决问题或者进行发明创造。这可以赋予知识更蓬勃的生命力，而不仅仅是停留在纸上作业或考试中。

案例：趣味僵尸大战试卷

创意数学计算练习测试卷

2. 培养什么

1）培养自律性

孩子大部分时间是生活在他律的状态，被管着上课不能交头接耳，被要求作业按时完成，被规定玩游戏不能超时，等等。编程可以很好地将他律变成自律，因为在程序的世界里不需要外人评说，程序本身会直接告诉你对与错，运行的结果也会给出实际反馈。

培养一个孩子的自律性，需要制定适合他年龄的规则，培养他遵守规则的意识，以及适当地延迟满足他的需求。

每一门编程语言都有自己的规则，我们无法讨价还价，程序的运行也会完全按照规则来执行。编程通过严谨的逻辑判断和程序规则帮助孩子不断地提升自律性。

通过编程制作项目、设计游戏都不是一蹴而就的，它需要一个过程。例如，想玩一款自己脑海中设计的游戏，需要先进行**策划-分析-编程-测试-调错-优化**这一过程，然后才能玩游戏。这个过程可以给予孩子很好的延迟满足感。所以说编程学习可以锻炼孩子的自律性。

自律性对孩子的成长和成绩的提高都非常重要，至少在小学和初中阶段，一个自律的孩子通常其学习成绩都很优秀。

2）培养高效的自主学习能力

我们获取知识的路径越来越便捷，无论是大人还是孩子都可以快速地通过网络或者书籍获取各种知识，这个时候的竞争力就体现在自主学习的能力上。

自主学习能力表现在两个方面：一是主动地学习，二是高效地学习。

在高中阶段高效地学习非常重要，这也是成绩从自律性拉开的关键。一个自律的人可

以很好地做到主动学习，尤其学业压力非常重的时候，如果没有好的学习方法，将会学习得非常辛苦且不见成效。到了大学阶段，主动学习将是核心，孩子将会有大把的自由支配时间，如果可以做到主动而高效地学习，那将会是人生的一笔宝贵财富。

孩子在幼儿园、小学、初中、高中一直都有家长的督促、老师的管理，虽然学习压力和负担逐渐上升，但是在这些阶段，孩子无须直面社会的挑战。然而，在大学这个最邻近社会挑战的学习阶段中，没有了家长时刻的督促，也没有了老师严格的管理，毫无自主学习能力的孩子虽然可以轻松愉快地度过四年大学时光，但接下来迎接他们的是比高考还要严峻的就业挑战，要面临的是比高中学业还要辛苦的工作。因此，这就需要我们的孩子提前具备自主而高效的学习能力，如此才能在这样的成长路径下成为佼佼者。

3）培养想象力和创造力

孩子的想象力是无穷的，哪怕是梦境也是一部有趣的动画片。编程可以将孩子的想法结合音乐、绘画通过游戏、动画、模拟等一系列的方式生动地表达出来。

创新创造的意识会在孩子不断表达和实现内心想法的过程中得到充分的锻炼和提高。具备了自律性和高效学习能力的孩子，再融合上想象力和创造力，将会是未来极具竞争力的人才。

4）培养表达力

通过项目进行中的团队协作，可以锻炼孩子的沟通协调能力。将自己的项目进行讲解和展示，可以提升孩子的表达能力。

自己创造的趣味游戏或者炫酷项目更容易激发孩子的分享欲，通过追逐兴趣和内在的成就感，让孩子不断地感受到分享带来的乐趣，在潜移默化中锻炼表达能力。

5）培养耐心和细心

编程项目不是一蹴而就的，会遇到各种各样的问题，需要反复调整，程序的优化和错误的寻找是一个极需要耐心和细心的活。

不断思考优化和排错的过程可以逐步培养孩子的耐心和细心。

3. 训练什么

1）计算思维

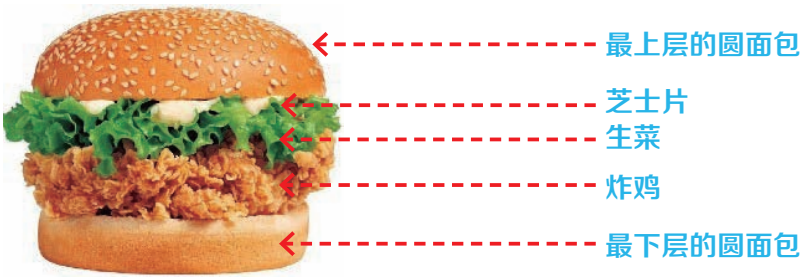
计算思维是一种具有逻辑性和抽象化的利用计算手段科学地解决问题的能力，主要包括4个方面：分层思维、模式识别、流程建设和抽象化。

分层思维

分层思维帮助我们将大项目拆分成若干个小项目，将复杂的大问题拆解成一个一个的小问题，让复杂的大部分变成容易理解和解决的小部分。通过完成小项目、解决小问题来完成大项目、解决复杂的大问题，使整体变得更加易懂和简单。

如同制作汉堡，乍一看是一个多层级的复杂项目，但是当我们运用分层思维将它拆解后，就会变得更加简单。

首先，将汉堡分层划分出它的组成部分。



然后，完成每一个组成部分，再将各部分组合起来，一个汉堡就做好了。

模式识别

任何事物都有相似性，模式识别正是教会我们找到事物之间的共性规律。通过总结以往经验并进行类比推理，利用这些相同的规律去解决问题。当我们将大项目、复杂的大问题分解为小项目、小问题后，通常可以在这些小项目、小问题中找到模式和规律。

一个典型的例子就是多个数字的加法。我们都会运算两个数字的小加法，那么把这些小加法组合起来，就可以用相同的加法逻辑来解决更大范围的加法问题。

9+10+8+6+7=?

第一步：先分解为4个式子。

9+10=?

? +8=?

? +6=?

? +7=?

第二步：运用我们掌握的加法逻辑逐个进行计算。

9+10=19
19+8=27
27+6=33
33+7=40

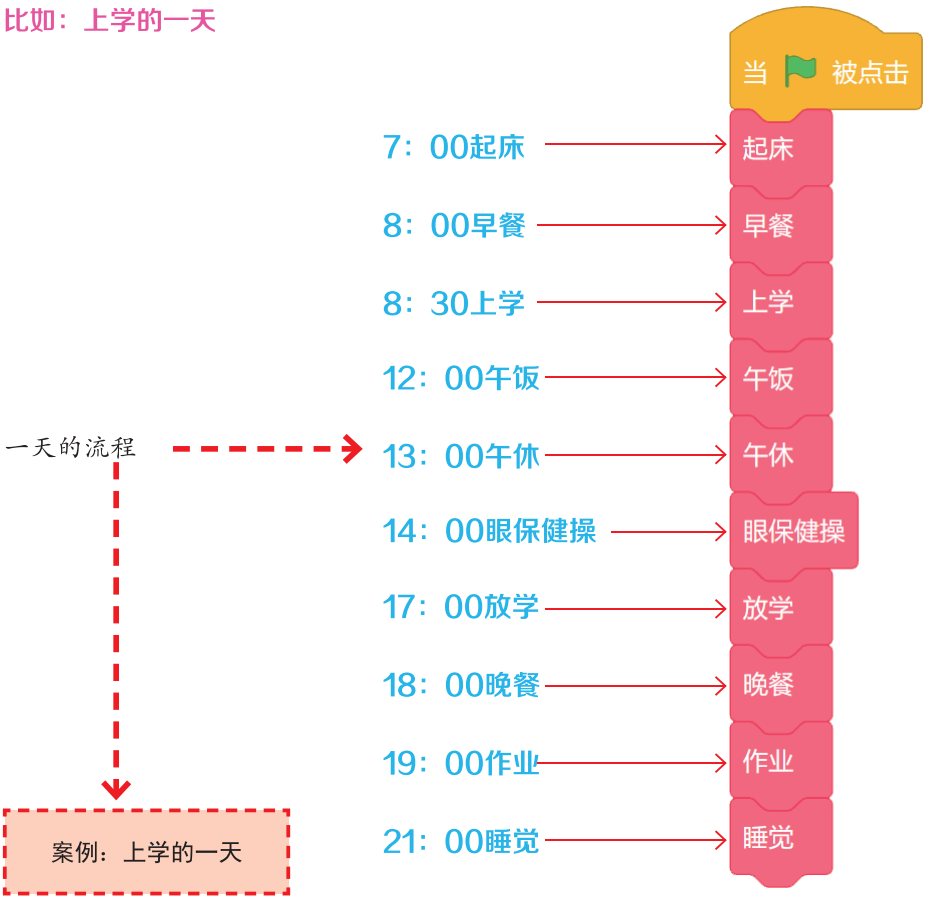
或许上面的每一个加法式子我们之前都没有计算过，但是加法法则的模式和规则是相同的，所以我们可以快速地计算出每个小式子的答案，通过一层一层的计算，得到最终的答案。

