

Pencil Code

Aerber Zhou

Published
with GitBook



Table of Contents

Introduction	0
First Chapter	1

Pencil Code

第九章

这是Pencil Code的第九章。

介绍的是嵌套循环。

9.0.1 教学目标

这一章有两个目的,第一个目的是介绍嵌套循环的概念和展示给学生如何构建他们.这节课使用ASCII艺术来教学嵌套循环.第二个目的是介绍二维输出网格系统的概念.取遍历一个二维网格最常规的方法是使用嵌套循环.这使得嵌套循环对于学生来说是一个需要学习和掌握的有力的结构思想.

9.0.2 大纲

9.0 本章介绍

9.0.1	教学目标
9.0.2	大纲
9.0.3	关键术语
9.0.4	关键概念

9.1 教案

9.1.1	教学建议
9.1.2	推荐日程
9.1.3	CSTA标准
9.1.4	教学计划I:使用易理解的嵌套循环
9.1.5	教学计划II:遍历二维网格来构建ASCII艺术

9.0.3 关键术语

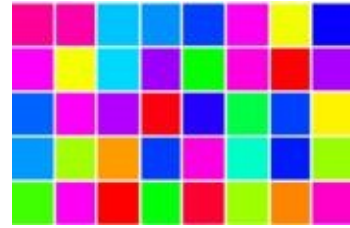
索引	列表
矩阵	ASCII 艺术
嵌套	追踪
动画	内外循环

9.0.4 关键概念

当循环嵌套在循环中，一个程序可以创建一个重复的重复。下面是一个例子.

```
for (var j = 0; j < 5; ++j) {
  for (var k = 0; k < 8; ++k) {
    typebox(random(color));
  }
  typeline();
}
```

输出:



内循环和外循环

当嵌套循环，循环扮演非常不同的角色：

- 外循环从头到尾的运行。它完成一次又运行一次。在上面的实例中，外循环使用的变量使其重复五次去运行内循环，然后在每次在每一行结尾的输出后,做一个typeline,做整整5次。
- 内循环从头到尾的运行。它可以反复地重复完成。在这里,内循环循环5次，每次K从0开始计数到8，做8次typebox。当程序完成时，内循环将重复 $5 \times 8 = 40$ 次。

内循环快速运行，每一次外循环,都会完成一个完整的内循环。外循环循环的偏慢。

内部和外部的循环类似时钟指针的运转。例如，一个时钟的秒针就像内部循环同时分针就像外循环.对于每一次分针的转动,秒针都需要转一圈,转动60整整秒.当分针转动一整个小时时,秒针将转动 $60 \times 60 = 3600$ 秒。

里程表和日历的都是用这样相似的工作方式。

非独立的内循环

有时，内部循环的工作方式取决于外部循环的哪一步运行。例如，创建一个程序打印在一年中所有的日期，您可以使用嵌套循环去打印月份的每天，但内部循环的次数取决于外部循环，这是因为每个月有不同的天数。

```
lengths = [31, 27, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];
for (var month = 0; month < 12; month++) {
  for (var day = 0; day < lengths[month]; day++) {
    write((month + 1) + "/" + (day + 1));
  }
}
```

//一个打印一年中每一天日期的嵌套循环

在这里，内部循环重复的是不同的次数，这取决于它在哪个月份上，因为它的结束条件是 `day < lengths[month]`。

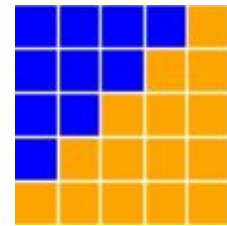
对于月份，我们绘制了一个依赖于存放每一月的天数的数组的内循环,但是对于给一个非独立内循环构思一个规则需要更加的泛化:这样一个程序员需要考虑使用的特殊情况,同时要尝试泛化这个规则。

泛化一个内部循环的规则

假设我们希望创建一个程序可以创建类似于下面的图案。

```
N = 5
for (var row = 0; row < N; row++) {
  for (var b = 0; b < ??; b++) {
    typebox(blue);
  }
  for (var j = 0; j < ??; j++) {
    typebox(orange);
  }
  typeline();
}
```

预期输出：



规划一个非独立内循环时:我们如何选择红色的循环边界？

我们想要的输出是一个5x5分布的带颜色的方格，但是我们想泛化我们的程序以至于这个程序可以形成任意大小、相似的图案。每一行都有一定数量的蓝色方格，后面随着一定数量的橘黄色的方格，我们可以使用一个类似右边的嵌套循环程序，用一个外部循环来规定一行，用两个内部循环，一个内部训练对应蓝色方格，一个内部循环对应橘黄色方格。

