

# Lesson6

## Monkey by 51reboot

# 大纲

- 回顾
- 面向对象概念
- 面向对象语法
- 面向对象示例
- Q&A

# 面向对象概念

- 面向对象是什么
- 面向对象的基本思想
- 类是什么
- 类的特性

# 面向对象是什么

1. 面向对象是程序设计的一种范式
2. 面向对象对现实世界建立的对象模型

4

# 面向对象的基本思想

- 类 和 对象

类 是定义数据的抽象类型

对象 是类的实例化的产物

通俗的说:

人 就是一个类

程序员 就是类的一个对象

# 类是什么

属性 和 功能 组成一个类

6

# 类的特性

- 封装
- 继承
- 多态

# 封装

属性

性别/年龄/身高/.....

功能/封装

学python/上班挣钱/喝酒撸串/.....



# 继承

- 单继承
- 多继承
- 继承模型

程序员（父类）

前端程序员

后端程序员

python程序员

足球迷

球迷python程序员

go程序员

nodejs程序员

# 多态

一个类继承了多个不同的类，调用同一方法会有不同的反应

10

# 定义

- 语法

```
class ClassName(Base1, Base2, Base3): # default: object
    statement...
```

- 创建类的对象

```
# 一般不需要人为干预，由父类实现。(返回一个类的起始对象)
def __new__(cls):
    pass
```

- 构造函数

```
# 定义类的属性，初始化对象
def __init__(self, [...]):
    pass
```

- 析构函数

```
# 销毁一个对象时调用的，在被python的垃圾回收时调用的。
def __del__(self, [...]):
    pass
```

# 属性的访问控制

1. Public/Private(Go)
2. Python其实没有访问控制，没有提供私有属性的功能。
3. 完全依靠Python规范和命名规则进行约束。

12

# 定义类的方法

函数和方法的区别？

- 函数 可以直接调用函数名
- 方法 必须和对象结合起来使用， 方法是类的一部分

**@classmethod**

调用的时候用类名，而不是某个对象

**@property**

像访问属性一样调用方法

# 内建函数

<code>dir</code>	获取对象的方法
<code>type</code>	获取对象的类型
<code>isinstance</code>	判断是否为父类
<code>issubclass</code>	判断是否为子类

# 类和运算符

- 介绍
- 示例

15

# 介绍

- 比较运算符

<code>__cmp__(self, other)</code>	# 包含所有的比较情况
<code>__eq__(self, other)</code>	# 单一比较, 等于
<code>__lt__(self, other)</code>	# 单一比较, 小于
<code>__gt__(self, other)</code>	# 单一比较, 大于

- 数字运算

<code>__add__(self, other)</code>
<code>__sub__(self, other)</code>
<code>__mul__(self, other)</code>
<code>__div__(self, other)</code>

- 逻辑运算符

<code>__or__(self, other)</code>
<code>__and__(self, other)</code>



# 示例

```
'''程序猿
'''
class Programmer(object):

    # 构造函数
    def __init__(self, name, age):
        self.name = name        # 可以公开访问
        self.age = age

    # 方法
    def __eq__(self, other):
        return self.age == other.age

    # 方法
    def __add__(self, other):
        return self.age + other.age

p1 = Programmer('monkey1', 12)
p2 = Programmer('monkey1', 12)

print(p1 == p2)
print(p1 + p2)
```

# 类的展现

- 介绍
- 示例

# 介绍

- 对象转换成字符串

<code>__str__</code>	# 供人识别/查看
<code>__repr__</code>	# 供机器识别
<code>__unicode__</code>	# 另外一种编码

- 展现对象属性

<code>__dir__</code>	# 暴露那些可用的方法
----------------------	-------------

# 示例

```
'''程序猿
'''
class Programmer(object):

    # 构造函数
    def __init__(self, name, age):
        self.name = name        # 可以公开访问
        self.age = age

    # 方法
    def __str__(self):
        return '{} is {} years old.'.format(self.name, self.age)

    # 方法
    def __dir__(self):
        return self.__dict__.keys()

p1 = Programmer('monkey1', 12)
print(p1)
print(dir(p1))
```

# Q&A



zhengyscn by 51reboot(<https://www.51reboot.com/>)

# Thank you

[zhengyscn@gmail.com](mailto:zhengyscn@gmail.com) (mailto:zhengyscn@gmail.com)

<http://github.com/zhengyscn> (http://github.com/zhengyscn)