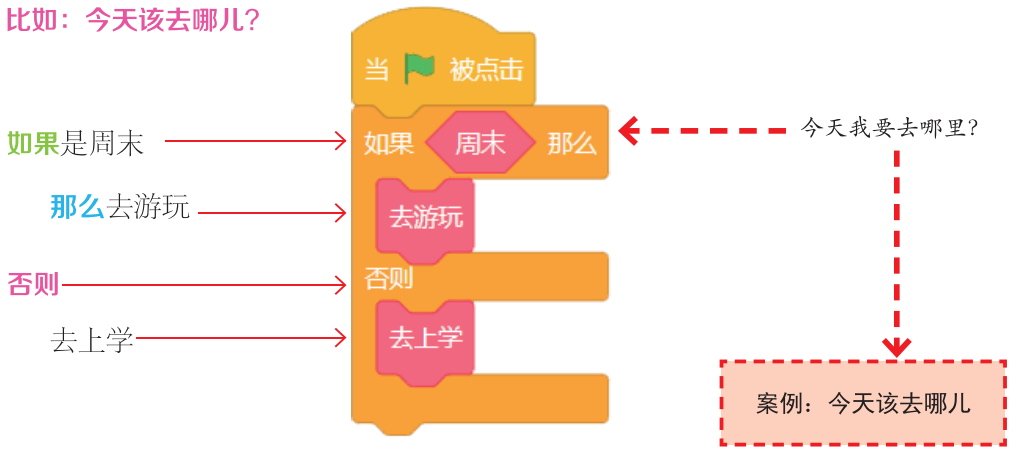
- **流程建设**

流程建设是一步一步解决问题的过程，按照一定的顺序完成一个又一个的任务。设计程序的每一个步骤，编写程序的每一行代码，都是流程建设的过程。

我们已经在日常生活中运用到了流程建设。

比如：上学的一天





● **抽象化**

抽象化是关注关键信息，将重要的内容提炼出来，忽略不必要的细节。掌握了抽象化的能力，我们就可以将一个解决方案应用于其他事物中，制定出解决方案的总体思路。

例如，或许我们并不认识所有贝类，也叫不出它们的名字，但是关注到它们都有坚硬的贝壳、柔软的肉质，并生活在水域，因此可以进行抽象化，把它们统一叫作贝类。当我们知道如何烹饪花甲的时候，对于其他贝类，通过它们都是贝类这个关键信息，就可以忽略它们的外形和名字，得出它们能用与花甲相同的方式进行烹饪。

2）各种思维的培养

学习编程还能培养描述性思维、批判性思维、类比思维、程序思维、逆向思维、发散思维、组合思维、求异求同思维、因果思维、建模思维等。

既然少儿编程的教育理念是这样的，那么整个学习过程就应该围绕这个教育理念来开展，这样才不会让少儿编程教育变成知识灌输式的学科。少儿编程的关键在于教学的形式和过程。

1.2 学编程的好处

尽管我不认为每个人都需要精通编程，但我还是鼓励每个孩子都接触编程。

学习编程并非为了成为程序员，而是为了培养缜密的思考方式和严谨的思维逻辑。程序运行的对与错的最终判定者是计算机，它不会投机，也不会主观判定；程序里处处是因果，因而需要思考各种因和所导致的各种果，需要全面思考、清晰分析。

学习编程对孩子的帮助很大，可以归纳为以下10点：

1 强化思维逻辑能力。

编程是一种高度逻辑化的思考方式，它需要编程者不断地分析问题、拆解问题、探索方案、验证方案，并将其转化为计算机可以理解的程序语言。整个编程过程需要不断地进行推理和分析，以实现一个清晰、有效的解决方案。

编程需要严谨的思维逻辑、多维度分析、多方面组合，最终形成一个完整可运行的程序。这种思维方式可以帮助孩子们在日常生活中更好地进行逻辑思考，更清晰地理解因果关系，思考多维度因素，考虑各种情况，更系统地处理复杂信息。

2 培养专注力和细心度。

编程是一种需要高度专注和细心的活动，孩子在编程过程中需要专注地思考实现方案，细心地编写每一行代码，细心地排查问题和专注地修复各种错误。这种思维练习可以帮助孩子养成细致专注的思考习惯，提高错误排查能力，增强留心细节的观察力，从而提高专注力和细心度。

编程中少写一个字母、忽略大小写、末尾缺一个分号，都会造成程序错误。因此，在编写程序的过程中，排错是不能敷衍了事的，这个过程能有效改正孩子马虎行事的坏习惯，避免当个“差不多”先生。

3 提高耐心，建立延迟满足感。

当父母的一定遇到过这种情况，带孩子去吃饭，孩子怎么也坐不住，总是跑来跑去，到处吵闹玩耍。这时如果拿出手机，孩子会瞬间回到座位安静地玩游戏。

爱玩是天性，以游戏项目作为课程就是运用这种天性，让孩子在玩中学、学中玩。但是编写游戏和玩游戏是有很大区别的，编写游戏需要经历一个比较长的过程，需要**构思**、**分析**、**拆解**、**编程**、**尝试**、**验证**、**优化**等，它不是一蹴而就的，这就是一个培养耐心和建立延迟满足感的过程。

4 增强抽象思维能力。

编程需要将具象化为抽象，将现实中的问题抽象成计算机可以处理的形式，这需要孩子具备较强的抽象思维能力。编程中的抽象思维表现在对问题进行适当的简化和抽象，将问题分解成更小的单元，并对它们进行分类和组合，从而建立起一个完整的程序结构。

通过编程实践，孩子可以逐渐培养抽象思维能力，学会将复杂的问题、相似的问题进

行抽象处理。这种能力不仅可以帮助他们更好地理解和解决编程问题，同时也可以应用于其他领域的问题解决和思维创新。

5 提升信息整理、融会贯通的能力。

程序是一堆计算机指令的组合，好比我们先学汉字、学词语，然后组词造句，最后撰写文章编写书籍。程序中的基本指令就像汉字，建立对象就像组词，创建函数方法就像造句，编写项目就如同撰写文章。

编程可以帮助孩子学习如何快速获取并处理信息，学会有效地组织和管理信息，从而提高学习和工作效率，以及跨学科的运用。

6 提升孩子的沟通表达能力。

显而易见，各行各业已经离不开计算机，编程语言已然成为一种重要的沟通工具。它不但可以跟计算机沟通，还可以跟所有运用计算机的人沟通。

编程是一种普遍适用于全球的表达方式，能够帮助孩子掌握一种国际性的沟通方式。这种跨文化的交流可以帮助孩子更好地理解和尊重不同的文化和价值观，培养国际意识和沟通技能。

7 让孩子学会团队协作。

在适当的教学课程设计下，学写程序就像玩游戏闯关一样，同学们可以彼此讨论破关攻略。结对编程，你分享，我倾听；我编程，你帮忙；你设计，我体验。同时，在互动中彼此协作，学习如何利用团队的力量解决问题。比起老师在讲台上单方面地讲课，同学之间的交流和探讨更容易提高学习效率。

在团队中共同学习可以帮助孩子不断提高自己的沟通能力和协作能力，锻炼接受不同角度的提议和多维度思考的能力。这种团队合作和共同学习的经验不仅可以帮助孩子在编程领域取得成功，也可以应用于其他领域，如体育、艺术和社会活动等，从而培养孩子的领导能力和协作能力。

8 训练孩子的空间思维能力。

在编程中控制游戏人物，是训练空间思维能力的一个很棒方式。孩子仿佛自己在高低起伏的地形中穿梭跳跃，在编程世界的闯关游戏中探索，在游戏中学习，感受空间的方向和三维架构。

在编写程序时，需要考虑各种数据、代码和函数之间的关系和空间布局，这就需要孩子具备空间感知力和创造性思维，能够从不同角度考虑问题并找到最优解决方案。例如，孩子需要将不同的图形、图像和动画元素组合在一起，创建出一个完整的游戏或应用程序。在此过程中，需要考虑不同元素之间的空间关系、层次结构和交互方式，从而设计出一个美观、立体的程序。

