

## 第八章：一维数组介绍

### 8.0.1 教学目标

一维数组（也称为向量）是一个程序存储许多线性结构数据元素的数据结构基础。在这个单元中，学生们应学会创建和遍历数组，以及如何在数组中增加、删除、插入和查找元素。通过使用 **Pencil Code**，学生们将探索如何使用来自网络的数据创建数组以及怎样使用数组中的数据创造可视化的效果。

### 8.0.2 主题大纲

#### 8.0 章节介绍

##### 8.0.1 教学目标

##### 8.0.2 主题大纲

##### 8.0.3 关键术语

##### 8.0.4 核心概念

#### 8.1 课堂计划

##### 8.1.1 建议时间安排

##### 8.1.2 CSTA 标准

##### 8.1.3 教学建议

##### 8.1.4 教学计划 I：在数组中使用“for...each”循环

##### 8.1.5 教学计划 II：遍历数组——文本模式

##### 8.1.6 教学计划 III：数组行为演示

##### 8.1.7 教学计划 IV：使用数组创建以一个数组中的元素为内容的条形图

##### 8.1.8 教学计划 V：使用数组创建以一个数组中的元素为内容的饼图

##### 8.1.9 教学计划 VI：使用数组查询数组中的数据元素

### 8.0.3 关键术语

线性数据结构	从 0 开始的索引
条形图	数组语法的规模
范围错误	
索引	
数组容量	

### 理解 CoffeeScript 中的数组

数据结构是一个电脑程序用来在某一时刻组织多条信息的对象。下面是一个带有 **for** 循环的数据结构。（要注意的是要创建这个程序需要在文本模式下完成。尝试着转到文本模式并编辑如下的 **for** 语句）

```
for x in [red, orange, yellow]
  dot x, 100
  fd 50
```

方括号中的对象（**[red,orange,yellow]**）是一个叫做数组的数据结构。**For** 循环遍历数组，在给 **x** 依次赋数组中的每一个值的过程中重复缩进的语句。

这段代码的微妙之处在于数组是他本身。数组可以它自己的变量表示，也可以用一个名称代替。例如，这个程序和上面的程序是等价的。

```
mycolors = [red, orange, yellow]
for x in mycolors
  dot x, 100
  fd 50
```

在这个代码中，用到的 **mycolors** 代表包含了三种颜色的数组。数组可以用于重复一个循环，

但是同一个数组也可以用其他方法使用。

基础使用：数组索引和数组长度

一个数组可以称为容器，因为他是一个包含了其他对象的对象。每一个数组都包含很多定义了明确顺序的元素，并且每一个数组都有一个长度（**Length**），它表示数组中的元素数量。元素之间互不相关，数组长度可以从中得到。（他们可以在文本模式中输入若干数组片段之后随时转换为块模式）（原文有错误）

Write mycolors.length	这里将输出”3”，因为有三个元素在数组中
Write mycolors[0]	这里将输出”red”，因为第一个元素是 red
Write mycolors[1]	这里将输出”orange”
Write mycolors[2]	这里将输出”yellow”

数组数字后的方括号的使用叫做 **indexing**，方括号中的数字是一个数组中特定元素的 **index**。**Indexing** 也可以用于改变数组元素。例如，下面的代码可以改变数组中的第一个颜色。

```
mycolors[0] = blue
```

改变数组中的一个元素对其它元素不产生影响。

零底数索引

**CoffeeScript** 和 **JavaScript** 以及其它大多数的现代编程语言中的数组都是以零作为底数的。以零作为底数，即数组的第一个元素是从 **0** 开始的（而不是 **1**）。这也意味着一个数组的最后一个元素的底数是数组的长度减 **1**。

自从人们逐渐习惯于从 **1** 开始计数，将 **0** 作为底数一开始让人们觉得违反常理。从 **0** 开始计数是语言设计的一个选择，像 **Fortran** 这样的更老的程序语言确实在用从 **1** 开始索引的形式。许多程序员在体验了从零开始索引和从 **1** 开始索引之后认为从 **1** 开始索引稍微清晰一些，因为这种方式允许程序员用某元素到数组开始处元素的距离来表示它的底数。因为第一个元素就在数组的开始处，它的到数组开始处的距离，它的底数，肯定就是 **0**。（另外，更多的关于从 **0** 开始索引的数学争议可以通过搜索 “**Edger Dijkstra’s discussion of zero-indexing**” 在网上找到。）

