

rz面向对象编程 (OOP)

程序 = 指令 + 数据

代码可以选择以指令为核心或以数据为核心进行编写

两种范式

以指令为核心：围绕“正在发生什么”进行编写

面向过程编程：程序具有一系列线性步骤;主题思想是代码作用于数据

以数据为核心：围绕“将影响谁”进行编写

面向对象编程 (oop)：围绕数据及为数据严格定义的接口来组织程序，用数据控制对代码的访问

所有东西都是对象

程序是一大堆对象的组合

通过消息传递，各对象知道自己该做什么

消息：即调用请求，它调用的是从属于目标对象的一个方法

每个对象都有自己的存储空间，并可容纳其它对象

通过封装现有对象，可以制作成新型对象

每个对象都属于某一类型

类型，也即类

对象是类的实例

类的一个重要特性为“能发什么样的消息给它”

同一个类的所有对象都能接受相同的消息

对象的接口：类的方法

类：将同一种具体物事的共同特性抽象出来的表象

状态和转换这些状态的操作

数据：状态

变量：属性

方法：操作

函数：操作变量引用的数据代码

类间关系：

依赖：一个类的方法操另一个类的对象

聚合：类A的对象包含类B的对象

继承：描述特殊与一般关系

面向对象模型机制3个原则：封装、继承、多态

封装 (Encapsulate)：隐藏实现方案细节

将代码及其处理的数据绑定在一起的一种编程机制，用于保证编程程序不受外部干扰且不会被误用

继承 (Inheritance)

一个对象获得另一个对象属性的过程；用于实现按层分类的概念

一个深度继承的子类继承了类层次中的它的每个祖先的所有属性

超类、基类、父类

子类、派生类

多态 (Polymorphism)

允许一个接口被多个通用的类动作使用的特性，具体使用哪个动作与应用场合有

关

“一个接口，多个方法”

用于为一组相关的动作设计一个通用的接口，以降低程序的复杂性

def and(a,b):

return a+b

```
and(1,2)
and('a','b')
and(['a','b','c'],[4,5,6])
```

类：是一种抽象概念。

python 类、实例

类是一种数据结构，可用于创建实例

一般情况下，类封装了数据和可用于该数据的方法

python类是一个可调用对象，即类对象

类实例化事可以使用\_\_init\_\_、\_\_del\_\_ 两个特殊方法

类体可以包含：声明语句、类成员定义、数据属性、方法

python 类定义、实例化

```
class ClassName(bases):
    data = value
    def method(self,...):
        self.member = value
```

```
instance = ClassName
```

```
class FirstClass():
    data = "hello class"
```

```
def printdata(self):  
    print self.data
```

```
ins1 = FirstClass()
```

```
ins1.printdata()
```

```
class SecClass():  
    data = "hello SecClass"  
    def printdata(self):  
        print "Content from method: %s" % self.data
```

```
ins2 = SecClass()
```

```
ins2.printdata()
```

可调用的属性:方法

```
object.method()
```

## python 构造器

创建实例是，python会自动调用类中的\_\_init\_\_方法，以隐形地为实例提供属性

\_\_init\_\_ 方法称为构造器

如果类中没有定义\_\_init\_\_方法实例创建之初仅是一个简单的名称空间

```
class MyClass():  
    gender = "Male"  
    def __init__(self,who):  
        self.name = who
```

```
x = MyClass('Tom')
```

```
x.gender , x.name
```

```
y = MyClass('jaml')
```

```
y.gender , y.name
```

类的特殊属性：

可以使用dir()、\_\_dict\_\_ 获取类属性

```
MyClass.__dict__
```

```
dir(MyClass)
```

方法可以变量

实例变量：self.变量名

局部变量：方法内部创建的变量，可直接使用

类变量：类名.变量名

全局变量：直接使用