

# 学习Python编程的11个资源

2013/07/06 · [Python](#) · [1 评论](#) · [Python](#)

分享到: [1](#)  
本文由 [伯乐在线](#) - [atupal](#) 翻译。未经许可，禁止转载！  
英文出处: [Alex Ivanovs](#)。欢迎加入[翻译组](#)。

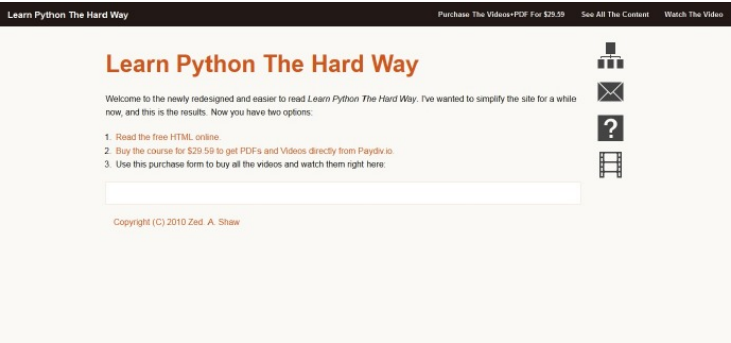
用 Python 写代码并不难，事实上，它一直以来都是被声称为最容易学习的编程语言。如果你正打算学习 web 开发，Python 是一个不错的选择，甚至你想学游戏开发也可 以从 Python 开始，因为用 Python 来构建游戏的资源实在是太多了。这是一种快速 学习语言的一种方法。

许多程序员使用 Python 作为初学语言，然后接着是像 PHP 和 Ruby 这样的语言。它也是[2014最热门的 web 开发语言之一](#)，并被高度推荐学习。但是，怎么学习 Python 呢？我们应该从哪里开始学？

我在这里就是为你解决这个问题。因为我自己学习编程的时候也依靠了很多这些资源，并开始开发。当然这仅仅是一个友好的建议，最好的方法还是在做中学，这里推荐的一些书和资源只是为了把你引向正确的方向。

刚开始学习的时候可能会特别令人沮丧，不过一旦你学完了基本的东西，之后的学习对你 来说就很自然了，你不需要思考就知道要怎么做。

## 笨办法学 Python



这本书绝对是最简单的学习 Python 的方法。你会被如此简单的基础学习感到震惊，你会真实地感受到自己的学习进程，一步步的获取新的知识。我也了解到它非常鼓励 学习者取尝试并创造自己的程序。

程序可能会很小，但是它们肯定会有助于你更好的理解语言本身和它的语法。Python 非常流行，所以当 you 遇到了问题是，很有可能在像 StackExchange 这样的网站已经 一些答案了，当你遇到问题需要解决时尽管去谷歌。

你将会学到:

- 在所有的平台上安装 Python 编程环境
- 写 Python 程序
- 理解 Python 语法和文档
- 像一个程序员一样思考
- 还有更多的东西

本书的 HTML 在线版是完全免费的，大部分人都是使用的在线版。我愿意鼓励你捐赠/购买完整的书，毕竟作者花了很大的力气来完成它。如果你发现使用视频学习更容易的话，它的高级版就含有视频。

## [Online Python Tutor: Python 初学者的好帮手](#)

Write your Python code here:

```
1 def listSum(numbers):
2     if not numbers:
3         return 0
4     else:
5         (f, rest) = numbers
6         return f + listSum(rest)
7
8 myList = (1, (2, (3, None)))
9 total = listSum(myList)
```

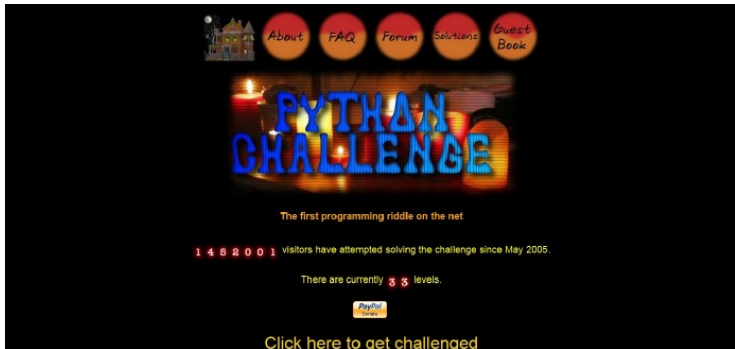
Execute code using Python 2.7, hide frames of exited functions, inline primitives and nested objects, hide environment parent pointers, draw references using arrows, and show everything.

