

5.7. 使用dict和set

6. 函数

6.1. 调用函数

6.2. 定义函数

6.3. 函数的参数

6.4. 递归函数

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

8.1. 高阶函数

8.1.1. map/reduce

8.1.2. filter

8.1.3. sorted

8.2. 返回函数

8.3. 匿名函数

8.4. 装饰器

8.5. 偏函数

9. 模块

9.1. 使用模块

9.2. 安装第三方模块

10. 面向对象编程

10.1. 类和实例

10.2. 访问限制

10.3. 继承和多态

10.4. 获取对象信息

10.5. 实例属性和类属性

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

面向对象编程



廖雪峰

资深软件开发工程师，业余马拉松选手。

面向对象编程——Object Oriented Programming，简称OOP，是一种程序设计思想。OOP把对象作为程序的基本单元，一个对象包含了数据和操作数据的函数。

面向过程的程序设计把计算机程序视为一系列的命令集合，即一组函数的顺序执行。为了简化程序设计，面向过程把函数继续切分为子函数，即把大块函数通过切割成小块函数来降低系统的复杂度。

而面向对象的程序设计把计算机程序视为一组对象的集合，而每个对象都可以接收其他对象发过来的消息，并处理这些消息，计算机程序的执行就是一系列消息在各个对象之间传递。

在Python中，所有数据类型都可以视为对象，当然也可以自定义对象。自定义的对象数据类型就是面向对象中的类（Class）的概念。

我们以一个例子来说明面向过程和面向对象在程序流程上的不同之处。

假设我们要处理学生的成绩表，为了表示一个学生的成绩，面向过程的程序可以用一个dict表示：

```
std1 = { 'name': 'Michael', 'score': 98 }
std2 = { 'name': 'Bob', 'score': 81 }
```

而处理学生成绩可以通过函数实现，比如打印学生的成绩：

```
def print_score(std):
    print('%s: %s' % (std['name'], std['score']))
```

如果采用面向对象的程序设计思想，我们首选思考的不是程序的执行流程，而是 `Student` 这种数据类型应该被视为一个对象，这个对象拥有 `name` 和 `score` 这两个属性（Property）。如果要打印一个学生的成绩，首先必须创建出这个学生对应的对象，然后，给对象发一个 `print_score` 消息，让对象自己把自己的数据打印出来。

```
class Student(object):
    def __init__(self, name, score):
        self.name = name
        self.score = score

    def print_score(self):
        print('%s: %s' % (self.name, self.score))
```

给对象发消息实际上就是调用对象对应的关联函数，我们称之为对象的方法（Method）。面向对象的程序写出来就像这样：

```
bart = Student('Bart Simpson', 59)
lisa = Student('Lisa Simpson', 87)
bart.print_score()
lisa.print_score()
```

面向对象的设计思想是从自然界中来的，因为在自然界中，类（Class）和实例（Instance）的概念是很自然的。Class是一种抽象概念，比如我们定义的Class——Student，是指学生这个概念，而实例（Instance）则是一个个具体的Student，比如，Bart Simpson和Lisa Simpson是两个具体的Student。

所以，面向对象的设计思想是抽象出Class，根据Class创建Instance。

面向对象的抽象程度又比函数要高，因为一个Class既包含数据，又包含操作数据的方法。

小结

数据封装、继承和多态是面向对象的三大特点，我们后面会详细讲解。

《 安装第三方模块

类和实例 >>

5.7. 使用dict和set

6. 函数

6.1. 调用函数

6.2. 定义函数

6.3. 函数的参数

6.4. 递归函数

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

8.1. 高阶函数

8.1.1. map/reduce

8.1.2. filter

8.1.3. sorted

8.2. 返回函数

8.3. 匿名函数

8.4. 装饰器

8.5. 偏函数

9. 模块

9.1. 使用模块

9.2. 安装第三方模块

10. 面向对象编程

10.1. 类和实例

10.2. 访问限制

10.3. 继承和多态

10.4. 获取对象信息

10.5. 实例属性和类属性

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

 gitee | 企业版

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](#)



Rory @ 2025/12/20 02:16:31

start learning object



火凤凰 @ 2025/12/7 21:05:20

java 老程序员 报道



Mr_RightMen @ 2018/1/10 02:02:09

上帝

class 类 (人) instance 实例 (你,我,他) 你会有些属性(身高,年龄,体重) 你会有些技能(吃饭,泡妞)

`__init__` 方法的主要作用,就是初始化你的属性,这些属性,在上帝初始化你的时候就要赋予给你,比如
`zhangsan = Person(170,29,50)` 这时上帝就把你创造出来了,也就是实例化了你,然后,你到底有哪些技能呢,这就看有没有在类里面定义了,如果有定义泡妞的技能,那么你就可以调用泡妞的技能来泡妞,大致就是这样吧,看看下面的例子就更清楚了

```
class Person(object):  
    # 这里就是初始化你将要创建的实例的属性  
    def __init__(self,height,weight,age):  
        self.height = height  
        self.weight = weight  
        self.age = age
```