需要放入的值是由问题决定的，这使得程序需要泛型化。我们可以通过一些特殊输出查找图案规律来达到这一点。下面是一个总结了每一行的图形的表格。

行数	蓝色方格数目	橘色方格数目	蓝色方格的公式	橘色方格的公式
row = 0	4	1	$4 = N - 1 - \text{row}$	$1 = \text{row} + 1$
row = 1	3	2	$3 = N - 1 - \text{row}$	$2 = \text{row} + 1$
row = 2	2	3	$2 = N - 1 - \text{row}$	$3 = \text{row} + 1$
row = 3	1	4	$1 = N - 1 - \text{row}$	$4 = \text{row} + 1$
row = 4	0	5	$0 = N - 1 - \text{row}$	$5 = \text{row} + 1$

你可以在表格的前三列填写通过寻找的例子形状和计数。但为了概括，你需要找到在最后两列中存在的规则。这需要找到一条一行中的方格的数目和行数之间的关系。例如，橘色方格的数量在每行总是比行数多一个。因此，橘色方格数目的公式都是 行数+ 1。

对于蓝色方格，行数每增加一个，方格就减少一个，所以规则应该使用减法。在这个例子中，适用的公式是 $4 - \text{行数}$ 。但是这个并不是很泛化。想象一个更大的 10×10 的图案， $N = 10$ 。在这种情况下，在第0行的蓝色的数目将不会是4，它将会是9！因此，一个完全的广义公式 $N - 1 - \text{row}$ 适用于任何一个大小 N 的任意一行。

填充在公式中的两个内部循环的条件 $b < \text{row} + 1$ 和 $j < N - 1 - \text{row}$ 将使程序适用于任何一个 N 。

由于内部循环和外部循环的关系，嵌套循环可以使具有挑战性的程序正确。设计嵌套循环的一个很好的策略是在一个例子中仔细地找出你想要的运行效果，然后找到一个方法来泛型它。

9.1.1 教学建议

下面的课程计划是结构化的以便于让学生掌握嵌套循环的思想。鼓励学生在遍历的过程中追踪值，然后引入二维网格的思想。确保Pencil Code一直处于JavaScript代码模式。尽可能多地鼓励学生多处于在文本模式然后将他们的代码成型。他们可以切换到模块模式（在需要时），然后把模块放到程序中。

9.1.2 推荐日程 (155分钟的课堂时间)

教学日	建议
第一天	教学计划I
第二天	教学计划II


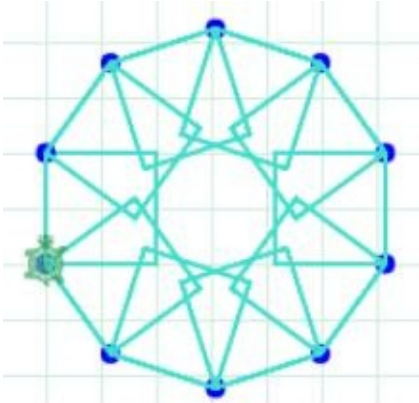
9.1.3 CSTA标准

CSTA标准	CSTA部分	CSTA覆盖的学习目标
等级3 B (等级9-12)	协作 (CL)	使用项目协作工具，版本控制系统，集成开发环境 (IDE)，协同软件项目工作。
等级3 A (等级9-12)	计算思维 (CT)	解释如何序列，选择，迭代和递归去构建块的算法。

9.1.5 教学计划I

本课介绍嵌套循环。这有两个程序。这有二种不同的教学工具在这展示。第一个是显示程序的逻辑运行路径，追踪代码。播放视频演示给学生循环是如何执行发生。第二个是对代码修改提出简单的问题，阐述每一行代码的作用。这很可能无论在进行什么其他概念教学时都需

要使用的两种思路。

具体内容	教学建议	时间
<p>代码:</p> <pre> speed(100); for (var i = 0; i < 10; ++i) { for (var j = 0; j < 19; ++j) { typebox(purple); } typeline(); } </pre> <p>输出:</p> 	<p>使用在最左边一列中提供的代码。</p> <p>转换到Text-mode</p> <p>运行程序同时阐述嵌套循环是怎样运行的。</p> <p>运行程序同时展示外部循环的每一个迭代和内部循环是怎么完成每一个迭代。</p>	<p>阐述时间: 30分钟</p>
<p>代码:</p> <pre> speed(1); pen(purple, 1); for (var i = 0; i < 10; ++i){ dot(blue, 10); for (var j = 0; j < 4; ++j){ fd(50); rt(90); } lt(36); bk(50); } </pre> <p>输出:</p> 	<p>让学生优化装饰巢程序。</p> <p>鼓励学生处在Text-mode。</p> <p>当他们在显示器上看这个程序的时候，要求学生回答以下问题。</p> <p>(问题可以是被投影，或者写上白板，或者作为一个需要上交的填写作业)</p> <ol style="list-style-type: none"> 创造了什么内部循环？ 创造了多少蓝方格？ lt和bk指令是作为哪个循环的一部分？ 他们得到多少次执行？ 当内部迭代增加时发生了什么？ <p>给学生五分钟左右回答问题，然后，作为一个大的分组讨论学生分享他们的答案。(另外，这可以作为一个作业或明天的一个热身)</p>	<p>阐述时间: 30分钟</p>

