

**PYTHON教程**[1. 简介](#)[2. Python历史](#)[3. 安装Python](#)[3.1. Python解释器](#)[4. 第一个Python程序](#)[5. Python基础](#)[6. 函数](#)[7. 高级特性](#)[8. 函数式编程](#)[9. 模块](#)[10. 面向对象编程](#)[11. 面向对象高级编程](#)[12. 错误、调试和测试](#)[13. IO编程](#)[14. 进程和线程](#)[15. 正则表达式](#)[16. 常用内建模块](#)[17. 常用第三方模块](#)[18. 图形界面](#)[19. 网络编程](#)[20. 电子邮件](#)[21. 访问数据库](#)[22. Web开发](#)[23. 异步IO](#)[24. FAQ](#)[25. 期末总结](#)[下载PDF](#)

# Python解释器



廖雪峰



资深软件开发工程师，业余马拉松选手。

当我们编写Python代码时，我们得到的是一个包含Python代码的以 `.py` 为扩展名的文本文件。要运行代码，就需要 Python解释器去执行 `.py` 文件。

由于整个Python语言从规范到解释器都是开源的，所以理论上，只要水平够高，任何人都可以编写Python解释器来执行Python代码（当然难度很大）。事实上，确实存在多种Python解释器。

## C<sub>Python</sub>

当我们从[Python官方网站](#)下载并安装好Python 3.x后，我们就直接获得了一个官方版本的解释器：C<sub>Python</sub>。这个解释器是用C语言开发的，所以叫C<sub>Python</sub>。在命令行下运行 `python` 就是启动C<sub>Python</sub>解释器。

C<sub>Python</sub>是使用最广的Python解释器。教程的所有代码也都在C<sub>Python</sub>下执行。

## I<sub>Python</sub>

I<sub>Python</sub>是基于C<sub>Python</sub>之上一个交互式解释器，也就是说，I<sub>Python</sub>只是在交互方式上有所增强，但是执行Python代码的功能和C<sub>Python</sub>是完全一样的。

C<sub>Python</sub>用 `>>>` 作为提示符，而I<sub>Python</sub>用 `In [序号]:` 作为提示符。

## PyPy

PyPy是另一个Python解释器，它的目标是执行速度。PyPy采用[JIT技术](#)，对Python代码进行动态编译（注意不是解释），所以可以显著提高Python代码的执行速度。

绝大部分Python代码都可以在PyPy下运行，但是PyPy和C<sub>Python</sub>有一些是不同的，这就导致相同的Python代码在两种解释器下执行可能会有不同的结果。如果你的代码要放到PyPy下执行，就需要了解[PyPy和C<sub>Python</sub>的不同点](#)。

## J<sub>ython</sub>

J<sub>ython</sub>是运行在Java平台上的Python解释器，可以直接把Python代码编译成Java字节码执行。

## I<sub>ronPython</sub>

I<sub>ronPython</sub>和J<sub>ython</sub>类似，只不过I<sub>ronPython</sub>是运行在微软.NET平台上的Python解释器，可以直接把Python代码编译成.NET的字节码。

## 小结

Python的解释器很多，但使用最广泛的还是C<sub>Python</sub>。如果要和Java或.NET平台交互，最好的办法不是用J<sub>ython</sub>或I<sub>ronPython</sub>，而是通过网络调用来交互，确保各程序之间的独立性。

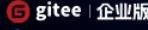
本教程的所有代码只确保在C<sub>Python</sub> 3.x版本下运行。请务必在本地安装C<sub>Python</sub>（也就是从[Python官方网站](#)下载的安装程序）。

[« 安装Python](#)[第一个Python程序 »](#)

PYTHON教程

- 1. 简介
- 2. Python历史
- 3. 安装Python
- 3.1. Python解释器**
- 4. 第一个Python程序
- 5. Python基础
- 6. 函数
- 7. 高级特性
- 8. 函数式编程
- 9. 模块
- 10. 面向对象编程
- 11. 面向对象高级编程
- 12. 错误、调试和测试
- 13. IO编程
- 14. 进程和线程
- 15. 正则表达式
- 16. 常用内建模块
- 17. 常用第三方模块
- 18. 图形界面
- 19. 网络编程
- 20. 电子邮件
- 21. 访问数据库
- 22. Web开发
- 23. 异步IO
- 24. FAQ
- 25. 期末总结

[下载PDF](#)



# 一站式 DevOps 研发效能平台

灵活选择部署方式 | 支持 SaaS 在线使用 | 私有化部署

[进入 Gitee 官网](#)

