



Python语言基础与应用

高级特性 / 面向对象：什么是对象

陈斌 北京大学 gischen@pku.edu.cn

面向对象：什么是对象

- › 对象的基本概念
- › 创建对象
- › 对象属性和方法的引用
- › 面向对象编程

对象的基本概念

> “万物皆对象”

Python中的所有事物都是以对象形式存在，
从简单的数值类型，到复杂的代码模块，都
是对象



对象的基本概念

› 对象(object)

既表示客观世界问题空间中的某个具体事物，
又表示软件系统解空间中的基本元素

› 对象 = 属性 + 方法

对象以id作为标识，既包含数据(属性)，也
包含代码(方法)，是某一类具体事物的特殊
实例

对象的基本概念

```
>>> id(1)
4297537952
>>> type(1)
<class 'int'>
>>> dir(1)
['__abs__', '__add__', '__ad
['__rshift__', '__divmod__',
['__floordiv__', '__hash__',
['__lt__', '__mod__', '__mul__',
['__lshift__', '__pos__', '__
['__reduce_ex__', '__repr__',
['__rtruediv__', '__setattr__',
['__subclasshook__', '__geta
ugate', 'denomina
>>> id('a')
4300773280
>>> type('a')
<class 'str'>
>>> dir('a')
['__add__', '__class__', '__contains__', '__delattr__', '__
['__eq__', '__format__', '__ge__', '__getattribute__', '__
['__gt__', '__hash__', '__init__', '__iter__', '__le__', '
['__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__
['__rmod__', '__rmul__', '__setattr__', '__sizeof__', '__
['__str__', '__subclasshook__', 'capitalize', 'casefold', 'ce
expandtabs', 'find', 'format', 'format_map', 'index', 'is
ecimal', 'isdigit', 'isidentifier', 'islower', 'isupper',
ace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'p
rtition', 'replace', 'rfind', 'rindex', 'rstrip', 'split',
', 'split', 'splitlines', 'startswith', 'strip', 'title',
', 'upper', 'zfill']

>>> abs(-1)
1
>>> id(abs)
4298931872
>>> type(abs)
<class 'builtin_function_or_method'>
>>> dir(abs)
['__call__', '__class__', '__delattr__', '__dict__', '__
['__get__', '__getattribute__', '__gt__', '__hash__', '__
['__lt__', '__module__', '__name__', '__ne__', '__new__',
uce__', '__reduce_ex__', '__repr__', '__self__', '__seta
__str__', '__subclasshook__', '__text_signature__']
```

创建对象

› 对象是类的实例，是程序的基本单元

要创建一个新的对象，首先必须定义一个类，用以指明该类型的对象所包含的内容(属性和方法)

同一类(class)的对象具有相同的属性和方法，但属性值和id不同

创建对象

› 对象的名称

赋值语句给予对象以名称，对象可以有多个名称（变量引用），但只有一个id

例：a = complex(1, 2)

› 对象实现了属性和方法的封装，是一种数据抽象机制

提高了软件的重用性、灵活性、扩展性

对象属性和方法的引用

› 引用形式

＜对象名＞.＜属性名＞

可以跟一般的变量一样用在赋值语句和表达式中

例：“abc”.upper()

(1+2j).real

(1+2j).imag

对象属性和方法的引用

- › Python语言动态的特征，使得对象可以随时**增加**或者**删除**属性或者方法

```
44 print(f3.fx, f3.fy)
45 f3.fz = 3.4
46 print(f3.fz)
47 del f3.fz
```

```
0.0 4.5
3.4
```

面向对象编程

› 面向对象编程(OOP)是一种程序设计范型，同时也是一种程序开发方法

程序中包含各种独立而又能互相调用的对象

每个对象都能接受、处理数据并将数据传递给其他对象

› 传统程序设计

将程序看作一系列函数或指令的集合