9 增进孩子解决问题的能力。

面对一道道关卡的挑战，孩子会不断思考如何利用学到的知识、程序达到闯关的目的。一旦养成这样的习惯，在碰到生活中的问题或其他课业问题时，自然也会试着自己动手动脑去解决，父母只需要给予适当的工具和提示即可。

在编写程序的过程中，可以培养孩子解决问题的能力，例如定义问题、描述问题、分析问题、寻找方案、验证方案和评估结果。通过不断尝试和探索，可以提高解决问题的能力。这些技能和能力还可以应用于数学、科学等其他领域。

10 Learn to code，code to learn。

编程思维（Computational Thinking）就像阅读能力一样，是一项基本的能力。编写程序、控制计算机，这是一种新的学习方式。在学习编程的过程中，可以激发孩子的学习兴趣和主动性。父母再也不用逼着孩子学习数学，因为当他想要在编程项目中实现3D空间时，就需要学习立体几何；想要完成计分规则，就需要使用代数知识；想要完成角色的跳跃，就需要用到万有引力；想要完成地图设计，就需要美术功底；想要融合背景音效，就需要音乐知识；等等。编程项目中必然需要数学技巧、物理知识、艺术创造等多学科的融合。让孩子在编程的同时自主学习，就是我们所说的code to learn的理念，也是实现跨学科融合的一个好途径。

通过学习编程，不仅可以掌握编程语言和技能，更重要的是可以培养学习能力和思维方式。编程是一种全新的学习方式，通过编写程序，孩子可以探索各种知识领域，例如数学、科学、艺术和社会学等，同时也可以学习各种技能，例如逻辑思考、创新能力和团队合作等。编程还可以帮助孩子发现问题、分析问题和解决问题，并通过实践加深对知识的理解和应用。

1.3 什么是 Scratch 图形化编程

Scratch图形化编程是一种全新的程序设计语言，使用它可以很容易地创造交互式故事情节、动画、游戏、工具，还可以把自己的作品分享给其他人。使用这种计算机编程语言的目的是帮助孩子发展科技特长，培养核心思维，提升思考质量，习惯沟通协作。

与其让孩子玩游戏，
不如让孩子创编程。

Scratch图形化编程不是游戏，而是一款可以制作实用工具、文艺作品、游戏、动画等的编程工具。一旦掌握了这个工具，孩子就能自由自在地挥洒他们无限的创意，享受创造的乐趣。在中小学的教育中，通过Scratch图形化编程扩展了语文、数学、外语、音乐、体育、科学的教学深度，实现了跨学科学习。

Scratch图形化编程与Java、Python、C、C++、C#不同，因为它的形式不是基于文本的，而是一种可视化的积木式编程。无须输入任何复杂的命令和代码，孩子要做的仅仅是像搭建积木一样搭建程序块。这样的编程模式可以完全避免语法错误，使得孩子能更专注于逻辑思考和程序化的分析。

使用者可以不认识英文单词，也可以不会使用键盘，通过类似搭积木的方式就可以轻松完成编程。**避开了复杂的语法，却完美地保留了编程思维**。几乎所有的孩子都会喜欢上Scratch这个软件，并且产生编写程序的欲望。图形化编程不是密密麻麻的英文代码，它是积木式的，没有复杂的语法，有的仅仅是积木一样的凸槽和凹槽。所编即所得，一边写代码，一边看效果。

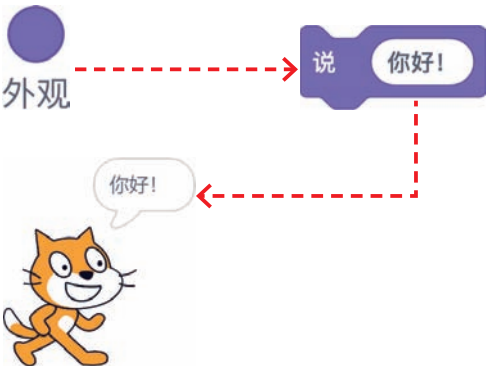
1.4 Scratch 图形化编程初体验

下面通过几个小例子来看看Scratch程序块是如何搭建的。

案例：一个积木块的Scratch程序

1. 在积木区的**外观模块**中，找到**说“你好！”**积木块。

2. 单击运行**说“你好！”**积木块，运行结果：小猫咪说“你好！”。



下面这个例子稍微复杂一点。

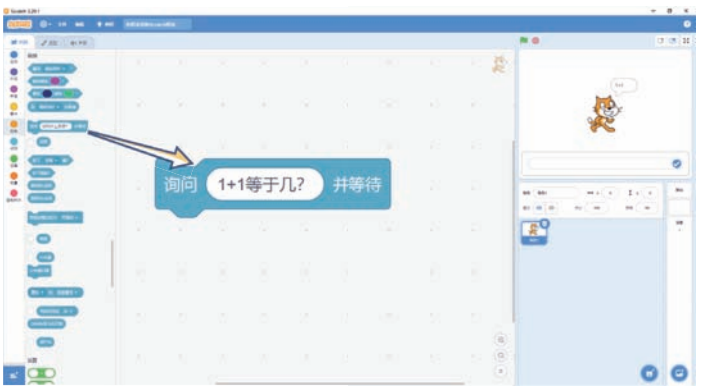
案例：多个积木块的Scratch程序

用一段小程序出一个考题**“1+1等于几？”**，通过程序判断你的回答是正确的还是错误的。分别给出正确答案和错误答案，看看程序返回的结果是什么。

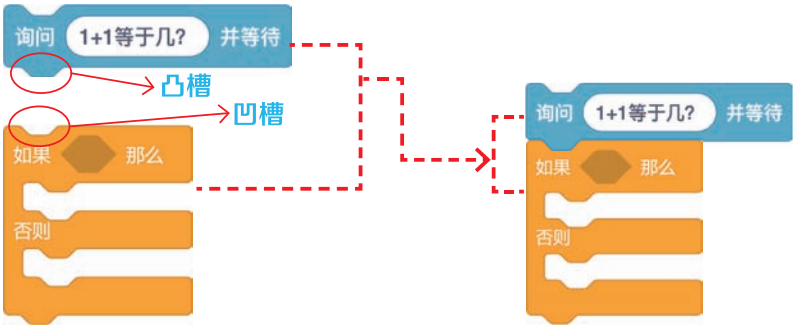
下面简单地讲解这个程序的积木块组合，逻辑思考部分在这里暂不涉及，将会在后面章节详细讲解。

1. 从**侦测模块**中找到**询问“你叫什么名字？”并等待**积木块。

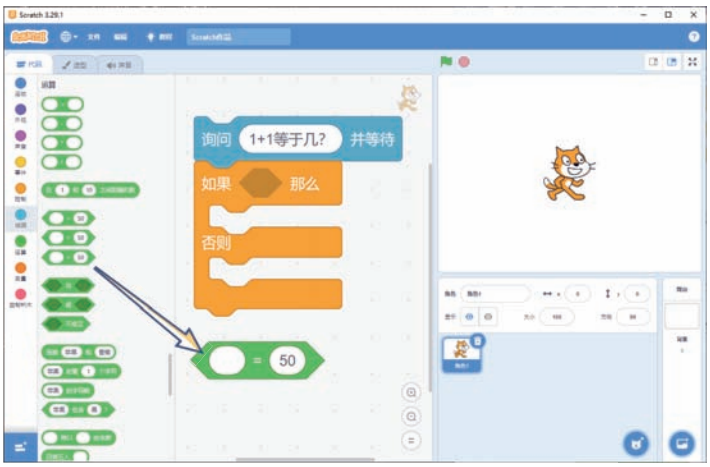
2. 将积木块拖动到脚本区，再将积木块中的内容修改成“1+1等于几？”。



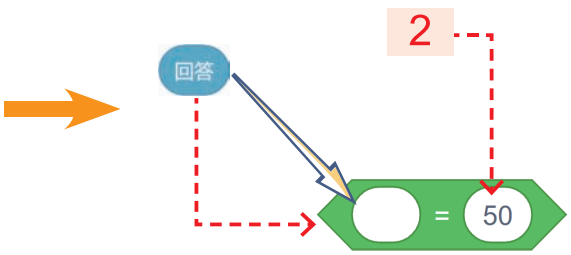
3 从**控制模块**中找到**如果……那么……否则**积木块，拖动到脚本区并拼接在**询问**“**1+1等于几？**”**并等待**积木块的下方，凹槽与凸槽拼接。



