第 1 课：第一步

**喜欢教这些课程吗？发现错误了吗？在 the-cc.io/feedback 上分享反馈。**

## 介绍

在本入门课程中，学习者将使用 Python 编写和执行他们的第一个程序。他们将介绍显示消息、为变量赋值以及从键盘接收输入的基础知识。

他们将熟悉一个与他们可能习惯的基于块的编程环境完全不同的编程环境。在这种环境中，他们需要熟记所有可以使用的结构，而不是将选项摆在他们面前。这也是一个如果他们弄错一个字母或符号就会出现错误的环境。

本课（和本单元）的主要目标之一是支持他们进行这种过渡，通过提供与他们已经熟悉的概念的联系，并建立他们克服常见障碍的信心。

在进行任何编程之前，将向学习者介绍什么是算法和程序，以及它们有何不同。通过这次讨论，他们将开始理解用正式语言表达指令意味着什么，以及这些指令最终如何由机器执行。

## 学习目标

* 描述什么是算法和程序以及它们有何不同
* 回想一下，用编程语言编写的程序需要翻译才能被机器执行
* 编写简单的 Python 程序，用于显示消息、为变量赋值和接收键盘输入
* 查找并更正常见语法错误

## 关键词汇

算法、程序、编程语言、程序转换和执行、解释器、编程环境、输入、输出、变量、赋值

## 

## 制备

**学科知识：**

* 您将需要了解算法和程序之间的差异。
* 您需要熟悉如何使用 Python IDE。
* 您将需要能够找到并纠正 Python 程序中的语法错误。
* 您需要熟悉在 Python 中使用输出、输入和赋值。
* 您需要了解新手学习者可能遇到的常见误解。请参阅 “教学法笔记”部分中的常见误解，了解与本课相关的误解列表。

**您将需要：**

* 幻灯片 — 请注意，某些幻灯片包含动画
* 活动工作表
* Python 解释器和 IDE — 我们建议使用 Mu 编辑器 （[codewith.mu](https://codewith.mu/)） 或在线环境，例如 [Repl.it](https://repl.it/);如果您使用的是 Mu 编辑器，那么您可能会发现以下指南有助于帮助您做好准备： Mu [入门](http://rpf.io/mu) （rpf.io/mu）

**您可能需要：**

* 用于为“遵循说明”活动制作纸条的打印输出
* “第一步”活动的 Python 代码
  + [任务 1 — solution](https://the-cc.io/py-output-1) （the-cc.io/py-output-1）
  + [任务 2 — 解决方案](https://the-cc.io/py-assignment-11) （the-cc.io/py-assignment-11）
  + [任务 3 — 解决方案](https://the-cc.io/py-assignment-12) （the-cc.io/py-assignment-12）
  + [任务 4 — 解决方案](https://the-cc.io/py-input-1) （the-cc.io/py-input-1）
* 关于输出、赋值和输入的 Python 备忘单

## 评估机会

您可以评估学习者对“第一步”工作表的回答。您还可以通过观察来评估学习者，例如，通过结对编程评估学习者如何互动并协作解决问题。

## 一目了然

|  |  |
| --- | --- |
| **入门活动**  8 分钟 | **按照说明操作**  要介绍算法的概念，请学习者执行“所有算法的祖父”。 |
| **活动 1**  10分钟 | **算法、程序和语言**  概述算法和程序之间的主要区别：语言。说明用高级语言（如 Python）执行程序涉及翻译它，并为学习者介绍语法错误做好准备。 |
| **活动 2**  30分钟  6分钟介绍  每个任务 6 分钟 | **Python 的第一步**  让学习者熟悉开发环境，并使用工作表指导他们完成涉及输出、作业和输入的介绍性示例。 |
| **活动 3**  2 分钟 | **Python 回顾**  要回顾本课中涵盖的 Python 内容，请学习者找出负责输出、输入和分配的代码片段。 |