给学生五分钟左右回答问题，然后，作为一个大的分组讨论学生分享他们的答案。（另外，这可以作为一个作业或明天的一个热身）

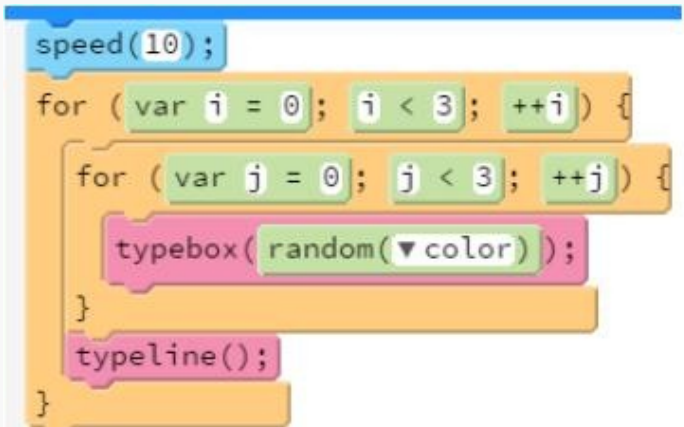

作为一个装饰巢程序的拓展，要求学生将这个程序修改'for'循环，例如使用额外的FD和RT指令来创建有趣的图案。

教学提示：奖励额外分数给在课上提交项目最有创意的学生。

学生练习：55分钟

9.1.5 教学计划II

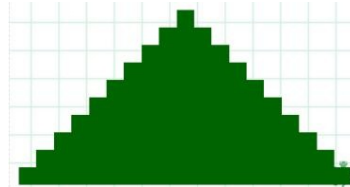
本课程着重于利用嵌套循环构建一个简单的二维输出图像。它是有趣和丰富多彩的创建ASCII文本艺术的伟大垫脚石。

具体内容	教学建议	时间
<p>代码:</p>  <pre> speed(10); for (var i = 0; i < 3; ++i) { for (var j = 0; j < 3; ++j) { typebox(random(▼color)); } typeline(); } </pre>	<p>给学生在最左边一列中的程序。让他们在text和block之间切换来理解程序的逻辑。</p> <p>要求他们修改程序使用这些可能的变量，如：</p> <ul style="list-style-type: none"> o 迭代的数字i和j o 颜色-不变化的颜色 <p>问他们如果迭代数字j和i不一样的话会发生什么？</p> <p>输出：</p> 	<p>阐述时间：20分钟</p>
<p>代码：</p> <pre> speed(500); for (var i = 0; i < 10; ++i) { for (var j = 0; j < 9-i; ++j) { typebox(transparent); } for (var j = 0; j < 1+i*2; ++j) { typebox(darkgreen); } typeline(); } </pre>	<p>让学生使用在最左边一列中的代码去设计一个三角形。</p> <p>然后，给学生展示如何使用typebox()去绘制一个typeline()单元，使他们可以创造ASCII/TEXT艺术设计。</p> <p>（如果需要的话，学生可以在Text和Block中切换思</p>	<p>阐述时间：</p>

```

speed(500);
for (var i = 0; i < 10; ++i) {
  for (var j = 0; j < 9-i; ++j)
  {
    typebox(transparent);
  }
  for (var j = 0; j < 1+i*2; ++j) {
    typebox(darkgreen);
  }
  typeline();
}

```



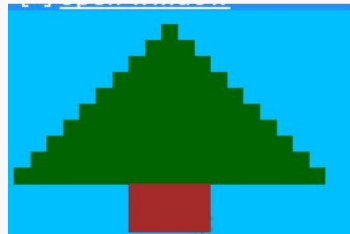
代码：

```

speed(500);
dot(deepskyblue, 5000);
moveto(6, 70);
for (var i = 0; i < 10; ++i) {
  for (var j = 0; j < 9-i; ++j){
    typebox(transparent);
  }
  for (var j = 0; j < 1+i*2; ++j) {
    typebox(darkgreen);
  }
  typeline();
}
jumpxy(4, 250);
for (var i = 0; i < 3; ++i) {
  for (var j = 0; j < 12; ++j){
    if (j < 7) {
      typebox(transparent);
    } else {
      typebox(brown);
    }
  }
  typeline();
}

```

使用这两种在课上被构建出的艺术形状，让学生用ASCII艺术构建一个类似在最左边一列中展示的有趣的图像。下面这个是一个程序例子输出：



阐述
时间：
55
分钟
学生
练习
时间：
100
分钟

9.2 资源

额外练习：

使用文字艺术的任务，让学生合作建立一个长期的项目。使用至少2个对象建立在文字艺术的最后的结果应该是一个有趣的场景。允许学生去使用其他在其他章节的库函数中的对象。要求每个学生至少应该独立创建一个文本艺术，同时要求他们提交自己的作品，因为这作为等