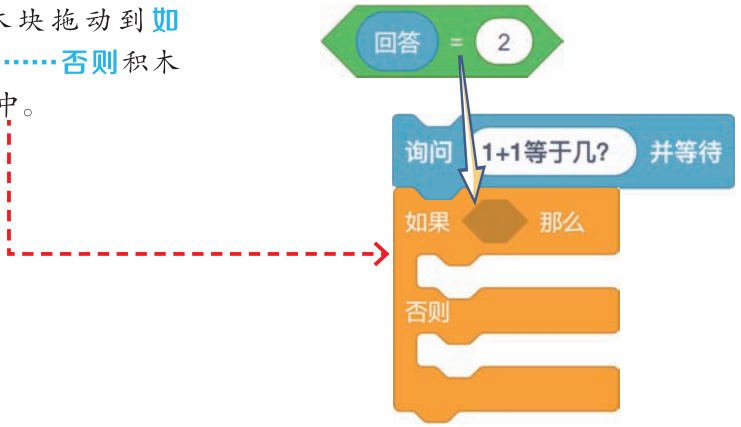
4 从**运算模块**中找到**等于**积木块，并将它移动到脚本区。



5 在**侦测模块**中找到**回答**积木块，将它拖动到**等于**积木块左边的白色椭圆框中，然后在**等于**积木块右边的白色椭圆框中输入数字“2”。



6 将等于积木块拖动到如果……那么……否则积木块的方块框中。



翻阅秘籍

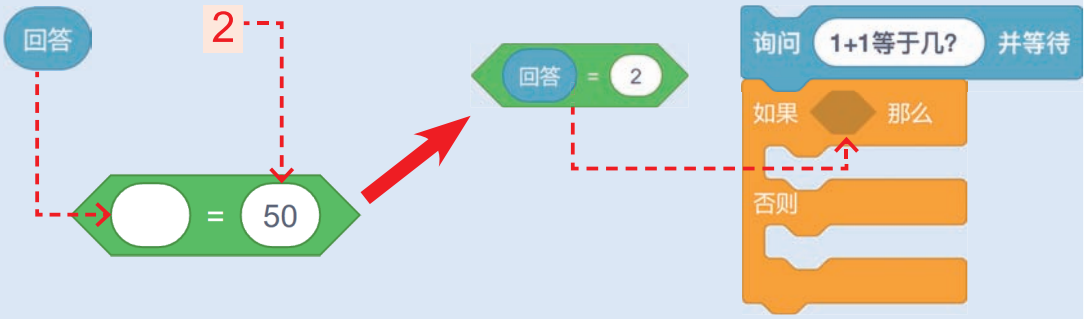
椭圆形积木块放入椭圆形框中，六边形积木块放入六边形框中，这就是Scratch的语法，形状不匹配是放不进去的。

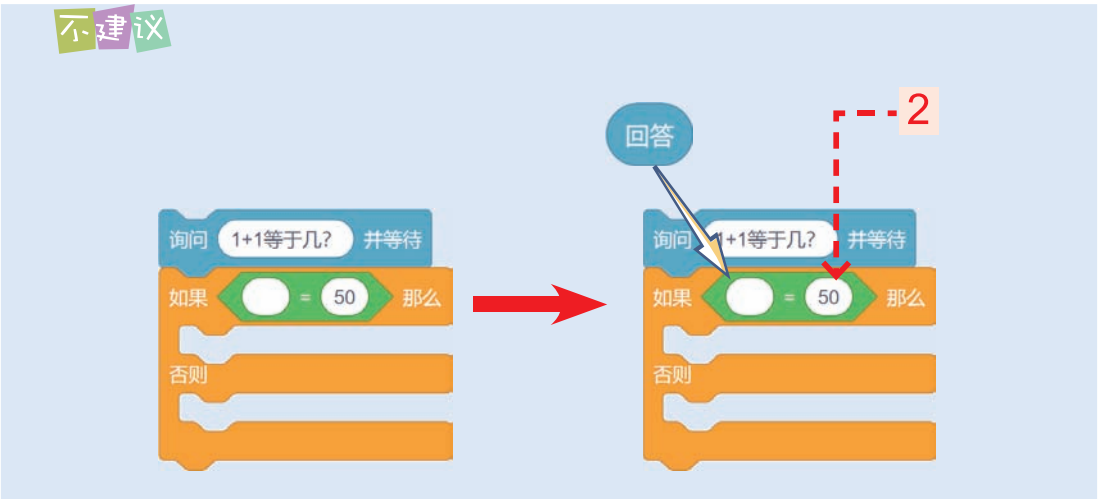
提示

建议先完成小模块的积木块拼接，再嵌入大模块的积木块程序中。

这里我们先完成“回答=2”积木块的拼接，再嵌入如果……之中。不建议先将等于积木块嵌入如果……之中，再修改成“回答=2”。

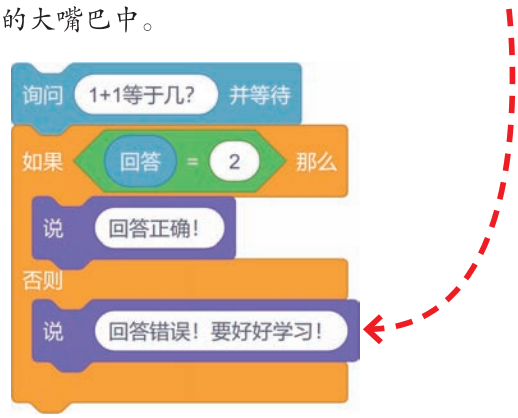
建议





7 从**外观模块**中找到**说“你好！”**积木块，将“你好！”改成“回答正确！”，然后拖动到**“那么”**下面的大嘴巴中。

8 重复第7步，将“你好！”改成“回答错误！要好好学习！”，然后拖动到“**否则**”下面的大嘴巴中。



9 单击脚本区的整个积木块，小猫咪立刻出了一个考题“1+1等于几？”，等待你的回答。

10 在舞台区底部的输入框中输入答案。

如果你回复的答案不是2，小猫咪会告诉你“回答错误！要好好学习！”。

如果你回复的答案是2，小猫咪会告诉你“回答正确！”。

这就是Scratch编程，是不是很神奇，很有趣。

1.5 如何自学 Scratch 图形化编程

相信你一定会爱上编程的，因为在这里没有枯燥的学习。本书将整个学习过程分解成一个一个的游戏关卡，你只需要奋勇向前闯关通关。

从一个一个的单积木块到复杂的积木块组合，从基础模块到特色模块，都需要不断尝试。单击运行积木块并观察舞台和角色的变化，了解每一个积木块在程序中所起到的作用。

1 需要知道每一个积木块所在的位置，但不是一个一个地死记硬背。在编程中一定不能生搬硬套，要理解性记忆，通过尝试、观察、修改、比较来掌握每一个积木块的作用。

2 理解每一个模块所对应的含义，然后有针对性地寻找积木块，在这个过程中会很自然地记下所有积木块的位置和功能。

3 暂时略过那些看不懂变化也看不懂书中解释的积木块，不需要着急，因为有些知识你还没有学习到。例如运算模块中的平方根，可能你还没有学习到相关数学知识，此时需要跳过这块的学习，到后面结合课堂学习再回过头来体验。

4 挑出你能明白作用的积木块，尝试修改其中的一些参数，再观察会有什么变化，并且去总结它的作用。

5 掌握了单独积木块的作用后，尝试将自己熟悉的积木块组合到一起，再观察有什么神奇的变化。

6 最后有针对性地选择积木块，并将它们组合到一起，实现自己的想法。

右转15度　积木块的学习方法

1 阅读——积木块上的文字“右转15度”。
2 猜测——这是让小猫咪右转。
3 尝试——单击积木块，看看小猫咪有什么变化。
4 观察——小猫咪旋转了。
5 修改——修改积木块中的数字15，继续观察。
6 总结——这是一个控制小猫咪右转的积木块，数字越大小猫咪旋转幅度越大。
7 结合——结合数学知识，理解角度的概念和应用。

组合积木块的学习方法

将说“你好！”积木块、重复执行10次积木块、右转15度积木块组合在一起。

观察小猫咪从单个积木块到组合积木块的变化。如果你已经可以自由组合积木块，并且能达到想要的效果，那么恭喜你已经入门了，基本掌握了简单的积木块。下面需要仔细地跟随本书内容制作一个一个小游戏，培养逻辑思维，提升思考质量。注意要多思考、多观察、多尝试。

