

阅读: 145933

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。



感觉本站内容不错，读后有收获？

¥ 我要小额赞助，鼓励作者写出更好的教程

还可以分享给朋友

分享

C-Cheris... , Confiden... 等4人分享过



珠峰NODE.JS全栈开发

Angular

React

Gulp

Meteor

Vue

微信开发

技术陪伴成长社区

珠峰培训

掘金

一个只有高手分享的技术社区

立即加入

麦子学院

百万级python导师亲身指导

保你120天变身python大牛

有时候，你需要的只是一句点拨

立即咨询

深度学习在线课程

通向无人驾驶的必经之路



课后理解，应该正确

天恩HoChac created at 11-25 16:26, Last updated at 11-25 16:26

Class定义的Students 指的最“学生这一类人”，应该会带有名字和测试成绩两个属性。
Instance 定义的 Students，就是指“第二小学三年级六班的小明同学”他的名字就是小明，数学成绩96分。

是这样理解吗？

☰ View Full Discuss

↩ Reply This Topic



老师请问对象和实例的区别

手机用户3002867223 created at 2015-12-8 10:24, Last updated at 11-20 21:09

以我目前的理解，对象和实例似乎是一个东西



倔强的leter

Created at 2015-12-16 21:35, Last updated at 2015-12-16 21:35

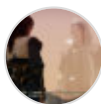
把类想象成为AV女优，一个抽象的职业，把实例想象成为苍老师，一个活生生的人，当你想lu的时候脑海会自动创建一个名字为‘苍井空’的AV女优实例，而不可能只想着AV女优这四个字来lu。这就是实例化对象的过程。



雪拉比小木屋

Created at 2015-12-16 22:15, Last updated at 2015-12-16 22:15

对象只是抽象出来的一个概念，实际上不占用内存。但是对象是占用内存的，是一个真实存在的东西。“鱼”是一个类，这个世界上没有“鱼”这个东西，但是有“草鱼”，“鲤鱼”“江团鱼”等等，它们是“鱼”的实例化，即对象。



神秘的龙吐枫

Created at 1-15 16:59, Last updated at 1-15 16:59

AV女优这个比方让我豁然开朗呐



奇葩的师弟

Created at 4-11 10:58, Last updated at 4-11 10:58

这个解释让我豁然开朗



墨岩74032

Created at 4-12 9:28, Last updated at 4-12 9:28

这个其实在平时经常将实例也说成对象，大家也能理解。对象你通过构造函数构建出一个实例，不过，为什么要举AV女优这职业，任何行当不都是这意思么。。。





后面那个男同学

Created at 4-22 9:18, Last updated at 4-22 9:18

我笑了。举例子太逗了。



gylpni

Created at 4-29 10:29, Last updated at 4-29 10:29

****av**女优的例子真的令人豁然开朗



gylpni

Created at 4-29 10:29, Last updated at 4-29 10:29

av女优的例子真的令人豁然开朗



爱梦男孩

Created at 5-20 0:10, Last updated at 5-20 0:10


果然AV令人豁然开朗啊~~



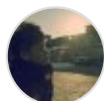
吃pizza要加王致和

Created at 5-20 10:28, Last updated at 5-20 10:28

这例子果然豁然开朗，以后面试官问我我也这么答了

 View Full Discuss

 Reply This Topic



python3的类class(object)括号中的object可以不用写吧？

Fleeting-Jiang created at 10-9 14:22, Last updated at 10-26 20:38

python3的类class(object)括号中的object可以不用写吧？




郭畅璐

Created at 10-26 20:38, Last updated at 10-26 20:38

可以不写(object)。 因为object是所有类的父类，会创建instance时，这个类会自动加载。



 View Full Discuss

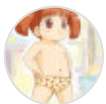
 Reply This Topic



那么 __init__ 就类似构造了？ self就像是this一样了？

oO我是耀栋Oo created at 9-27 21:51, Last updated at 9-28 9:13

和java对比,是不是这个意思呐



匿名sina网友V

Created at 9-28 9:13, Last updated at 9-28 9:13

差不多，java的this是隐式的，它就叫this。

python的self是显式的，以形参的身份出现，你写成this，itself也没问题。

View Full Discuss

Reply This Topic



__init__ 然后还有self都有点难理解....不知道下面的理解对不对

笑劫戈和囡囡眉 created at 9-16 16:24, Last updated at 9-16 16:24

上网上查了一下：

原答案：

<http://stackoverflow.com/questions/625083/python-init-and-self-what-do-they-do>

<http://stackoverflow.com/questions/8609153/why-do-we-use-init-in-python-classes>

init称为构造函数或者初始程序，初始化类或对象的实例，在这个_**init**_下，新创立的对象就是**self**。

class Dog:

def **init**(self, legs, colour):

self.legs = legs

self.colour = colour

fido = Dog(4, "brown")

spot = Dog(3, "mostly yellow")