Visualize Execution

Clear

[Online Python Tutor](#) 是由 Philip Guo 开发的一个免费教育工具，可帮助学生攻克编程学习中的基础障碍，理解每一行源代码在程序执行时在计算机中的过程。

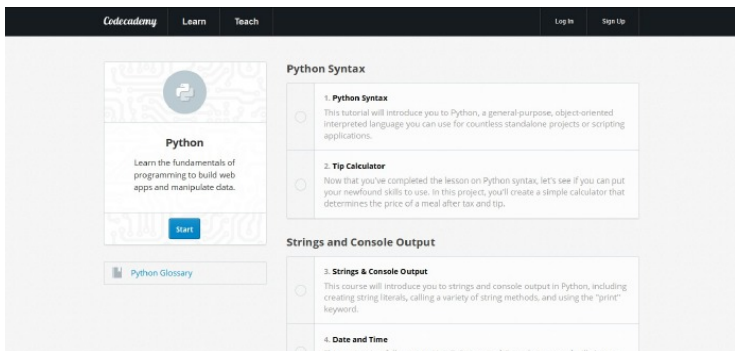
### [The Python Challenge](#)



如果你在编程生涯中从没做过这样的挑战的话，这可能有点棘手。但是配和上面介绍的那本书的话也不是很难。你应该试一试。它有 33 个关卡(谜题)，可以用你 Python 编程技能解决。

几百万的人都在尝试解决这个，即使你不能完成所有的关卡，你也会将学到很多新的东西，尤其是批判性思维和发散思维。你的大脑会过热，但这是编程！

### [Learn Python Programming @ Codecademy](#)

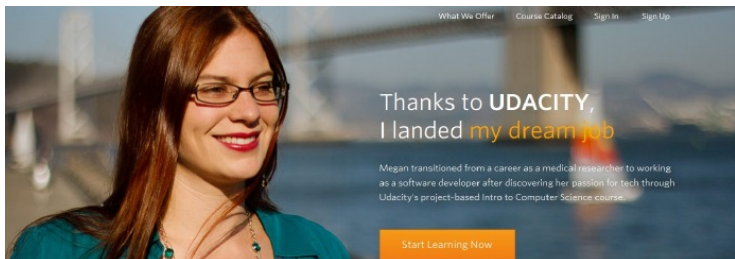


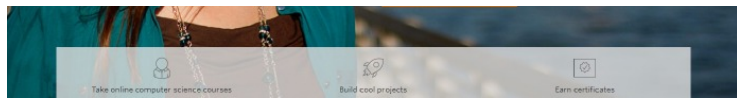
你会看到许多“精英”程序员骂这个网站，但那并不重要。我们想要做的是看/测试一门编程语言的基本语法是怎么工作的，以及当和函数组合时可以做什么。而不是打印一个‘Hello World’。

在这个 Codecademy 的课程中你将会学到如何用文件，如何使用循环。什么是函数以及他们能用来做什么。这些都是很基础的，但是对初学者很有好的东西。当需要帮助时有一个交流的论坛，但通常所有东西都能被理解。

你不需要安装任何工具，如果你想在你自己的电脑上重写一遍代码以便更好地理解它，你可能需要的唯一东西就是一个 Notepad++ 编辑器。我就是这样做的，并且我向所有想学习编程的人推荐 Codecademy，不管是 Python 还是其他的语言。

### [Intro to Computer Science @ Udacity](#)





Udacity 提供了一个很棒的免费课程，带你引入 Python 编程学习，学习搜索引擎的更多东西，以及如何构建你自己的小型网络爬虫。它的确是一个值得参与的有趣的课程，并且还提供了额外的引导和社区支持。