1.6 如何亲子共学

如果你是一位家长，想教孩子学习编程。那么首先要把自己当成孩子，看完全书，掌握每一个积木块，做一个博学的家长，然后摇身一变成为一个玩伴。

如果你的孩子还没有上小学，那么建议让孩子通过“编程一小时”网站学习Scratch或者学习 ScratchJr（ScratchJr更适合幼儿）。如果你的孩子已经在读小学了，那么按照本书的内容顺序学习即可，本书内容都是以孩子的视角编写的。在教学过程中需要特别注意，在你看来十分简单的程序和过程，对于孩子来说或许并不简单，因此需要更多的耐心。此时此刻你不是老师，也不是家长，而是孩子的一个玩伴，陪伴孩子一起探索程序的奥秘，感受程序的神奇，体验游戏的乐趣。在教学过程中，不能灌输理念，应该更多地进行演示，并让孩子自己动手修改程序，观察通过修改程序所带来的变化。

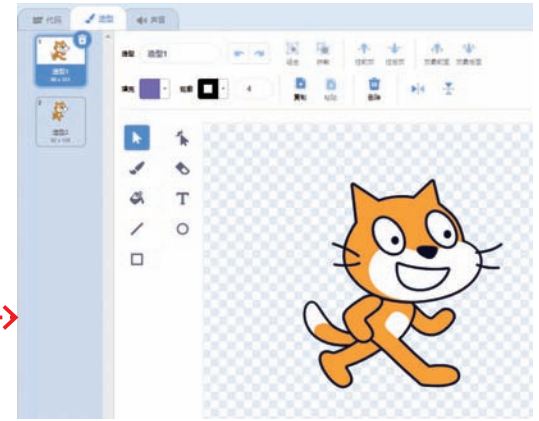
造型切换教学

首先，不断单击**外观模块**中的**下一个造型**积木块，将看到小猫咪的样子不断在两个图片中来回切换，就像奔跑一样。

让孩子自己动手尝试这个积木块，感受积木块对角色的作用。

然后，提出问题：为什么会有两个造型？为什么只有这两个呢？

最后，带孩子来到**造型**界面，观察小猫咪拥有的造型，可以看到小猫咪有两个造型。



在**造型**界面，让孩子分别单击两个小猫咪造型，观察舞台中小猫咪的变化。

多次对比不同单击方式带来的效果，让孩子明白，原来**造型切换**就是这样的。

1.7 教师备课建议

如果你是一名少儿编程老师，相信在程序知识方面就不需要我多说什么了。如果对程序还不是特别了解，那么可要下苦功夫了，自己拥有丰富的程序知识，才能更好地传递给孩子。但是只有丰富的程序知识是远远不够的，如何让孩子接受知识更为重要。像给成人上课一样去解释坐标、变量、比较、判断等，孩子一定是两眼发呆地看着你而不知所云。

对于编程的教学，我的建议是将自己的知识储备尽可能地放空，达到和孩子一样的知识量，然后在课堂上运用思维方式，和孩子一起去探索学习新的知识。因为真正要学习的是思维方式，而不仅仅是编程知识。

有时候孩子的学习是“知其然而不知其所以然”的，让他们理解这个事情为什么会这样，不能只是纯粹的概念解释和灌输，而应该让孩子保持对世界的好奇心、想象力以及创造力。因此，这需要我们站在孩子的视角去思考问题、看待问题，减少书面式的告知，让孩子尽量动手尝试，去改变、去创造。

以项目为向导，完成一个一个的项目，带给孩子学习的乐趣、活跃的课堂气氛，吸引孩子的注意力，让孩子融入其中的同时达到锻炼思维逻辑的效果。

在课程中加入更多角色，如奥特曼、铠甲勇士等动画片里的角色，相信孩子们会兴趣大增。

小朋友们，你们玩过植物大战僵尸吗？里面哪个僵尸最厉害呀？这样两个问题下来，小朋友们一定可以欢快地进入联想。现在我们就用程序控制僵尸走路，首先添加小朋友们认为最厉害的僵尸角色（老师需要在计算机文件夹里准备三四个僵尸角色，不能过多，否则孩子会选择困难）。

拖动**移动10步**积木块控制僵尸行走，小朋友们动动脑筋，怎么控制僵尸走快点或走慢点呢？老师想让僵尸转晕，该怎么办呢？发动你们的小脑筋去寻找合适的积木块吧！

当孩子知道旋转后，我们就该再次提问了，让孩子去思考。

1. 老师想让僵尸向左旋转，该怎么办呢？
2. 让僵尸向右旋转90度，该怎么办呢？
3. 让僵尸不停顿地旋转，该怎么办呢？

4 让僵尸一边走一边旋转，该怎么办呢？

不要低估孩子的想象力和学习能力，你会发现孩子其实都能完成，而且似乎不需要你的提示。就像你给孩子一个iPad，他就能很快地掌握如何使用一样。孩子都有一颗探索和尝试的心，大人反而会担心这样按键是不是会坏，所以不要阻碍孩子去探索和尝试。教学要注重孩子思维逻辑的培养、解决问题能力的提升、思考方式的养成，而不仅仅是为了完成作品以给家长一个交代。

STEM（Science、Technology、Engineering、Mathematics，科学、技术、工程、数学）教育其实是基于标准化考试的传统教育理念的转型，更加注重学习的过程，而不是结果。从本质上来说，要敢于让孩子们犯错，让他们尝试不同的想法，让他们听到不同的观点。与标准化试题相反，我们希望孩子们有与众不同的想法，并且创造能够应用于真实生活的知识。

教学需要注意以下5点：

1 联系（Connect），注重学习与现实世界的联系。

2 建构（Construct），“玩中学”“做中学”“创中学”，逐渐形成建模思想。

3 反思（Contemplate），反思编程过程，更深刻地理解和思考。

4 延续（Continue），应用到生活中，去解决一些实际问题，不断挑战和积累。

5 过程（Course），注重学习的过程，而不是体现在试卷上的知识结果和作品展示。

本书提供配套的教学视频，以及全套的案例源代码，希望可以帮助每一位学习者更好地学习。

Scratch

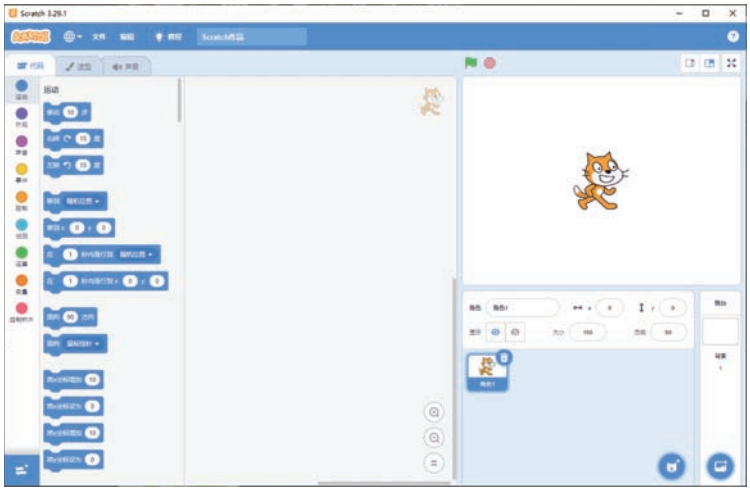
第2章

踏入编程世界

1. 界面初体验

双击打开Scratch 3.0软件，清爽的编程界面映入眼帘，这里就是我们尽情编织梦想的地方。

（本章体系化地讲解软件界面，你也可以选择快速翻阅并进入正式的课程学习，把本章内容作为界面的工具手册。本章的知识在后续的课程中都有融合。）



编程界面划分为**功能区**、**积木区**、**脚本区**、**舞台区**、**角色区**和**背景区**，它们就像人体的器官一样有着自己的特性和作用，为编程的正常运转贡献着自己的一份力量。