对一只狗来说：_**init**_就是第一次写了一些信息的出生证明，几斤重，在哪里出生等等

变量**self**就是它以后会成为的那条狗。而**self.color** 或者**self.legs**或**self.xx**就是它今后的改变会成为怎样一只狗，黄色 3条腿或其他

然后有个例子关于**Class**中的变量和 **init** 函数中的变量的差别

[Read More](#)

View Full Discuss

Reply This Topic



关于_**init**_的理解

机修贾森 created at 8-23 23:33, Last updated at 9-6 14:31

在看到这一章之前，Student类我一般习惯这么写：



```
>>> class Student:
    def init(self, name, score):
        self.name = name
        self.score= score
    def print_score(self):
        print(' %s: %s' % (self.name, self.score))
```



实例化和调用：

```
>>> std1 = Student()
>>> std1.init('Ian', 96)
>>> std1.print_score()
Ian:96
```

再对比老师的代码:

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

▼ [Read More](#)



[机修贾森](#)

Created at 8-23 23:35, Last updated at 8-23 23:35

第一个是init, 第二个是(下划线下划线)init(下划线下划线)...



[000JC](#)

Created at 9-3 9:56, Last updated at 9-3 9:56

`__init__`



[ksven](#)

Created at 9-6 14:31, Last updated at 9-6 14:31

init就是一个普通的赋值方法, 没实例化student之前还不能调用(那当然), 而且实例化之后可以多次调用。“用来实例化对象std1”这句话不合适, 这个普通方法的作用只是赋值, 不能叫做实例化。

init 是实例化方法(构造方法), 在实例化student的时候自动调用, 之后就不再调用(对这个已经实例化的对象来说)。

前者是你买了个毛坯房, 然后你再自己装修, 甚至装修的不满意还能推到重来
后者是你直接买了个装修好的房子。你能再装修, 但是房子已经不用再买一次了

≡ [View Full Discuss](#)

↩ [Reply This Topic](#)



[关于self的一点思考](#)

[早起的达先生](#) created at 8-5 10:59, Last updated at 8-5 10:59

感觉self和实例的名称是互通的, 只不过self负责在类里面的定义, 实例负责类外

➤ ≡ [View Full Discuss](#)

↩ [Reply This Topic](#)



[为什么显示object takes no parameter](#)

[邱宇曜曜](#) created at 6-10 11:21, Last updated at 8-1 0:25

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))
```

然后运行

bart=Student('nano', 99)结果显示object() takes no parameter是怎么回事儿啊



yicustb

Created at 6-10 22:45, Last updated at 6-10 22:45

init函数的名字是两个下划线



碉你妹的堡

Created at 8-1 0:25, Last updated at 8-1 0:25

楼上正解！！ `_int_` 也卡了我半天。。。

View Full Discuss

Reply This Topic



廖哥.这个None是怎么回事呢

叫我玮仔 created at 4-4 3:45, Last updated at 5-31 13:22

代码是完全按照教程里的写的,诧异的是在输出成绩的同时还换行输出了一个None,请教是为什么

```
sd1={'name':'ShaoWei','score':100}
sd2={'name':'ZuoBing','score':100}

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

print(print_score(sd1))
print(print_score(sd2))
```

```
ShaoWei: 100
None
ZuoBing: 100
None
[Finished in 0.1s]
```



廖雪峰

Created at 4-5 9:28, Last updated at 4-5 9:28

把语句

```
print(print_score(sd1))
```

拆开来。



另外有：

```
r = print_score(sdl)
print(r) # r是啥？
```



Aesthetic_KSY

Created at 4-11 14:25, Last updated at 4-11 14:25

```
def lazy_sum(*args):
    def sum():
        ax = 0
        for n in args:
            ax = ax + n
        return ax
    return sum

f = lazy_sum(1, 3, 5, 7, 9)
print f()
```

请问为什么这段代码需要`print f()`?



黑岩74032

Created at 4-12 9:25, Last updated at 4-12 9:25

楼上，`lazy_sum()`返回的是`sum()`函数，这时`sum()`并没执行，所以，`f()`才是具体的执行



MC的暖暖阳光

Created at 5-31 13:22, Last updated at 5-31 13:22

输出的一三行 是你直接打印了`std1`，`std2`； 输出`None` 是因为你的函数没有返回值

 View Full Discuss

 Reply This Topic



`init` 就是C++中的构造函数，`self`就是指向对象首地址的`this`指针

巴拉巴拉巴拉哈哈 created at 5-10 10:00, Last updated at 5-12 9:21

`init`就是C++中的构造函数，`self`就是指向对象首地址的`this`指针



廖雪峰

Created at 5-12 9:21, Last updated at 5-12 9:21



可以这么理解



 View Full Discuss

 Reply This Topic

发表评论

Sign In to Make a Comment

廖雪峰的官方网站©2015
Powered by [iTranswarp.js](#)
由[阿里云](#)托管
[广告合作](#)



友情链接: [中华诗词](#) - [阿里云](#) - [SICP](#) - [4clojure](#)