[http://www.youtube.com/watch?v=Pm\\_WAWZNbdA](http://www.youtube.com/watch?v=Pm_WAWZNbdA)

你可以报名成为高级学生以获取导师的引导，并且可以在课程末得到一个证书。或者你可以自己使用免费的课件学习。不幸的是，高级课堂的容量有限，所以你 得在申请后进入等待列表。

它总共有 11 个课程，每一个都有很赞的文档。取它的官方主页查看更多的信息或者查看你可能想问的问题的答案。

## Google's Python Class

Google

Developers

Educational Materials

Search

Q

Home

Products

Conferences

Showcase

Live

Groups

Products > Educational Materials

Educational Materials 

891.3K

Python Course

Python Set Up

Introduction

Strings

Lists

Sorting

Dicts and Files

Regular Expressions

Utilities

Google's Python Class

Welcome to Google's Python Class -- this is a free class for people with a little bit of programming experience who want to learn Python. The class includes written materials, lecture videos, and lots of code exercises to practice Python coding. These materials are used within Google to introduce Python to people who have just a little programming experience. The first exercises work on basic Python concepts like strings and lists, building up to the later exercises which are full programs dealing with text files, processes, and http connections. The class is geared for people who have a little bit of programming experience in some language, enough to know what a "variable" or "if statement" is. Beyond that, you do not need to be an expert programmer to use this material.

To get started, the Python sections are linked at the left -- [Python Set Up](#) to get Python installed on your machine, [Python Introduction](#) for an introduction to the language, and then [Python Strings](#) starts the coding material, leading to the first exercise. The end of each written section includes a link to the code exercises for that section's material. The lecture videos parallel the written materials, introducing Python, then strings, then first exercises, and so on. At Google, all this material makes up an intensive 2-day class, so the videos are organized as the day-1 and day-2 sections.

This material was created by [Rick Dalato](#) working in the engEDU group at Google. Special thanks for the help from my Google colleagues John Cox, Steve Glassman, Piotr Kaminski, and Antoine Picard. And finally thanks to Google and my director Maggie Johnson for the enlightened generosity to put these materials out on the internet for free under the under the [Creative Commons Attribution 3.0 license](#) -- share and enjoy!

Tip: Check out the [Python Google Code University Forum](#) to ask and answer questions

Google 它本身就被一大堆 Python 代码驱动着，只有他们支持社区并且愿意 帮助其他人学习语言才有意义。这是我曾看过的最喜欢的指南/课程之一。它真的 很详细，而且适合初学者，看起来也很有趣。

<http://www.youtube.com/watch?v=tKTZoB2VjuI>

去看几分钟它的第一堂课，看看你是否喜欢那个老师，然后决定是否开始学习！Python 官方主页有所有的这些练习题和例程的链接。

## 简明 Python 教程

SWAROOP C H

WELCOME

BLOG

A BYTE OF PYTHON

A BYTE OF VIM

BUY PAPER BOOKS

CONTACT

ABOUT

ARCHIVES

Python

A Byte of Python

'A Byte of Python' is a free book on programming using the Python language. It serves as a tutorial or guide to the Python language for a beginner audience. If all you know about computers is how to save text files, then this is the book for you.

This book is written for the latest Python 3, even though Python 2 is the commonly found version of Python today (read more about it in [Python 2 versus 3 section](#)).

Who Reads A Byte of Python?

Here are what people are saying about the book:

“

This is the best beginner's tutorial I've ever seen! Thank you for your effort.

”

ABOUT

Swaroop C H is a web backend engineer. Has written popular free books on Python and Vim. Popular blog in India. Previously worked at Yahoo!,

非常像 LPTHW，但是在如何制定你的完美学习计划并实施，以及如何开始第一步上 提供了更深的引导，而不会让你丧失自信心。对于那些想学 Python 的人来说它被认为是 最好的初学者教程之一，当然你要先看看前几章，看看自己是否喜欢它的写作风格。

## Think Python

Think Python

How to Think Like a Computer Scientist

by Allen B. Downey

[Buy this book at Amazon.com](#)

Download [Think Python in PDF](#)

Read [Think Python in HTML](#)

Example programs and solutions to some problems are [here](#) (links to specific examples are in the book)

Description

Think Python is an introduction to Python programming for beginners. It starts with basic concepts of programming, and is carefully designed to define all terms when they are first used and to develop each new concept in a logical progression. Larger pieces, like recursion and object-oriented programming are divided into a sequence of smaller steps and introduced over the course of several chapters.