现在就跟随我一起来探寻它们。只有当你实实在在地看到了它们，才能有更直观的感受和更深刻的记忆。

接下来，我们将迎来新的任务挑战和学习计划，那就是了解每一个区域的特性和作用。

2. 功能区

下面让我们逐个看一看，功能区中的每个按钮都蕴含了什么功能。

1）语言小地球

它被我们称之为“**语言小地球**”，单击右边的**白色倒三角**可以打开软件的语言列表，通常“简体中文”在末尾。

2）文件操作

对程序文件进行操作，需要单击“文件”。它里面有三大功能：

①创建新的程序作品。

②从电脑中打开已经编写好的程序文件。

③将创作好的程序文件保存到电脑中。

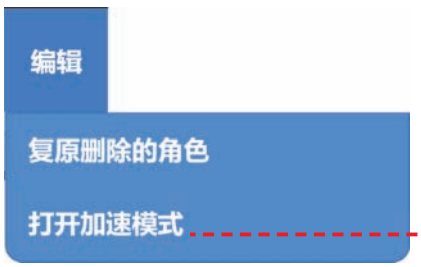
提示

保存在计算机中的.sb3文件，通常直接双击是打不开的，需要在软件中单击“文件”→“从电脑中打开”，然后选择打开对应文件。

3）编辑操作

- **恢复**

当我们不小心错删了角色时，“编辑”中的“恢复”会变成“复原删除的角色”。单击“复原删除的角色”可以帮助我们快速找回删除的角色以及角色中的代码。



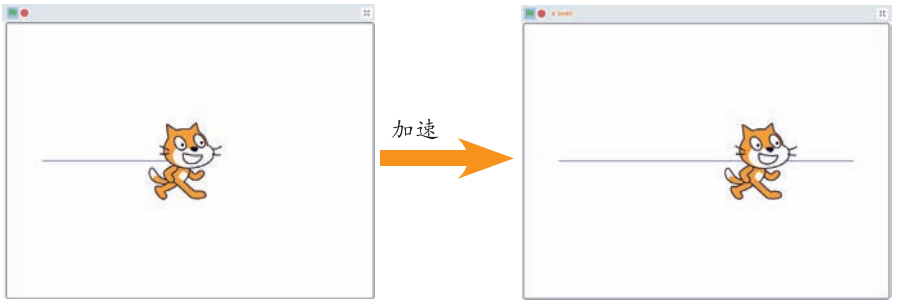
- **打开加速模式**

打开加速模式可以让程序运行速率加快。

不妨体验一下**案例：体验加速模式**，观察普通模式（关闭加速模式）和加速模式这两种状态下程序的运行效果。

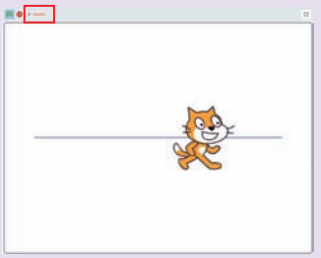
案例：体验加速模式

普通模式　　加速模式



洞查时刻

当加速模式开启后，小猫咪移动速度超快，线条也是瞬间绘制完成。同时你会观察到舞台区顶部有一道闪电，表示加速模式已开启。



4）教程

软件也为我们准备了形式多样的趣味教程，有动画、艺术、音乐、游戏、故事等。

5）作品名字

这里是给创作作品取名字的地方，一个优秀的作品应该配上一个不错的名字。作品的

名字可以是对作品的描述，可以是梦幻的构想，可以是记忆的亮点，更可以是你心中所想。你的作品，名字你做主。

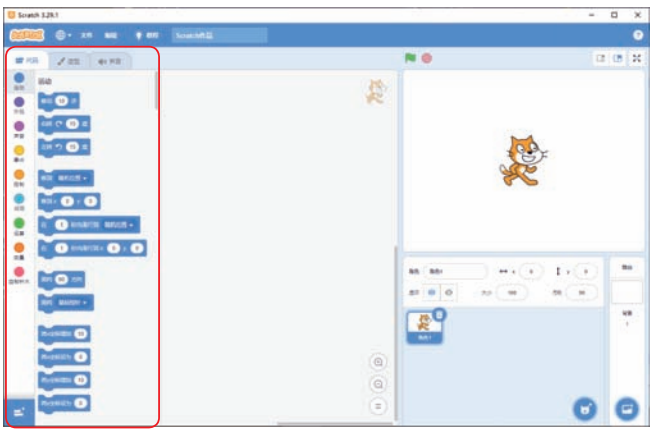
3. 积木区

积木区拥有三大标签，分别是**代码**标签、**造型**标签、**声音**标签。

1）代码标签

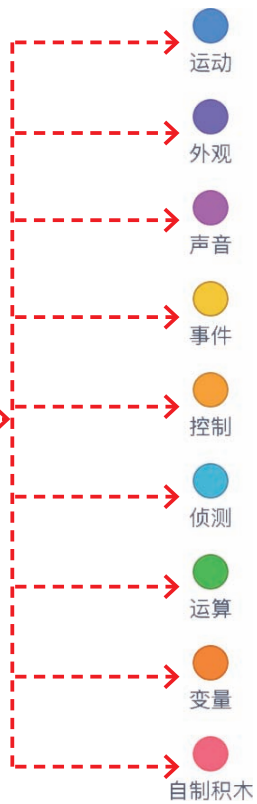
选择**代码**标签可以看到**积木模块**和**积木**，这是我们的积木重地。

它掌管着编程所需的所有积木块，通过积木块的各种组合创作出千千万万的程序，实现我们一个又一个的想法。



在**代码标签**下，积木块被分成了9个模块，并被赋予不同的颜色，方便我们查找。

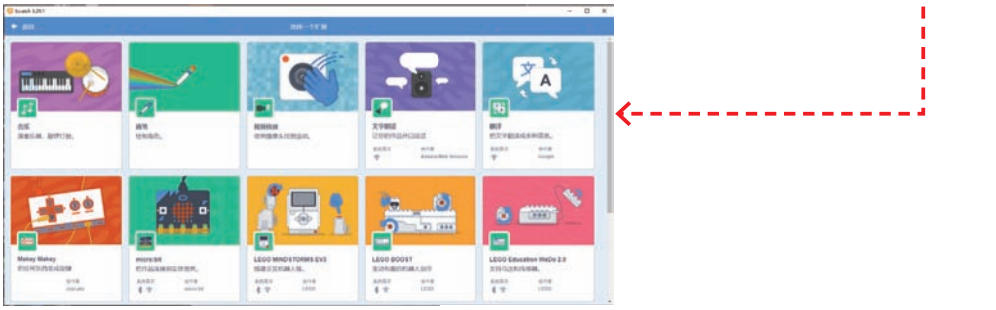
每一种颜色的小圆圈代表着一类积木块。每一类积木块都披上了和小圆圈相同颜色的外衣。这些模块分别是**运动模块**、**外观模块**、**声音模块**、**事件模块**、**控制模块**、**侦测模块**、**运算模块**、**变量模块**、**自制积木模块**。



每一个模块都有很多积木块，它们存放在积木区。来到积木区滚动鼠标，可以看到各式各样颜色各异的积木块。

9个模块里的**自制积木模块**与众不同，因为它里面空空如也，这里的积木块需要由你来创造。

除此之外，还有很多特色功能。例如**音乐**、**画笔**、**视频侦测**、**文字朗读**、**翻译**、**Makey Makey**、**micro:bit**、**LEGO MIND S TORMS EV3**、**LEGO BOOST**、**LEGO Education WeDo 2.0**、**Go Direct Force & Acceleration**，它们通通被放进了扩展功能区。