---

## Comments

Comments loaded. To post a comment, please Sign In

Mooooses @ 2025/12/14 03:00:40  
你们在学习时会做笔记吗? 还是不用

涪 @ 2025/12/31 02:53:36  
会.方便总结回顾

编码A096 @ 2025/11/28 09:13:34  
兄弟们, 当你在官网上下载并安装好python的时候就已经安装好Cpython了

Mooooses @ 2025/12/14 03:02:18  
👉

土土兔 @ 2025/9/23 01:05:39  
32岁小白打卡第一天,

已下载并安装完成Python3.7.7(64-bit)和Pycharm, 查询才了解到Cpython编辑器已包含在Python3.7.7(64-bit)下载安装包中, "Win+R"启动左下角跳出对话框, 输入"CMD"成功打开窗口, 即是安装成功 查询了解到电脑中不同编辑器同时存在也并不冲突

人 @ 2025/10/29 22:15:25  
不 cmd是windows自带本来就有的 进去输入python或是看图形化应用

Edward @ 2025/10/27 08:48:16  
学习打开 2025/10/27

I、阳光 @ 2025/9/10 04:22:19  
Visual Studio Code, 也是一种解释编辑器吗? AI推荐安装的

Bacium @ 2025/10/22 04:33:08  
不是,VScode是代码编辑器,就是个写代码的工具,不是解释器,解释器是把高级编程语言编译成电脑能运行的二进制文件执行.这两不是一个东西.

joker @ 2025/10/20 23:10:00  
天晓得坚不坚持的下去, 总得先开始做吧。 Day1

<https://liaoxuefeng.com/books/python/install/interpreter/index.html>

2/4

PYTHON教程

- 1. 简介
- 2. Python历史
- 3. 安装Python
- 3.1. Python解释器
- 4. 第一个Python程序
- 5. Python基础
- 6. 函数
- 7. 高级特性
- 8. 函数式编程
- 9. 模块
- 10. 面向对象编程
- 11. 面向对象高级编程
- 12. 错误、调试和测试
- 13. IO编程
- 14. 进程和线程
- 15. 正则表达式
- 16. 常用内建模块
- 17. 常用第三方模块
- 18. 图形界面
- 19. 网络编程
- 20. 电子邮件
- 21. 访问数据库
- 22. Web开发
- 23. 异步IO
- 24. FAQ
- 25. 期末总结

下载PDF

Bacium @ 2025/10/22 04:30:47  
弱类型语言还好吧,相对强类型语言简单点

R.. @ 2025/10/7 21:53:13  
打卡

Pilgrim. @ 2025/10/1 10:02:36  
学习打卡-2025-10-01

mh @ 2025/9/4 05:34:37  
所以如何查看使用的是什么解释器呢?  
如何查看解释器的版本号呢?

真的郝漂亮 @ 2025/9/3 20:56:50  
day3

正午阳光 @ 2025/7/29 06:08:45  
第二天打卡。

~ @ 2025/7/2 00:50:19  
day2

南安 @ 2025/7/1 11:44:47  
打卡, 官网安装目录下确实有大量cpython的命名文件

熙.Alice @ 2025/6/25 05:18:52  
win11已经从官网安装了python 但没有CPython, 是公司内网的原因吗?

瓦坎达啃大瓜 🍉 @ 2025/6/3 10:19:39  
打卡

wind & me @ 2025/6/3 04:51:49  
打卡

😊 @ 2025/5/19 21:13:52  
冲啊 打卡第二天

Ada @ 2025/5/13 10:07:37  
打卡

日向夏橘 @ 2025/4/28 02:28:50  
打卡

The screenshot shows a dark-themed web page for the "Python Interpreter" chapter of the "Python Tutorial" by Liao Xuefeng. At the top right, there is a user profile icon with the name "阿求" and the timestamp "2025/4/21 23:08:02". Below the profile, a blue button says "打卡+1". At the bottom right, there is a copyright notice: "©liaoxuefeng.com - 微博 - GitHub - License". On the left side, a sidebar lists various chapters of the tutorial, with "3.1. Python Interpreter" currently selected. A large, empty central area is present, likely a placeholder for the chapter content.

PYTHON教程

- 1. 简介
- 2. Python历史
- 3. 安装Python
- 3.1. Python解释器
- 4. 第一个Python程序
- 5. Python基础
- 6. 函数
- 7. 高级特性
- 8. 函数式编程
- 9. 模块
- 10. 面向对象编程
- 11. 面向对象高级编程
- 12. 错误、调试和测试
- 13. IO编程
- 14. 进程和线程
- 15. 正则表达式
- 16. 常用内建模块
- 17. 常用第三方模块
- 18. 图形界面
- 19. 网络编程
- 20. 电子邮件
- 21. 访问数据库
- 22. Web开发
- 23. 异步IO
- 24. FAQ
- 25. 期末总结

下载PDF