Like this book?

Think Python

Like

Are you using one of our books in a class?

Please consider filling out [this short survey](#).

Think Python

How to Think Like a Computer Scientist

Think Python 是一个面向 Python 初学者的一个指南。它从编程的基本概念教起，而且它很 仔细地定义了所有的术语当他们第一次用到时，并且很有逻辑性地展现新的概念。像递归和面向 对象这样大一点的知识块被分成了一系列更小的步骤，穿插于课程几个章节。

在这个链接上你可以发现大量的示例代码。它是最专业的书之一，它有一个很严肃的原则：“教你计算机科学”。它需要接近 \$40 才能买到，但是你可以下载免费的 PDF 和 HTML 版本。如果从头开始学 Python 的话，我肯定会从它上面得到好处。

## Python at Learnstreet

LearnStreet

Courses

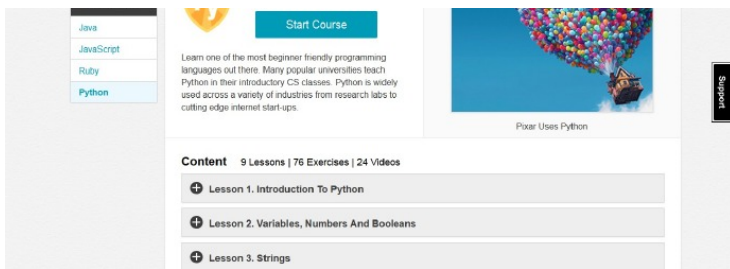
Projects

Teach

Sign In | Sign Up

Courses

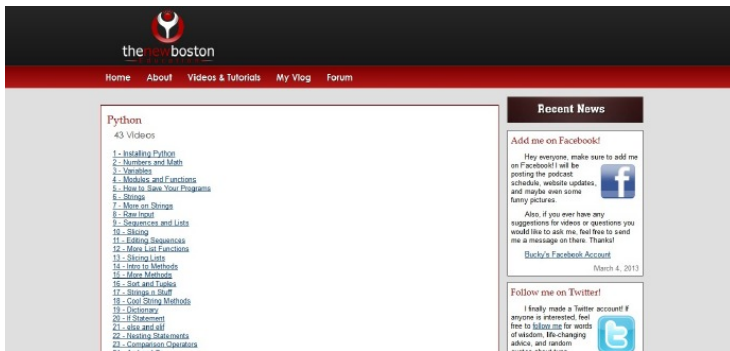
Python



你可能会想一个提供编程教程的网站应该知道如何在他们的页面上添加一个 HTML 标，哈哈。严肃一点地说，Learnstreet 提供了最棒的 Python 交互式学习教程，就像这篇文章所说的所有东西一样，它对初学者很友好。如果你遇到了问题，最好用谷歌搜索他们。