有了这些功能，我们可以让程序朗读课文、翻译外语、设计体感游戏，甚至可以控制乐高机器人，既有趣又酷炫。后面的课程我将和你一起探索。

试一试

单击积木区的积木块，观察舞台区中的小猫咪角色，选择正确的角色行为选项。

（1）单击**移动10步**积木块，小猫咪动了吗？

A. 动了　　B. 没动

（2）单击**右转15度**积木块，小猫咪会怎么运动？

A. 小猫咪没动　　B. 小猫咪会向左转　　C. 小猫咪会向右转

（3）单击**说“你好！”2秒**积木块，角色有什么变化呢？

A. 出现文字气泡“你好！”

B. 动了一下

C. 出现文字气泡“你好！”，一会后气泡消失

通过试一试，你是不是已经明白积木块的作用呢？如果你还困惑上面试一试问题的答案，可以通过下面两步来解决。

1. 每个积木块上面都有文字功能说明，结合前面讲的方法进行探索式学习。
2. 验证答案，单击问题中的积木块，观察小猫咪的变化。

例如：第（3）题。

当积木块上围绕一圈黄色的光芒的时候，说明积木块程序正在运行，通过观察小猫咪的变化选出答案。

积木块被单击后，小猫咪身边出现文字气泡**“你好！”**

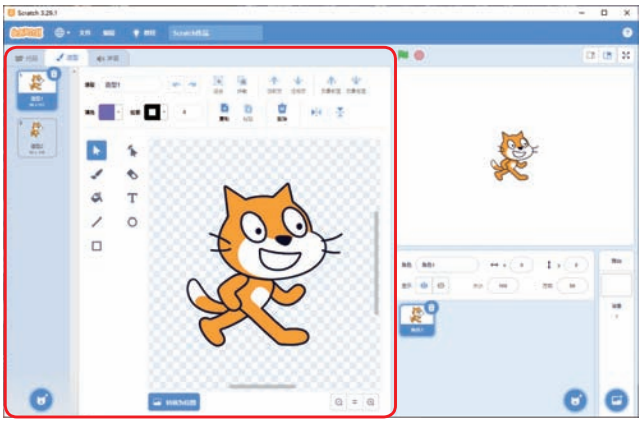
过了2秒，文字气泡“你好！”消失。

因此，答案选择C，因为程序运行后的效果是出现文字气泡“你好！”，过了一会气泡消失了。

2）造型标签

选择**造型**标签可以看到角色的造型列表和画板。在这里你可以化身神笔马良，自由绘制角色造型。

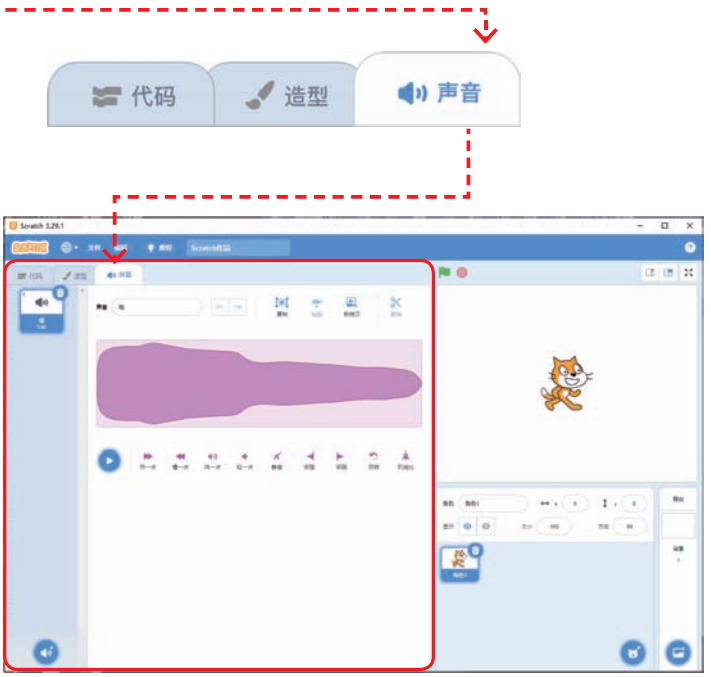
想要创作属于自己的角色造型吗，从这里落笔吧！



3）声音标签

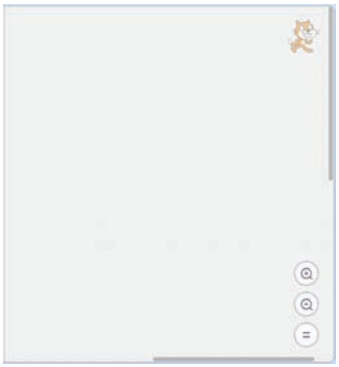
选择**声音**标签可以看到角色的声音列表，在这里可以给角色增添音效和录制对话。

想不想展示一下歌喉，想不想录制一段台词。单击**声音**标签录制一段音频，再剪辑一番，一段美妙的音乐就诞生了。



4. 脚本区

积木区就像存放积木的盒子，脚本区如同搭积木的玩耍区。在脚本区，你可以尽情地构建和拼搭——创作游戏、制作工具、设计动画、模拟现实、跨学科学习……这里是展示无尽想象力和无边创造力的地方。