空数组和从 **0** 开始索引的循环语句

为了在 **CoffeeScript** 中实现使用从零开始索引方式的循环语句，我们将使用“**three-dot**”范围形式 **[0...N]**，就如下面程序的第二行所显示的：

```
mycolors = [red, orange, yellow]
for j in [0...mycolors.length]
write element #' + j + ' is ' + mycolors[j]
```

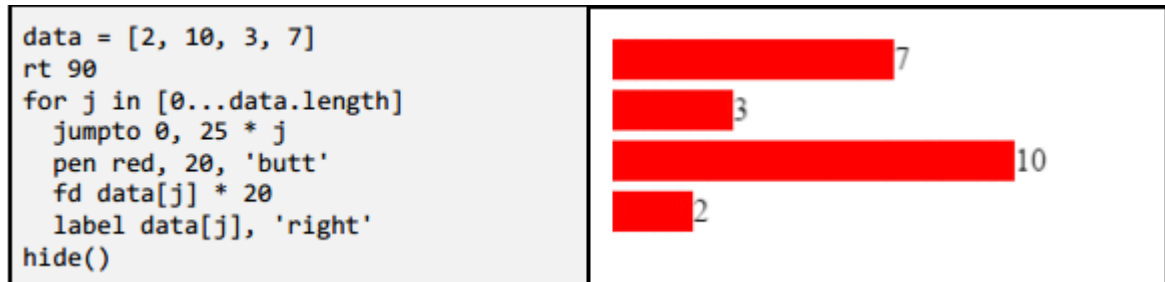
“**three-dot**”形式表示的数据范围被称为一个半闭范围，它从第一个数字开始计数但是明确忽略了最后一个数字，所以 **[0...3]** 表示 **[0,1,2]**，这也正是从零开始索引形式所需要的。“**three-dot**”形式范围还有相对于“**two-dot**”形式的另外一个优点：它在数组长度为 **0** 时也能正常工作！当计数到 **0** 时，并不是 **[1..0]** 这样倒置了两个数，而是用 **[0...0]** 这样从 **0** 开始到 **0** 结束，正因为它用了“**three-dot**”形式。这样做的结果是获得一个空序列，这意味着循环语句无法进入：这正是我们所需要的空数组。

一个空数组是一个极好且有用的数组！我们可以通过这样写创建一个空数组：

```
favoritecolors = []  
write 'the length is ' + favoritecolors.length  
空数组的长度当然是 0。
```

用数组做一个图表

数组可以包含任何类型的元素，如数字或字符串。”Visualization”是一个用来在数组中用数字创建一个图表的有用的编程类型。这是一个例子。



这个程序使用[0...data.length]来从 0 到 3 遍历数据元素底数，然后它使用 data[j]在某个时刻将一个数据元素从数组中读取出来。这些数字使用”turtle”功能来画一个图。这个程序使用了一些”pen”和”label”等另外的参数功能来做更精确的设计。”pen”在”butt”选项中，它是一个用来做一个直方形线条而不是一个圆形线条的绘图工具。”label”在”right”选项中，用来在特定位置的右方放置一个标签而不是在乌龟处。

从字符串和文件中创建数组

数组非常强大因为一个一维数组对象可以包含数千计或者数百万计的元素。然而，为了将其付诸实践，数组需要从程序外的数据文件中加载进来。

为了进行下次体验，我们在 Pencil code 中创建一个文件并更名为”mydata.txt”。一旦你给它的命名用”.txt”结尾，Pencil Code 将会知道这不是一个普通的程序，然后它会期待一个无格式的 txt 数据文件。在文件中保存一系列的数字不使用空格，仅仅是用逗号隔开，就像这样：

96,73,93,95,85,89,85,99,79,75,89,82,90,85,84,85,88,95,78,96,91,93

所有的数字都可以被使用；也许这是一串考试成绩。

这是一个将文件中的数据加载到数组并进行基础的统计学计算的示例程序。

```
await load 'mydata.txt', defer textdata  
mydata = textdata.split(',')  
total = 0  
for j in [0...mydata.length]  
  mydata[j] = Number(mydata[j])  
  total += mydata[j]  
write 'Total: ' + total  
write 'Average: ' + total / mydata.length
```