我最喜欢 Learnstreet 的是每一个联系后面的大量提示/解释，就在你写代码的控制台里。

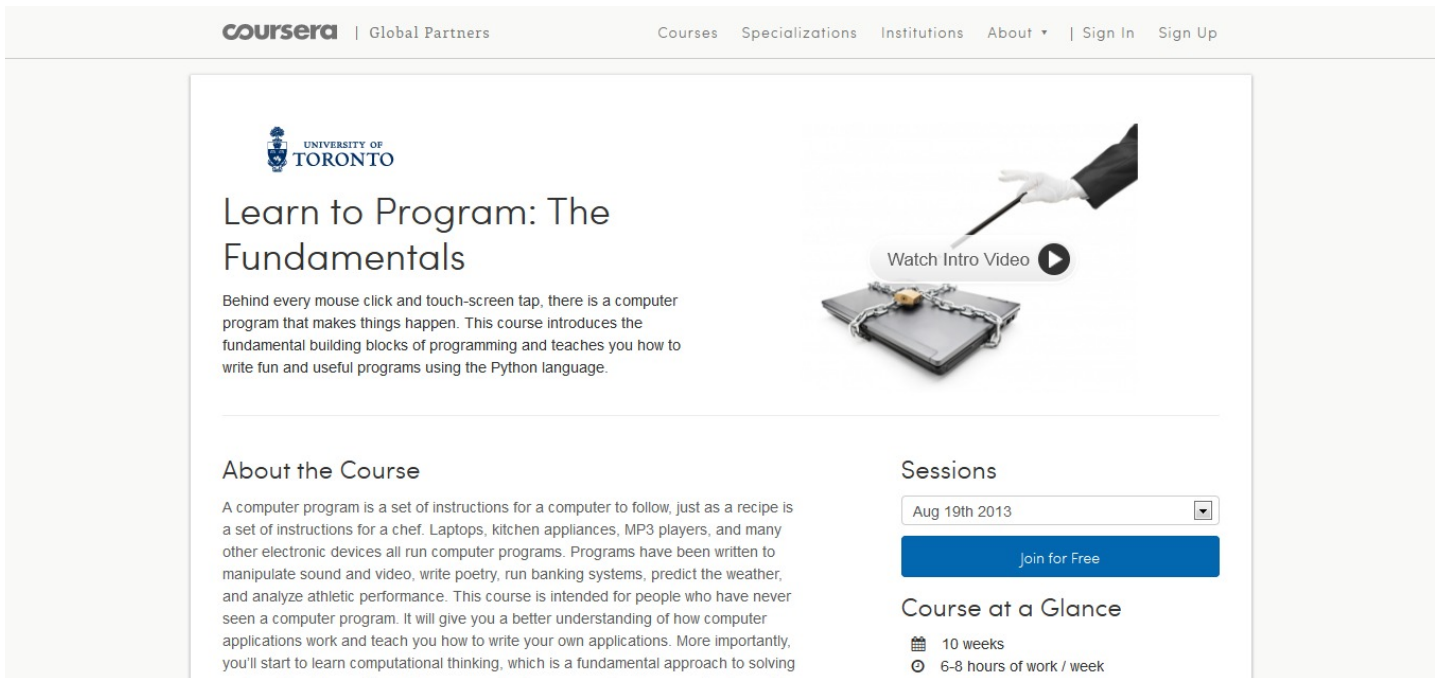
## The New Boston



如果你更喜欢视频学习，我不确定是否还有比 The Nes Boston 的 Python 系列视频教程更全面的了，其他的许多编程语言也可以在他们的 Youtube 频道里找到。

它唯一的缺点就是没有可以看或者下载的实体，你只能看视频。我是那种不能忍受得一直看视频的程序员，这可能与 I 选择 Google 的 Python 课程有冲突，但它们是在太有趣了。

Python @ Coursera



这个课程是为从来没接触过编程的人准备的。你只需要十年级的数学基础就可以了：懂简单的数学 方程和算术优先级。以及像  $f(x) = x + 5$  这样的函数。

它应该在十周内就可以学完，每周大概花10小时来完成课后作业。如果你有时间来做这个，并且不会压垮你自己，我十分推荐你注册并完成这个课程，它只会巩固你的知识，并可以结合上面介绍的 任何资源以更好的理解 Python。

## 在哪里学习 Python ？

结果是我自己尝试过了大部分这些课程，我实际上希望有更多的资源和链接加入到这个列表当中来，但我们仅仅看一下主要的，还有许多你将要学的东西。

你的编程经验是什么？你想用你新发现的技能做什么？我认为任何人想建立自己的经验首先应该知道他们想要建立的东西是什么，然后在那个项目上工作知道做完。它的美妙之处在于你将学到明确的东西，然后很简单就能重新创造相似的东西。

交互式平台很酷，但是他们还不能替代书籍，课程和专家。祝你的 Python 学习顺利，并且如果你有任何想问的问题的话，请在评论里提出。

照片由 [hdwallon](#) 提供