

无限循环，它会永远重复执行。本章将详细学习 Scratch 中的重复结构、停止积木（用来结束无限循环），以及用户输入的验证等。

本章还会讨论嵌套循环（循环中包含循环）、递归（过程自己调用自己，这是另一种重复执行的方式）以及相关案例。最后将探索多个使用循环和条件语句的有趣的程序，将循环与实际程序相结合。

## 循环结构

早在第 2 章我们就知道了循环积木能重复执行脚本。Scratch 支持三种类型的循环，如图 7-1 所示。



图 7-1 : Scratch 的重复结构

我们之前多次使用过**重复执行…次**和**重复执行**，下面将介绍**重复执行直到**积木以及相关的术语。

循环积木每重复执行一次，我们则称其进行了一轮或一次迭代。迭代的次数使用术语循环计数表示。你熟悉的**重复执行…次**积木就是确定型循环，因为它的迭代次数是确定的。当我们可以确定迭代次数时，这种积木便是首选。例如，之前绘制多边形时，我们指定的边数即为迭代次数。

而**重复执行直到**积木为不确定型循环，因为它是根据条件测试的结果决定是否重复执行其内部的脚本。当事先不知道循环次数且直到某些条件成立之前希望一直循环时，我们通常使用该积木。使用它可以很自然地实现某些功能，例如，“重复执行**询问…并等待**，直到用户输入了正数”或者是“重复发射导弹，直到玩家的精力值低于某个值”。下面就随我一起探索不确定型循环吧！

### 重复执行直到积木

假设某游戏向玩家提出一道数学题，如果回答错误，游戏重新给玩家一次回答的机会。换言之，游戏会询问相同的问题，直到玩家回答正确。显然，此处使用**重复执行…次**积木不是非常合适，因为我们事先并不知道玩家要输入多少次，也许第一次就回答正确了，或者运气不好尝试了 100 次还是错的。这种情形使用**重复执行直到**