将文件加载到数组的三个步骤

正如上述程序中的例子所示，将一个文件加载到数组中需要两三个步骤：

1. 将一个文件加载为一个一维大字符串的文本数据：(await load 'mydata.txt', defer textdata)
2. 将文件划分成由多个小字符串组成的数组，每个元素一个数组：(mydata = textdata.split(','))

3. （如果数据是数字）把字符串转换成数字：(mydata[j] = Number(mydata[j]))

“load”功能能够从网上加载一个文件 URL 并调用一个“callback”功能获得一维字符串类型的文件目录。这里的程序加载一个短的文件名“mydata.txt”，文件名能够在程序正在运行的 Pencil Code 的文件夹里查找到。通过使用一个以“http://”开头的全名的 URL，Pencil Code 可以从网上加载数据文件。注意加载是输入的一部分（来自于网络而不是用户），它的运行就像是 I/O 模块中的读取功能。我们在这用“await”组合读取来在等待回复的过程中暂停程序。

“split”功能将一个 txt 字符串通过用定界符拆分的方式划分成一个由多个字符串组成的数组。定界符可以是任意的字母或符号。例如，用每行的“entry”单词拆分文件，用“\n”拆分文件（反斜杠 n 是一个出现在一个 txt 文件每行末尾的用于进入新行的符号）。

“Number”功能是将一个字符串转化为一个数字。为了避免“96”+“73”会获得“9673”这样的结果，被加载的字符串必须在进行运算之前转换成数字。

探索：将一个文档划分成数组，再将它连接起来

这是另一个在文件中查找字符的示例程序。它利用“split”功能通过一个特殊的符号将文件的所有单词都分割开而后使用“join”功能将数组组合回来成为一个大字符串并输出。

为了给这个程序准备数据，保存一个命名为“document.txt”的文件，里面包含大量的文本，例如一片来自 Wikipedia 的论文或一本公共领域的书。

```
await load 'document.txt', defer textdata
words = textdata.split(/\b/)
await read 'A word to search for?', defer q
for j in [0...words.length]
if words[j] is q
words[j] = '<mark>' + q + '</mark>'
write words.join('')
```

这个程序用数组做了四件事：

1. 它用分隔符(/\b/)创建数组。特殊符号/\b/将字符串分割成一个个的单词。
2. 它用 q 测试串 words[j] 中的每个单词，来尝试找到一个相符的单词。
3. 对于相符的字符，它用一个 HTML 的 tag.words[j]= '<mark>' + q + '</mark>' 来增加一个格式化代码。
4. 其中的 words.join('') 用于把所有的数组中的元素连接起来生成一个一维的字符串用于输出。

程序运行的结果类似于下面所示：

A word to search for? who

There once lived, in a sequestered part of the county of Devonshire, one Mr. Godfrey Nickleby: a worthy gentleman, who, taking it into his head rather late in life that he must get married, and not being young enough or rich enough to aspire to the hand of a lady of fortune, had wedded an old flame out of mere attachment, who in her turn had taken him for the

same reason. Thus two people who cannot afford to play cards for money, sometimes sit down to a quiet game for love.

数组让输出程序能够使用诸如大容量数字或文本的大数据成为可能。

（中间重复第一部分内容）

8.1.1 推荐时间安排：1（55 分钟课堂周期）

教学天数	主题
2 天（复数 days）	课堂计划 I
1 天	课堂计划 II
1 天	课堂计划 III
1 天	课堂计划 IV & V
1 天	课堂计划 VI

8.1.2 标准

CSTA 标准	CSTA 部分	CSTA 涵盖的学习目标
水平 3A (9-12 年级)	计算思维	解释顺序、选择、迭代和循环如何建立块算法。
水平 3A (9-12 年级)	计算思维	比较海量数据收集分析的技术
水平 3A (9-12 年级)	协作	描述查找和收集小大规模的数据集的技术
水平 3A (9-12 年级)	协作	部署关于不同问题的不同数据收集技术
水平 3A (9-12 年级)	计算思维	比较和对比简单的数据类型和它们的使用
水平 3A (9-12 年级)	电脑和通信设备	讨论对应用程序功能进行修改的影响