之前，我们在积木区单击积木块，它们只能一个一个地运行。但是在脚本区，它们可以拼接组合在一起，每一种组合都是一种可能，有了无限的组合，就有了无限的可能。

试一试

1 拖出积木块

①找到并点击**事件模块**。

事件

②找到**当 被点击**积木块。

③移动鼠标指针到积木块上，按下鼠标左键。

④一直按住鼠标左键，将积木块拖动到脚本区，再松开鼠标左键，积木块就留在了脚本区。

2 拼接积木块

①单击运动模块。

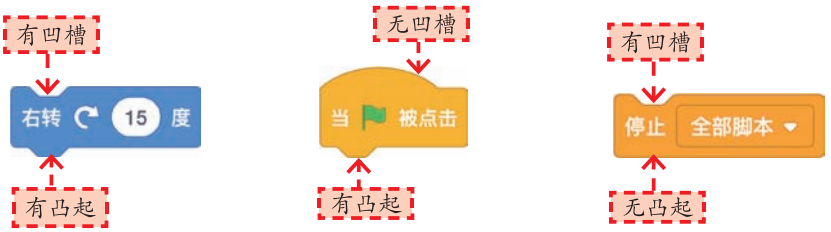
②找到右转15度积木块，将它拖动到脚本区。

③拼接到当被点击积木块下方。

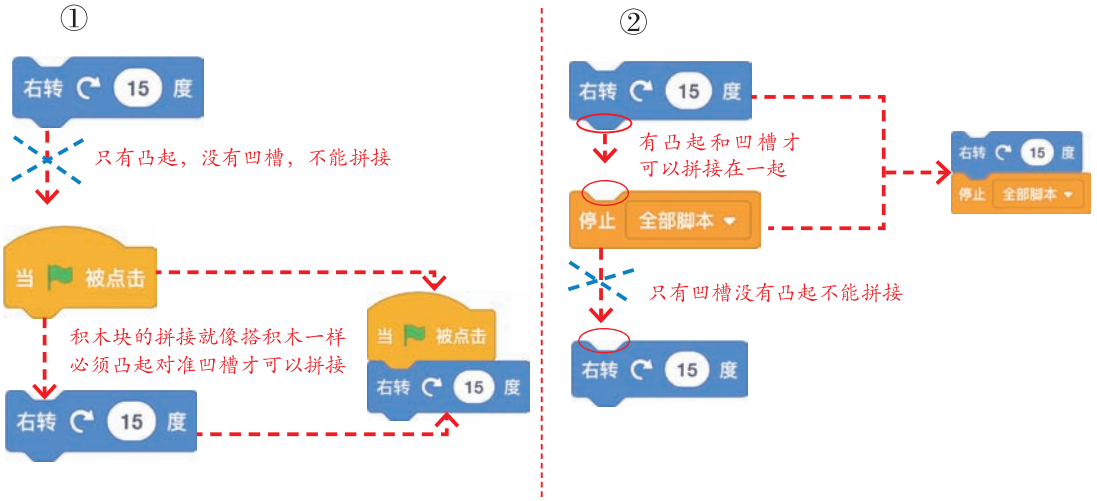
④单击积木块运行程序。

1）积木块的拼接

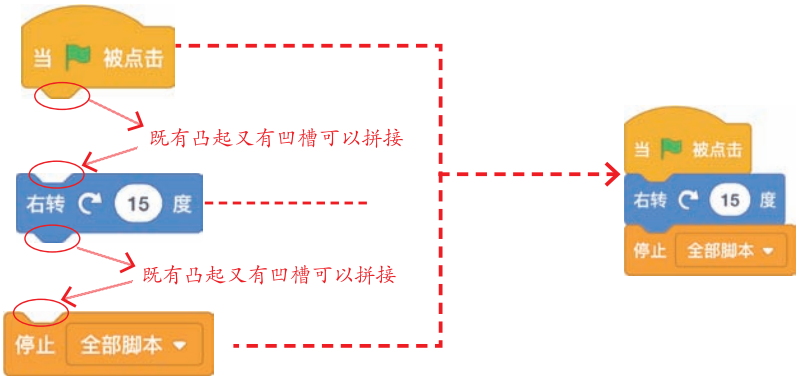
大多数积木块都有凹槽和凸起，但也有一些例外，下面来看看不同情况的积木块吧。



积木块拼接需要凹槽对准凸起才可以，否则不会拼接成功，下面来看看积木块的拼接情况。



③



就算积木块有凸起和凹槽，它们也不会自动拼接哟！需要你拖动它们将它们组合起来。

将凸起靠近凹槽或者将凹槽靠近凸起，被拖动的积木块会出现灰色背景，这个时候松开鼠标就可以实现拼接了。

①从下往上拼接

②从上往下拼接

2）积木块的删除

积木块拼搭难免会出错，需要删除多余或错误的积木块。这个时候只需要两步即可轻松完成。

1. 将要删除的积木块脱离积木组合。
2. 将要删除的积木块拖动到积木区，松开鼠标左键完成删除。

拖动积木块的时候，会将它下面的所有积木块一起拖动哟！

例如，直接拖动**右转15度**积木块，会将**停止全部脚本**积木块一起拖动。

3）积木块的包裹拼接

案例：被转晕的猫

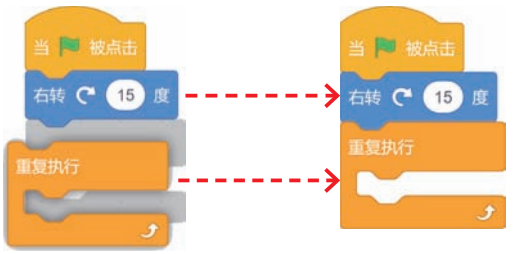
1 单击**控制**模块。

2 将**重复执行**积木块拖动出来。

3 移动到要包裹的积木块上面，“咔哒”一声积木块包裹就拼接完成了。

4 这样**右转15度**就会重复执行了。

如果像下面这样拼接，就不算包裹拼接了，而是凸起拼凹槽了。



4）积木块的嵌入

每次只旋转15度太固定了，如果可以随机旋转角度或许会更加有趣。

1 单击运算模块。

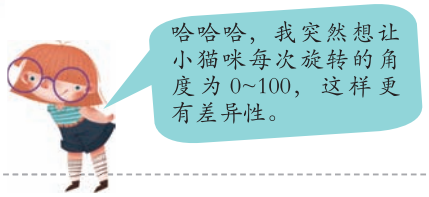
2 将在1和10之间取随机数积木块拖动出来。

3 拖到右转15度积木块中的圆圈里。

让积木块的最左端靠近小圈圈，这样更容易放进去。

这样积木块就嵌入进去了。

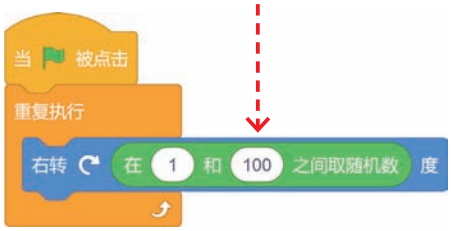
5）积木块的修改



案例：快慢旋转猫

1 单击随机数积木块小圆圈里的数字10，出现淡蓝色背景。

2 用键盘输入100。



5. 舞台区

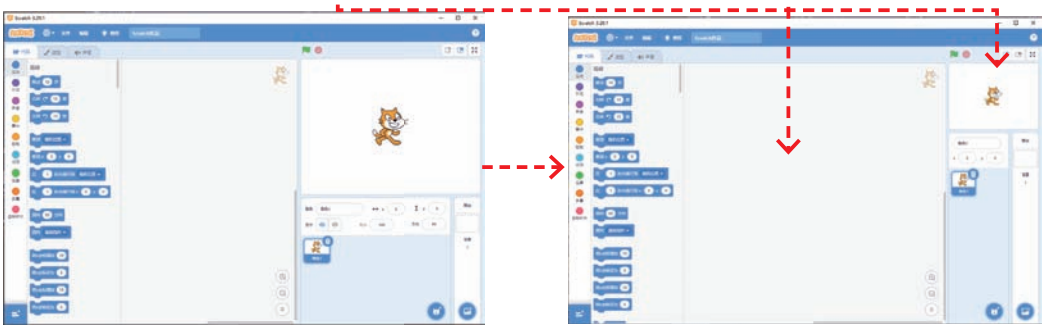
舞台区是展示程序效果的地方，如果你创作的是一个游戏，那么它就是游戏界面；如果你制作的是一个工具，那么这就是操作界面；如果你拍摄的是一部动画，那么这就是观影屏幕；如果你录制了一段音频，那么这就是音乐播放器……

下面介绍舞台区中各个图标按钮的功能。

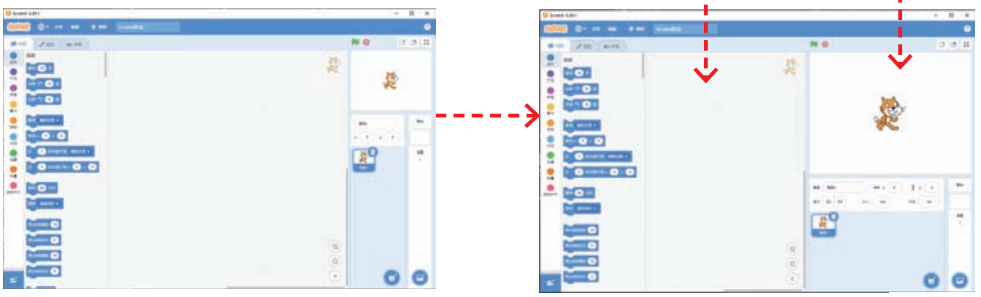
（1） 小绿旗，单击它启动程序。

（2） 小红圈，单击它停止程序。

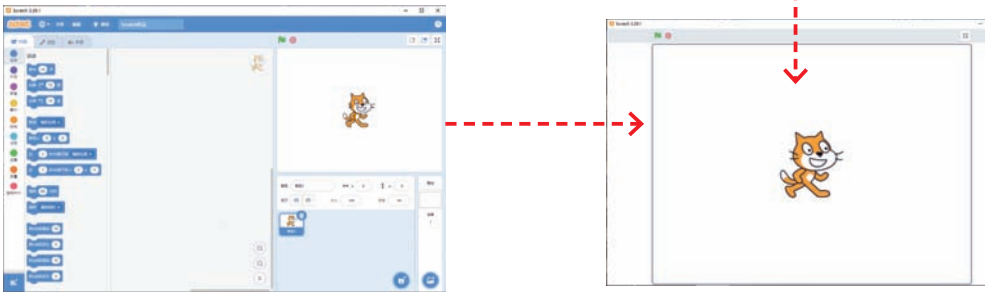
（3） 将舞台区缩小，放大脚本区，给你更多编程空间。



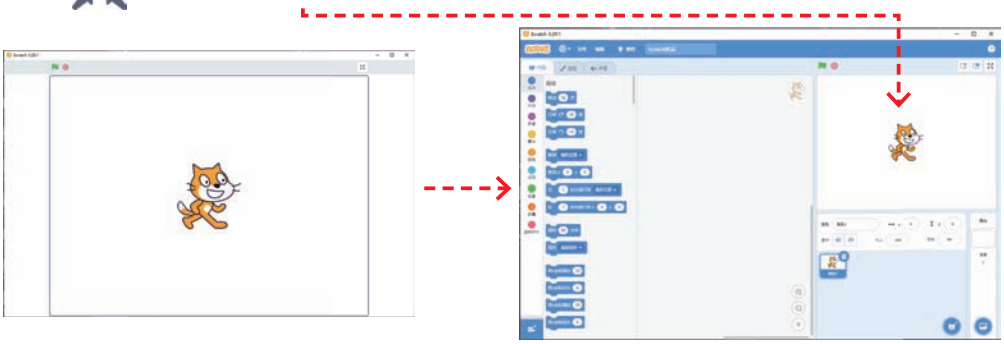
（4） 将舞台区放大，缩小脚本区，让编程效果更清晰。



（5） 将舞台区全屏化，沉浸式体验程序。



（6）
将全屏恢复到软件界面，所编即所得地实时查看。



5. 角色区

舞台区中的所有角色都会有一个缩略图存放在角色列表区。角色列表中记录着每个角色的相关信息，通过它可以设置角色的坐标，调节角色的大小，调整角色的方向以及控制角色的显示和隐藏等。

1）添加角色

将鼠标指针移动到按钮上方，在弹框中选择添加角色的方式，也可以直接单击该按钮进入系统角色库。

（1） 从电脑文件中上传角色。

（2） 随机选择系统库中的角色。

（3） 绘制全新的角色。

（4） 进入系统角色库中查找角色。

2）删除角色

选中角色，单击右上角的垃圾桶按钮即可删除角色。

7. 背景区

舞台区可不是只有角色呢，它还可以展示各种背景图片，这样舞台才能更加绚丽。掌管着舞台背景的区域就是背景区，想要跟换舞台背景，就需要来到这里。

▶ 思考提问

想要知道怎么添加背景吗？结合添加角色的讲解，自学一下吧。

写下背景添加的讲解说明：

_____

_____

_____