## 面向对象编程

### 13.1 引言

类与实例

类是对象得定义，而实例是真正的实物。

### 13.2 面向对象编程

### 13.3 类

### 13.4 类属性

>>> class A:

foo = 100

def \_\_init\_\_(self):

self.name = 'A'

def myfunc(self):

pass

>>> A.\_\_dict\_\_

{'\_\_module\_\_': '\_\_main\_\_', 'foo': 100, 'myfunc': <function myfunc at 0x0000000003067B38>, '\_\_init\_\_': <function \_\_init\_\_ at 0x0000000003067978>, '\_\_doc\_\_': None}

>>>

>>>

>>> a = A()

>>>

>>> a.\_\_dict\_\_

{'name': 'A'}

>>> dir(A)

['\_\_doc\_\_', '\_\_init\_\_', '\_\_module\_\_', 'foo', 'myfunc']

>>> dir(a)

['\_\_doc\_\_', '\_\_init\_\_', '\_\_module\_\_', 'foo', 'myfunc', 'name']

>>> vars(A)

{'\_\_module\_\_': '\_\_main\_\_', 'foo': 100, 'myfunc': <function myfunc at 0x0000000003067B38>, '\_\_init\_\_': <function \_\_init\_\_ at 0x0000000003067978>, '\_\_doc\_\_': None}

|  |  |
| --- | --- |
| 特殊类属性 | 用途 |
| C.\_\_name\_\_ | 类C的名字 |
| C.\_\_doc\_\_ | 类C的文档属性 |
| C.\_\_bases\_\_ | 类C的所有父类构成的元组 |
| C.\_\_dict\_\_ | 类C的属性字典 |
| C.\_\_moudule\_\_ | 类C定义所在模块 |
| C.\_\_class\_\_ | 实例C对应的类（仅新式类） |
| C.\_\_mro\_\_ | 类继承的查找顺序元组（新式类） |

### 13.5 实例

### 13.6 实例属性

### 13.7 绑定方法和调用

### 13.8 静态方法和类方法

class **TestStaticMethod**:

def **foo**():

print *"calling static method foo()."*

foo = staticmethod(foo)

class **TestClassMethod**:

def **foo**(cls):

print *"calling class method foo()."*

print *"foo is part of class %s."* % cls.\_\_name\_\_

foo = classmethod(foo)

if \_\_name\_\_ == *"\_\_main\_\_"*:

tsm = TestStaticMethod()

tsm.foo()

TestStaticMethod.foo()

tcm = TestClassMethod()

tcm.foo()

TestClassMethod.foo()

### 13.9 组合

### 13.10 子类和派生

### 13.11 继承