8.1.3 教学建议

鼓励学生在文本模式和块模式直接转换来不多扩充他们自己以致超越他们原有的水平。教学计划建议在不同模式间进行转换。学生们最好在他们尝试着理解逻辑的整体流程时使用块模式。在另外的时候用文本输入各种关键词更加方便，因为块模式也许不能包含所有需要的块。这很重要，因为用文本键入能够连接到 JavaScript 的更加丰富的可用库。

8.1.4 教学计划 I

这节课介绍 CoffeeCript 的数组。

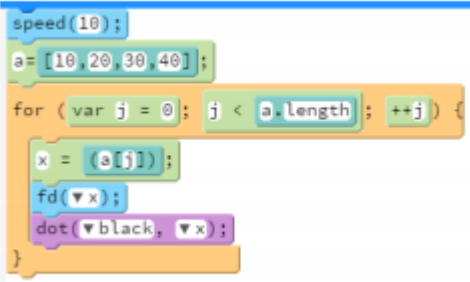
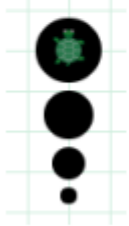

目录摘要	教学建议	时间
------	------	----

<div><p><u>Code:</u></p><pre>for color in [red, orange, yellow]   dot color, 100   fd 30</pre></div> <div><p><u>Code in text:</u></p><pre>for color in [red, orange, yellow]   dot color, 100   fd 30</pre></div> <div><p><u>Output:</u></p></div>	<p>导出程序“rainbox Colors”并演示运行程序。</p> <p>指出这在第四章的教学计划 III 中使用过。</p> <p>解释 color 是一个由一个数组中的红，橙，黄三种颜色转换而来的变量。</p> <p>在[]括号之间的是存储在数组数据结构里的各种颜色。命令'dot'在数组中起作用并且画了相对应颜色的小点。</p> <p>鼓励学生玩各种颜色。</p> <p>变量：程序可以获取一个包含点大小的数组而不是颜色本身改变自己的值。</p> <p>For 循环类似于这样：</p> <p>The for loop would look like this:</p> <pre>for x in [10,20,30,40]   dot red,x   fd 30</pre>	<p>示范：15 分钟</p> <p>学生实践：20 分钟</p>
---	--	-----------------------------------

8.1.5 教学计划 II

这里对数组的第一次介绍是给学生展示如何遍历由一系列已知的元素组成简单的数组。程序绘制的元素的大小保存在数组中。敲下所示的代码（用块模式和文本模式）或者导出代码而后通过代码引导学生，解释概念给学生。

目录摘要	教学建议	时间
------	------	----

<p><b>Code: Blocks</b></p>  <p><b>Code: Text</b></p> <pre> speed(10); a=[10,20,30,40]; for (var j = 0; j &lt; a.length; ++j) {   x = (a[j]);   fd(x);   dot(black, x); } </pre> <p>Array values: 10,20,30,40</p> 	<p>敲下代码（或导出到工程中）而后通过代码引导学生</p> <p>展示数组是如何声明的。</p> <p>解释长度如何表示数组的大小。</p> <p>解释数组为什么从 0 开始以及 for 循环如何遍历整个数组。</p> <p>解释 x 代表数组的某一位置的值，所以有 j=0, x=10, j=3, x=40。</p> <p>Turtle 移动依靠 x 的每个值而且点的大小的绘制也依靠 x 的值。</p> <p>教学技巧：一个愉快的联系是要让学生们改变数组的值以使得他们不墨守成规。这能够让学生们看到圆点大小的改变依赖于数组中索引的位置。让他们使用一些如 10,50,15 和 30 这样的大数字来观察圆点变大或变小。</p> <p>数组值：10,50,15,39</p> 	<p>示范：30 分钟</p>
---	--	-----------------

### 8.1.6 教学计划 III

这节课计划将展示使用后入先出的栈的概念如何从一个数组中增加或删除一个元素。这个示例程序可以用于给学生展示数组中的元素是如何被加入或删除的。使用流行的特征来演示即使元素被移除，数组为那个元素创建的空间也只会变成无主之物。将程序分发给学生，让他们用程序体验并理解数组行为和栈的特性。