## 

## 大纲图

请注意，幻灯片在右上角标记了活动，以帮助您浏览课程。

*\*时间安排是粗略的指南*

|  |  |
| --- | --- |
| **入门活动**  （幻灯片 2-3）  8 分钟 | **按照说明操作**  当学习者进入教室时，显示幻灯片 2 以在黑板上显示一系列说明。说明有文本和视觉表示，因此使用两种不同的“语言”。询问学习者是否认为他们可以理解并遵循这些说明。  **注意：** 幻灯片上的算法具有重要的历史意义。尽管学习者最初可能没有意识到这一点，但它是欧几里得最大公因数算法的几何版本。唐纳德·克努斯（Donald Knuth）称其为“所有算法的祖父，因为它是幸存至今的最古老的非平凡算法”。  请志愿者执行说明，以 12 厘米和 18 厘米的纸条为起点（您可以使用提供的打印输出来切割任何长度的纸条）。要求他们在完成后测量条带的长度;两次迭代后，条带的长度应为 6cm（因为 12 和 18 的最大公因数为 6）。以 6cm 和 15cm 的纸条为起点重复该过程，这应该在三次迭代后产生两个 3cm 的纸条（因为 6 和 15 的 LCF 是 3）。  **一种算法**  显示幻灯片 3。告诉学习者，他们刚刚遵循的方法是由欧几里得在公元前 300 年左右记录的，它描述了寻找最大公因数的过程。引入术语“算法”来描述该方法。  询问学习者以前是否听说过“算法”一词。他们可能无法提供具体的定义，但他们可能对它的含义有一个大致的了解，尤其是与Facebook（一种用于创建Feed的算法）、Google（一种搜索算法）或Amazon（一种用于推荐产品的算法）有关。 |
| **活动 1** （幻灯片 6-12）  10分钟 | **算法、程序和语言**  显示幻灯片 6，并定义什么是算法。强调语言在表示、理解和执行算法方面的重要性;如果不理解算法所表示的语言，就不可能在“遵循指令”活动中执行算法。指出有些算法只是用图像直观地表达，并询问学习者是否能想到一些例子（例如乐高或宜家组装说明）。  接下来，提供什么是程序的定义（幻灯片 7）。关注算法和程序之间的主要区别：语言。程序是按照编程语言的严格规则表示的算法。这是 **必须** 的，因为程序最终将由机器执行。我们不能用自然语言编写程序，因为它是模棱两可和复杂的。编程语言被设计得足够明确和简单，用这些语言编写的程序可以被机器翻译和执行。  **Python 解释器**  简要描述一个 Python 程序为了执行它所经历的过程。解释一下，在其原始形式中，Python 程序只是文本，没有特殊含义。它的指令不能由我们的机器直接执行。我们需要运行一个名为 Python **解释器**的程序，它将遍历程序、翻译它并执行它。  如果您计划使用在线解释器，例如 Repl.it 或 Trinket，请向学习者提及 Python 解释器不需要在您的计算机上本地运行。  **语法（和错误）**  为了让学习者为语法错误入门做好准备，请简要讨论语法规则。解释编程语言有语法规则，就像自然语言一样。然而，虽然人类有时可以在不遵守语法规则的情况下正确推断出含义，但当解释器试图翻译和执行违反编程语言语法规则的程序时，情况并非如此。  说明学习者还不熟悉语法错误，因为在 Scratch 中不可能犯语法错误。但是，在基于文本的编程语言中，语法错误很常见且容易犯。显示幻灯片 11 以显示 Python 代码的简短片段，并说明任何与大写或小写的偏差，以及任何缺少的逗号、冒号、引号或括号，都会导致语法错误，解释器将无法翻译程序。展示幻灯片 12 并告诉学习者，他们不应该因语法错误而气馁。 |
| **活动 2**  （幻灯片 13-24）  30分钟  6分钟介绍  每个任务 6 分钟 | **Python 的第一步**  首先，向学习者展示传统的“Hello world”程序，并告知他们这是他们将要编写的第一个程序。  **实时编码：Hello world**  告知学习者他们将进行 **结对编程** （在下面的“教学法笔记”部分了解更多信息）。将他们配对，简要解释他们的角色，并要求他们在每项任务结束时切换。  打开您将使用的开发环境，并让学习者进行演练。帮助他们找到他们需要的功能。执行 **实时编码** （在下面的“教学法说明”部分了解更多信息）：输入“Hello world”程序，确保学习者跟随您。描述他们应该注意的语法方面：括号、引号等。  演示如何运行程序。解释这会调用解释器，并向他们展示他们应该期望看到的结果。制作几个常见的语法错误，并演示如何阅读错误消息，然后找到它们所引用的行并更正错误。  **第一步**  分发“第一步”工作表，并要求学习者开始处理各个任务，仍然使用结对编程。当学习者完成一项任务时，请他们暂停并将注意力转向您，以便您可以简要解释每个任务的代码中的一些细节（详见“评论”幻灯片）。这些包括他们需要注意的概念，如幻灯片中突出显示的那样，即变量、值和赋值，以及语法的微妙之处。  **注意：**一些幻灯片显示了与 Python 程序（相关部分）相对应的 Scratch 块，以便学习者可以与他们已经熟悉的概念进行关联。在线课程 Scratch to Python*： Moving from block-based to text-based programming* 包括[一个将 Scratch 和 Python 语法相提并论的步骤](http://rpf.io/scratchtopython)，以及最后的相关备忘单。 |
| **活动 3**  （幻灯片 25）  2 分钟 | **Python 回顾**  要回顾本课中涵盖的 Python 内容，请学习者找出负责输出、输入和分配的代码片段。 |
| **作业** | 这节课没有作业。但是，如果学习者有兴趣应用他们所学的知识，您可以向他们指出 Raspberry Pi 基金会的关于我 （rpf.io/about-me） 项目。这个项目还将为他们在下一课中遇到的一些想法做好准备。 |