[Read More](#) ▼



ifarewell @ 2018/1/19 00:04:38

就服你



罗田心Livia @ 2018/1/31 20:15:48

我看你的看懂了 谢谢~~~



格林匹思 @ 2018/2/5 03:55:37

666



莉莉娅的小暖炉 @ 2018/2/5 08:06:38

陈独秀同学, 请坐下来。



似水年华_ @ 2018/3/19 02:54:57

你比作者讲的清楚明白多了

5.7. 使用dict和set

6. 函数

6.1. 调用函数

6.2. 定义函数

6.3. 函数的参数

6.4. 递归函数

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

8.1. 高阶函数

8.1.1. map/reduce

8.1.2. filter

8.1.3. sorted

8.2. 返回函数

8.3. 匿名函数

8.4. 装饰器

8.5. 偏函数

9. 模块

9.1. 使用模块

9.2. 安装第三方模块

10. 面向对象编程

10.1. 类和实例

10.2. 访问限制

10.3. 继承和多态

10.4. 获取对象信息

10.5. 实例属性和类属性

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



雇佣兵911201705 @ 2018/3/21 00:02:11

李时珍啊



Destiny_Raymond @ 2018/3/25 05:50:57

李时珍的皮



凌久殇 @ 2018/3/26 08:16:57

讲的不错，大写的服



BlingBling33Stream @ 2018/3/29 05:09:42

解释得太好了！



快没时间啦 @ 2018/3/30 09:53:26

要不是陈老师您的精彩讲解,我是真滴看不懂..



TTTTV1993 @ 2018/4/15 06:12:09

厉害



简晓单F @ 2018/4/24 07:08:02

简单粗暴



W-Metro @ 2018/4/26 09:19:32

秦始皇规划建陵之时，李斯提议以陶俑陪葬，赵高提议以铁俑陪葬，二人争执不下，遂造陶铁二俑各百，置于大窖之中。谁料铁俑虽坚，却不耐阴湿，几日后，陶俑栩栩如生，铜俑却已然生锈，赵高心中焦急，却无能为力，眼见铁俑面目斑驳，渐渐朽烂，赵高绝望，伏地哭求：求求你们！求求你别锈了



源不胜 @ 2018/5/14 03:46:15

这么讲真的很易于理解。 btw，我们在编码的时候也相当于上帝了



鹏碰2013 @ 2018/5/16 03:28:46

厉害



BazingaLife @ 2018/5/16 05:24:36

顶上去~！！！！



风致 @ 2018/5/21 04:58:53

感谢成老师和博主



老爷们儿frankie @ 2018/5/23 04:58:29

大家都是九年义务教育，为什么你这么优秀？



Hello_景尊 @ 2018/5/24 11:15:18

前列腺你别总发言了，给扁桃体点机会

5.7. 使用dict和set

6. 函数

6.1. 调用函数

6.2. 定义函数

6.3. 函数的参数

6.4. 递归函数

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

8.1. 高阶函数

8.1.1. map/reduce

8.1.2. filter

8.1.3. sorted

8.2. 返回函数

8.3. 匿名函数

8.4. 装饰器

8.5. 偏函数

9. 模块

9.1. 使用模块

9.2. 安装第三方模块

10. 面向对象编程

10.1. 类和实例

10.2. 访问限制

10.3. 继承和多态

10.4. 获取对象信息

10.5. 实例属性和类属性

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

破碎 @ 2025/4/6 23:30:46

恩师

zoe @ 2025/3/1 13:02:41

留下点印记,,于我而言这是一套系统的教程, 半路出家, 工作一年了感觉基础还是很薄弱, 准备跟着廖神开始一步一个脚印, 先每天3小时学3个月看下能学多少

myEsn2E9 @ 2025/2/26 04:39:27

打卡

柠子Yo @ 2025/2/14 20:53:36

打卡Day8

多吃碳水 @ 2025/1/6 03:51:53

打卡 今天看了 10、19、21

菜问 @ 2024/3/27 02:24:57

类就是结构体的变种, 一个结构体声明后需要定义变量才能使用, 这件事放到面向对象语言中, 就是类的“实例化”过程, 而被定义的变量就叫“对象”

江有汜 @ 2024/11/15 00:52:18

言简意赅

jx @ 2024/11/12 01:19:39

打卡第五天

白衣未央ox @ 2024/6/25 04:54:33

掌握了Java再来学Python, 理解起来非常容易

鲸鱼 @ 2024/10/19 22:44:23

懂了, 先学JAVA

y @ 2024/9/30 07:23:33

面向过程就是相当于编写一行一行命令; 面向对象, 首先考虑到对象, 也就是说这个对象有哪些属性和方法, 以此设计一个类模板, 并基于这个模板创造对象实例

用户6610765864 @ 2021/4/26 05:13:42

笑死, 根本找不到对象: (

伊颿 @ 2021/5/5 23:34:42

笑死, 根本笑不死

殇Bscorpio @ 2021/7/6 10:46:32

你搁这搁这呢啊

5.7. 使用dict和set

6. 函数

6.1. 调用函数

6.2. 定义函数

6.3. 函数的参数

6.4. 递归函数

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

8.1. 高阶函数

8.1.1. map/reduce

8.1.2. filter

8.1.3. sorted

8.2. 返回函数

8.3. 匿名函数

8.4. 装饰器

8.5. 偏函数

9. 模块

9.1. 使用模块

9.2. 安装第三方模块

10. 面向对象编程

10.1. 类和实例

10.2. 访问限制

10.3. 继承和多态

10.4. 获取对象信息

10.5. 实例属性和类属性

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




On1on @ 2023/7/17 02:57:39

不行换个类



0 @ 2023/7/17 03:04:23

用java new一个就好了



镜子的你_ @ 2023/9/18 05:37:09

学JS给自己new一个也行

©liaoxuefeng.com - 微博 - GitHub - License