目录摘要	教学建议	时间
------	------	----

### Code

```
var stack = [];  
pen(green, 25);  
speed(Infinity);  
button('F', function() {  
  fd(10);  
});  
button('R', function() {  
  rt(30);  
});  
button('Push', function() {  
  dot(crimson, 50);  
  var record = {  
    xy: getxy(),  
    dir: direction()  
  };  
  stack.push(record);  
});
```

```
stack.push(record);  
});  
button('Pop', function() {  
  if (!stack.length) {  
    home();  
    return;  
  }  
  var record = stack.pop();  
  jumpto(record.xy);  
  turnto(record.dir);  
  dot(pink, 50);  
});
```

使用这个程序来示范基础的数组功能。

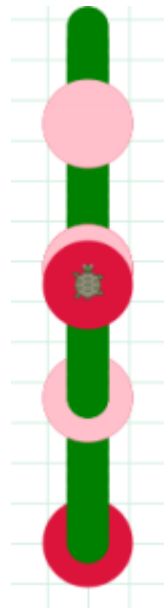
使用 F 按钮来使乌龟向前移动。

使用 Push 来在屏幕上创建一个红色的圆点，代表一个添加到数组中的元素。

使用 Pop 创建粉红色的圆点。这个点比红点更亮。这个例子可以用于表示一个元素被移出数组。

使用 'F' 按钮和 'Push' 按钮的组合来示范一个元素可以被替换到数组结构中的任意位置，只要那里有用于支持此元素一个整形的索引位置。

输出：



示范：20 分钟

学生实践：55 分钟

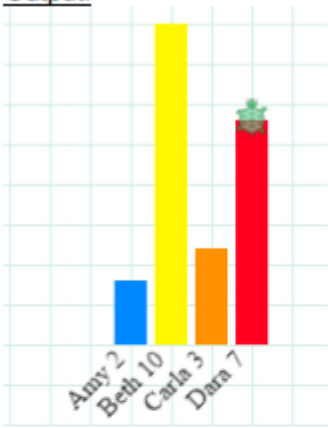


<p><b>Code: Blocks</b></p> <pre>stack = []; pen(▼green, 25); speed(Infinity); button('F', function() {   fd(▼10); }); button('Push', function() {   dot(▼crimson, ▼50);   stack.push([getxy()]); }); button('Pop', function() {   if (!stack.length) {     home(●);     return;   }   // [xy, b] = stack.pop();   jumpto(▼xy);   turnto(▼b);   dot(▼pink, ▼50); });</pre>		
<p>拓展活动：让学生们创建他们自己版本的两个程序。复制给他们两个实例程序，让他们能够思考他们自己的程序。鼓励学生创建第一个程序的他们的版本（你可以选择把程序 1 当做他们的实验活动，让他们自己设计）第二个程序只用于示例目的。你也可以让让学生思考它（思考拼写错误）并帮助他们更好地理解程序的运行。（55 分钟）</p>		

8.1.7 教学计划 IV

这节课计划示范数组关于保存数据的使用。学生们将学会遍历数组并用数组中的数据生成数据图。这节课将展示一个条状图如何用程序中的两个数组画出来。这节课将充分地显示出 Penci Code 能够在执行过程中实时生成可视化反应的特点。