### 关于教学法的说明常见误解

在这些课程中，您应该了解新手程序员可能出现的常见误解。以下误解摘自 Juha Sorva 的博士论文《*编程入门教育中的可视化程序模拟*》的附录 A，与本课相关。该列表将帮助您避免使用任何可能导致这些误解的示例或解释，并且还可以帮助您发现它们，如果它们出现在您的学习者中。

**M1、2、7：** 计算机知道或能够推断出程序或一段代码的意图，并采取相应的行动。机器懂英语。

**M4：** 系统不允许不合理的操作。

**M150：** 难以理解输入函数调用对执行的影响。

### 结对编程

结对编程是一种教学方法，涉及学习者**以结构化的方式**共同开发程序。学习者扮演驾驶员和导航员的角色，并定期交换角色。驱动程序控制鼠标和键盘来编写代码，而导航器则提供指导和说明。您可以在课程计划末尾引用的资源中找到有关结对编程的更多信息。

由于学习者配对对于这种方法的成功很重要，因此强烈建议您在上课前决定这些配对。

### 实时编码

格雷格·威尔逊（Greg Wilson）在他的《[*共同教授技术》（Teaching Tech Together*](https://teachtogether.tech/)）一书中对实时编码的描述如下：“老师在全班面前编写代码，而学习者则跟着编写代码，输入代码并边走边运行”。他称其为“最有效的编程教学方式”，并继续列举优势并引用相关研究。

实时编码不应即兴创作。这是一个有计划、结构良好的“表演”，但它也为教师提供了积极解决意外问题的机会。您可以在课程计划末尾引用的资源中找到有关实时编码的更多信息。

### 草图绘制、漫游和跟踪表

课程幻灯片中引用了“草图”变量及其值的概念。这与使用绘图来说明在程序执行过程中如何修改程序状态（例如变量的值和数据结构的内容）的想法有关。这可以为跟踪表提供一种可视化的替代方法，对于现阶段的学习者来说，跟踪表可能不那么繁琐。

Python Tutor [（pythontutor.com/visualize.html](http://pythontutor.com/visualize.html)） 是一个非常有用的工具，用于逐步演练 Python 程序和可视化程序状态。

## 笔记

### Python 开发环境

本单元中没有假设您将使用特定的开发环境来编写 Python 程序。有很多选择，最适合您的可能取决于您的情况。

在撰写本文时，我们的建议是使用 [Mu 编辑器](https://codewith.mu/) （codewith.mu），这是一个友好、简约、跨平台的环境，专为新手程序员设计。使用在线编码环境（例如 [repl.it](https://repl.it) 和 [trinket.io）也有好处](https://trinket.io/)。

### 在 Python 中显示消息

在这些资源中，每当需要显示涉及多个值的消息时，我们都会使用带有逗号分隔值列表作为参数的 print。  
  
例如：

print（lucky，“也是我的幸运数字”，姓名）

还有其他一些方法可以实现这一点（例如使用 + 运算符或 f 字符串进行字符串连接），但我们选择在 8 年级使用这种方法，因为我们认为它在语法上更简单。

### 在 Python 中获取键盘输入

在这些资源中，每当需要显示提示和接收键盘输入时，我们使用 print 来显示提示，然后使用 input，即使 input 也可用于显示提示：

print（“你叫什么名字？

名称 = input（）

而不是：

name = input（“你叫什么名字？

我们认为这是可取的，因为它清楚地区分了输入和输出，并导致学习者更容易管理的代码。

## 其他来源

* [Raspberry Pi 基金会的 Mu 项目 （rpf.io/mu）](https://rpf.io/mu) 入门
* 教学法快速阅读： [结对编程](https://the-cc/qr03)
* 教学法快速阅读： [实时编码](https://the-cc/qr05)

资源会定期更新 - 最新版本可在以下网址获得： [the-cc.io/curriculum](http://the-cc.io/curriculum)。



本资源由 [Raspberry Pi 基金会](https://www.raspberrypi.org/) 根据 Creative Commons 署名-非商业性使用-相同方式共享 4.0 国际许可获得许可。要查看此许可证的副本，请访问，请参阅 [creativecommons.org/licenses/by-nc-sa/4.0/](https://creativecommons.org/licenses/by-nc-sa/4.0/)。