

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

4.1. 使用文本编辑器

4.2. 输入和输出

5. Python基础

5.1. 数据类型和变量

5.2. 字符串和编码

5.3. 使用list和tuple

5.4. 条件判断

5.5. 模式匹配

5.6. 循环

5.7. 使用dict和set

6. 函数

7. 高级特性

7.1. 切片

7.2. 迭代

7.3. 列表生成式

7.4. 生成器

7.5. 迭代器

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基础



廖雪峰



资深软件开发工程师，业余马拉松选手。

Python是一种计算机编程语言。计算机编程语言和我们日常使用的自然语言有所不同，最大的区别就是，自然语言在不同的语境下有不同的理解，而计算机要根据编程语言执行任务，就必须保证编程语言写出的程序决不能有歧义，所以，任何一种编程语言都有自己的一套语法，编译器或者解释器就是负责把符合语法的程序代码转换成CPU能够执行的机器码，然后执行。Python也不例外。

Python的语法比较简单，采用缩进方式，写出来的代码就像下面的样子：

```
# print absolute value of an integer:
a = 100
if a >= 0:
    print(a)
else:
    print(-a)
```



以 `#` 开头的语句是注释，注释是给人看的，可以是任意内容，解释器会忽略掉注释。其他每一行都是一个语句，当语句以冒号 `:` 结尾时，缩进的语句视为代码块。

缩进有利有弊。好处是强迫你写出格式化的代码，但没有规定缩进是几个空格还是Tab。按照约定俗成的惯例，应该始终坚持使用 **4个空格** 的缩进。

缩进的另一个好处是强迫你写出缩进较少的代码，你会倾向于把一段很长的代码拆分成若干函数，从而得到缩进较少的代码。

缩进的坏处就是“复制 - 粘贴”功能失效了，这是最坑爹的地方。当你重构代码时，粘贴过去的代码必须重新检查缩进是否正确。此外，IDE很难像格式化Java代码那样格式化Python代码。

最后，请务必注意，Python程序是 **大小写敏感** 的，如果写错了大小写，程序会报错。

小结

Python使用缩进来组织代码块，请务必遵守约定俗成的习惯，坚持使用4个空格的缩进；

在文本编辑器中，需要设置把Tab自动转换为4个空格，确保不混用Tab和空格。

« 输入和输出

数据类型和变量 »



Comments

Comments loaded. To post a comment, please [Sign In](#)



Qipeng.H @ 2025/6/5 03:55:20

在文本编辑器中，把Tab自动转换为4个空格，就可以直接用Tab代替4个空格了么？具体在哪里设置？

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

4.1. 使用文本编辑器

4.2. 输入和输出

5. Python基础

5.1. 数据类型和变量

5.2. 字符串和编码

5.3. 使用list和tuple

5.4. 条件判断

5.5. 模式匹配

5.6. 循环

5.7. 使用dict和set

6. 函数

7. 高级特性

7.1. 切片

7.2. 迭代

7.3. 列表生成式

7.4. 生成器

7.5. 迭代器

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



Qipeng.H @ 2025/6/5 03:59:05

找到了，左下角齿轮->setting->搜索Tab，好像默认4个空格



Dove @ 2025/6/10 21:46:15

棒！正发愁找不到



JOhN @ 2025/8/10 15:07:04

设置里面：Editor: Tab Size 一个制表符等于的空格数。当 Editor: Detect Indentation 打开时，将根据文件内容替代此设置。



Guo. @ 2025/11/28 21:38:31

默认就是四个~



breeze @ 2025/12/24 02:12:24

感谢分享，我在Macbook上也找半天了



W @ 2025/11/27 04:52:51

`a >= 0`报错，`a > 0`没问题，前后必须都有空格么



电量1% @ 2025/12/18 04:43:59

也不会报错吧，我试过了。干问的解释：

= 前后加空格（即写成 `a = -100` 而不是 `a=-100`）并不是语法强制要求，Python 解释器本身完全接受 `a=-100` 这种写法，程序能正常运行。

但推荐在 = 前后加空格，这是出于代码风格规范和可读性的考虑。



日新 @ 2025/12/18 03:23:16

自然语言有多种解释的可能性，而编程语言是把语言限制在唯一的执行逻辑里，一段代码只有唯一的执行逻辑而没有其他可能性。对于python代码而言，其有着强制缩进的性质，增强代码可读性。