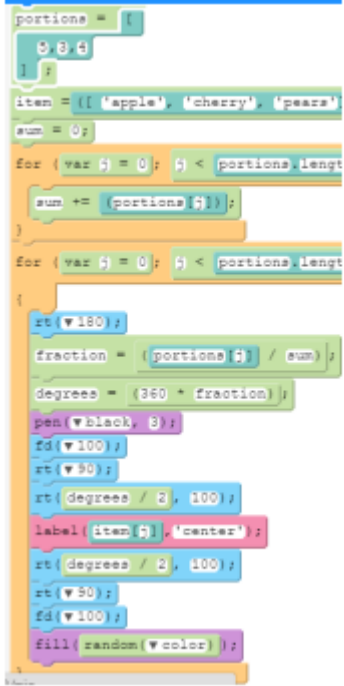

目录摘要	教学计划	时间
------	------	----

<p><b>Output:</b></p>  <p><b>Text Code:</b></p> <pre>data=[2, 10, 3, 7]; labels=['Amy','Beth','Carla','Dara']; for (var k = 0; k &lt; data.length; ++k) {   jumpto(25 * k, 0);   lt(45);   label(labels[k] + ' ' + data[k], 'bottom left rotated');   rt(45);   pen(random(color), 20, 'butt');   fd(data[k] * 20); }</pre> <p><b>Block Code:</b></p> <pre>data = [2, 10, 3, 7]; labels = ['Amy', 'Beth', 'Carla', 'Dara']; for (var k = 0; k &lt; data.length; ++k) {   jumpto(25 * k, 0);   lt(45);   label(labels[k] + ' ' + data[k], 'bottom left rotated');   rt(45);   pen(random(45), 20, 'butt');   fd(data[k] * 20); }</pre>	<p>通过程序代码引导学生。</p> <p>步骤 1：运行程序并演示输出。</p> <p>步骤 2：用 1-D 数组中的数据标记数组。</p> <p>步骤 3：Labels 构成 x 轴数据。Data 构成 y 轴数据并确定条形图的长度。</p> <p>步骤 4：随机彩色生成器决定图表中的颜色。</p> <p>步骤 5：展示程序中的循环语句。教给学生循环语句如何遍历数组。（data.length）</p> <p>步骤 6：展示给学生乌龟在 y 轴上向一个常量(数据数组中的值)移动。</p> <pre>fd(data[k] * 20);</pre> <p>步骤 7：展示给学生乌龟在 x 轴上向一个常量(数据数组中的值)移动。</p> <pre>jumpto(25 * k, 0); lt(45);</pre> <pre>jumpto(25 * k, 0); lt(45);</pre>	<p>示范：30 分钟</p> <p>学生实践：25 分钟</p>
<p>拓展活动：让学生用不同类型的数据值创建不同的数组并用提供的代码创建条形表。（55 分钟）</p>		

### 8.1.8 教学计划 V

在这节课中，学生们将用存储在一维数组中的简单数据创建一个饼图。这节课类似于条形图那节课。但是来自一个数组中的数据的表的绘制是不同的。

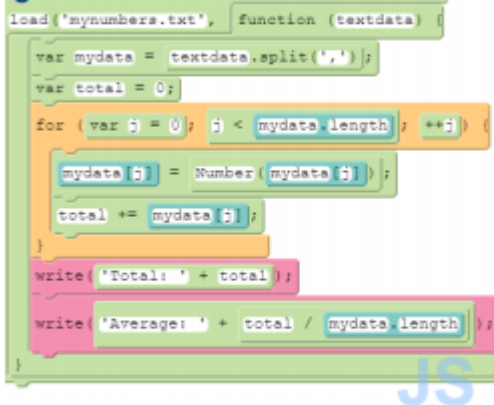
目录摘要	教学建议	时间
------	------	----

<p><b>Text Code:</b></p> <pre> portions = [ 5,3,4]; item =(['apple', 'cherry', 'pears']); sum = 0; for (var j = 0; j &lt; portions.length; ++j) {     sum += (portions[j]); } for (var j = 0; j &lt; portions.length; ++j) {     rt(180);     fraction = (portions[j] / sum);     degrees = (360 * fraction);     pen(black, 3);     fd(100);     rt(90);     rt(degrees / 2, 100);     label(item[j], 'center');     rt(degrees / 2, 100);     rt(90);     fd(100);     fill(random(color)); } </pre> <p><b>Block Code:</b></p> 	<p>步骤 1: 运行程序来示范饼图的创建。</p> <p>步骤 2: 改变数组中的值来展示它是如何影响结果图的。</p> <p>步骤 3: 单步调试代码来展示饼状图是如何计算的。</p> <p>步骤 4: 计算分数的公式</p> <pre> fraction = ( portions[j] / sum ) </pre> <pre> fraction = (portions[j] / sum); </pre> <p>步骤 5: 计算读书的公式。</p> <pre> degrees = (360 * fraction); </pre> <pre> degrees = (360 * fraction); </pre> <p>输出:</p> 	<p>示 范: 30 分 钟</p> <p>学 生 实 践: 1 课 时</p>
<p>拓展活动: (Ask 拼写错误) 让学生用不同类型的数据值创建不同的数组并用所给的代码创建各种描绘不同类型数据的饼图。(55 分钟)</p>		

