

## 文章标题

阿左<sup>1</sup>      Nobody<sup>2</sup>

May 17, 2012

<sup>1</sup>感谢读者

<sup>2</sup>感谢国家

# Contents

<b>I</b>	<b>语法基础</b>	<b>4</b>
<b>1</b>	<b>OOP 与类</b>	<b>5</b>
1.1	Python 基本类概念 . . . . .	5
1.1.1	类的继承 . . . . .	5
1.1.2	类的属性与方法 . . . . .	6

# **List of Figures**

# **List of Tables**

# Part I

## 语法基础

# Chapter 1

## OOP 与类

在 python 中，同一个模块只能有一个实例，当一个模块的代码被修改以后必须重新加载模块才能生效；而是 python 中的一个类可以同时创建多个实例（即对象）。

python 中把数据保存在对象中，相关的操作（方法）在类中。

### 1.1 Python 基本类概念

#### 1.1.1 类的继承

```
1 # coding=utf-8
2
3 class c1():
4     def show(self):
5         print 'c1.show()'
6     def helloC1(self):
7         print 'c1.helloC1()'
8
9 class c2():
10    def show(self):
11        print 'c2.show()'
12    def helloC2(self):
13        print 'c2.helloC2()'
```

```
14
15 class c3(c1, c2):           # 支持多重继承python
16     def show(self):
17         print 'c3.show()'
18
19 i1 = c3()
20 i1.show()                   # 调用重写的方法
21 i1.helloC1()                # 调用到父类的方法
22 i1.helloC2()                # 调用到父类的方法
```

Listing 1.1: 类继承

### 1.1.2 类的属性与方法

对于类和对象，无论属性还是方法。python 都作为变量处理。

```
1 # coding=utf-8
2
3 class Employee():
4     def __init__(self, name): # __init__ 声明构造函数
5         self.name = name
6     def showName(self):
7         print self.name
8
9 morgan = Employee("Morgan")
10 morgan.showName()
11 print morgan.showName
12
13 morgan.showName = "new name" # 可以把变量赋值给原来是方法的成员
14 print morgan.showName
```

Listing 1.2: 类的属性与方法