日新 @ 2025/12/18 03:27:14

注意：代码块必须以冒号开始，除非遇到更高级别缩进停止该代码块内容



^→奥霍斯德尔萨拉多 @ 2025/12/13 22:34:34

我在写VBA时为了读取方便也是用的缩进（虽然VBA不是根据缩进来识别代码块的），这习惯还算能适应

试了下，Pycharm在缩进这方面是自动设置的，粘贴到这里似乎也有点变形哈哈（还得是VS code）

```
#注释代码，:后面为代码块，Python采用缩进方式来组织代码块，应遵守约定俗成的习惯，坚持使用4个空格
#设置Tab自动转换为4个空格
#区分大小写
#Print absolute value of an integer
a=100
if a>=0:
    print(a)
else:
    print(-a)
```



砂糖与盐 @ 2025/12/6 03:22:20

评论区学习打卡

📅 今日足迹 | 潜心研习，步履不停

今日驻足评论区，于观点的交锋里萃取思维养分，在见解的碰撞中拓宽认知疆域。每一条独到分享，都是夯实知识体系的基石；每一次深度共鸣，皆是打磨学术视角的砺石。

深耕不辍，方得真知；学思践悟，终有所成。

✅ 打卡状态 | 今日学习任务已完成



西布 @ 2025/12/5 03:28:14

代码块是一组相关的语句，它们作为一个整体执行。在其他语言中，代码块通常用大括号 {} 包裹，但Python没有大括号，而是用缩进来定义代码块。



Rory @ 2025/12/1 20:19:25

一方面是四个空格的缩进==一个tab键；另一方面是python大小写敏感。



小韩 @ 2025/11/6 06:49:21

这个是在文本里面写好再去运行的吗？命令提示里面交互运行智能输入一行代码，好像做不到这个？



风色幻想 @ 2025/11/11 22:22:23

对，4.1不是说了嘛，用文本编辑器



神仙 @ 2025/8/18 14:14:59

print absolute value of an integer:

a = 100 if a >= 0: print(a) else: print(-a) 当我把a = 100改为 a = -1的时候，结果是1；然而我在改成 a = 1的时候，结果还是1.这TMD就尴尬了。2025年8月19日 02:14:56



断罪 @ 2025/8/21 15:48:16

负负为正呀，哥们



Magnolia @ 2025/8/27 04:21:32

这有啥问题，不是对的吗？



/fendou3戒界 @ 2025/9/24 20:57:58

陌拜高手，1+1=2,1-1=0，所以他觉得不对。



贤哥 @ 2025/10/15 05:11:40

你自己看你的标题：“print absolute value of an integer:”就知道原因了——打印一个整数的绝对值，整数的绝对值不是0就是正数。



青水王 @ 2025/10/30 09:43:50

对的哥们，你再仔细想想



. @ 2025/11/4 04:37:19

确实是个神仙

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

4.1. 使用文本编辑器

4.2. 输入和输出

5. Python基础

5.1. 数据类型和变量

5.2. 字符串和编码

5.3. 使用list和tuple

5.4. 条件判断

5.5. 模式匹配

5.6. 循环

5.7. 使用dict和set

6. 函数

7. 高级特性

7.1. 切片

7.2. 迭代

7.3. 列表生成式

7.4. 生成器

7.5. 迭代器

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

4.1. 使用文本编辑器

4.2. 输入和输出

5. Python基础

5.1. 数据类型和变量

5.2. 字符串和编码

5.3. 使用list和tuple

5.4. 条件判断

5.5. 模式匹配

5.6. 循环

5.7. 使用dict和set

6. 函数

7. 高级特性

7.1. 切片

7.2. 迭代

7.3. 列表生成式

7.4. 生成器

7.5. 迭代器

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



Bard @ 2025/11/2 04:10:07

#注释代码：后为代码块 应严格使用Tab进行四空格缩进



炬火 @ 2025/10/31 19:01:09

打卡第一天



R.. @ 2025/10/8 07:37:40

打卡



土土 @ 2025/9/23 06:27:30

打卡第一天记录

"#"为注释，"#"后写的东西给人看的，不参与运行过程和结果

": "后为代码块，大小写字母错误，程序会报错

缩进为4个空格，也可将"Tab"设置成默认4个空格，

VS Code中"设置"—"搜索"中打"tab"—显示已默认"tab"为4个空格



Passion @ 2025/6/3 11:32:37

怎么你们复制上来有格式呢



土土 @ 2025/9/23 06:21:12

评论写字框上方有一个" </> "图标，把要复制过来的代码替换掉"paste code here..."就可以咯



mh @ 2025/9/4 06:12:55

怎么把Tab自动转换为4个空格？



自由度 @ 2025/9/1 09:12:09

第二天打卡



Magnolia @ 2025/8/27 04:45:06

打卡



三七九五一 @ 2025/8/21 20:33:41

Python基础 done



努力努力再努力 @ 2025/8/14 05:22:59

打卡20250814



Ангел @ 2025/8/3 08:54:39

so easy