8.1.9 教学计划 VI

这节课计划阐释如何在一个文本文件中查找一个元素。课程计划有两部分。第一部分包括给学生展示如何加载（打开）一个文件并读取它。第二部分有一个遍历代码用于查找一个文件

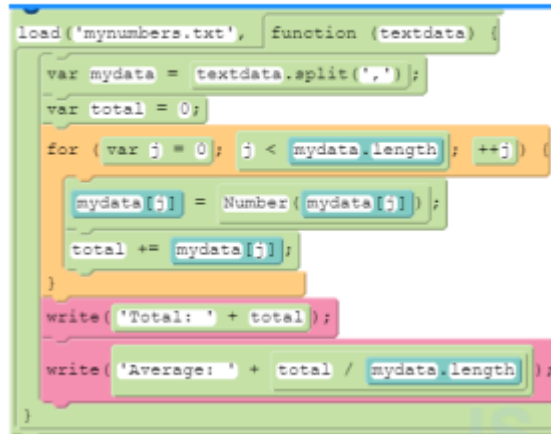
中的元素。

目录摘要	教学建议	时间
<p><b>Code: Text</b></p> <pre>load('mynumbers.txt', function (textdata) {   var mydata = textdata.split(',');   var total = 0;   for (var j = 0; j &lt; mydata.length; ++j) {     mydata[j] = Number(mydata[j]);     total += mydata[j];   }   write('Total: ' + total);   write('Average: ' + total / mydata.length); });</pre>	<p>步骤 1：导出所有“SearchingNumbers”程序。</p> <p>步骤 2：记得待在 JavaScript 模式下。</p> <p>步骤 3：运行程序并演示输出。</p> <p>步骤 4：单步调试程序。</p> <p>步骤 5：解释文件打开。当需要重用语法时参考核心概念。</p> <p>步骤 6：对于那个程序，数据文件（“mynumbers.txt”）可以在同名文件夹中找到。学生们可以通过在 Pencil Code 中创建一个新的文件来创建他们自己的数据文件而后点击“Save”按钮。</p> <p>步骤 7：现在解释数据是如何保存在一个数组中的，以及程序如何在发现值增长的时候查找一个相匹配的项的。</p> <p>步骤 8：用 for 循环追踪代码并解释遍历到的数据。</p> <p>注意：学生可以追踪 mydata[j],j 以及所有的值并在课堂上分享。</p>	<p>示范：30 分钟</p> <p>学生实践：60 分钟</p>
<p><b>Code: Blocks</b></p> 		
<p><b>Output:</b></p> <pre>Total: 15 Average: 3</pre>		

### Code: Text

```
load('mydata.txt', function
(textdata) {
  var words =
textdata.split(/\b/);
  read('A word to search for?',
function(q) {
    for (var j = 0; j <
words.length; j++) {
      if (words[j] == q) {
        words[j] = '<mark>' + q +
'</mark>';
      }
    }
    write(words.join(''));
  });
});
```

### Code: Block



```
load('mynumbers.txt', function (textdata) {
  var mydata = textdata.split(',');
  var total = 0;
  for (var j = 0; j < mydata.length; ++j) {
    mydata[j] = Number(mydata[j]);
    total += mydata[j];
  }
  write('Total: ' + total);
  write('Average: ' + total / mydata.length);
})
```

### Output:

A word to search for? Betty  
Betty, bought, some butter. But, the butter was bitter.  
better. But the bitter butter, made the better butter bitt

步骤 1: 导出 SearchingText 中的程序。

步骤 2: 运行程序。

步骤 3: 让学生单步调试程序并解释给你听。

注意: 两个程序都用到了 split() 函数。为了更好地解释 split 以及如何使用它, 看:

[http://www.w3schools.com/jsref/jsref\\_split.asp](http://www.w3schools.com/jsref/jsref_split.asp)

示范:  
30 分钟

学生实践:  
35 分钟