

Python入门与基础

第二讲 基础知识 Part2 函数与类

林平之老师



扫描二维码关注微信/微博
获取最新面试题及权威解答

微信: [ninechapter](#)

知乎专栏: <http://zhuanlan.zhihu.com/jiuzhang>

微博: <http://www.weibo.com/ninechapter>

官网: www.jiuzhang.com

版权声明

九章课程不提供视频，也严禁录制视频的侵权行为
否则将追求法律责任和经济赔偿
请一定不要缺课

- 函数与类
 - 成员变量与方法
 - 举例LintCode中的ListNode
- 类的继承
- 异常的捕捉



函数 Function

函数的特点：

- 有组织的有一定功能的代码块
- 可重复使用

Python有很多内建的函数，如 `print()`，用户也可以自己定义函数，称之为用户自定义函数。

如何定义一个函数

def 关键词开头

函数名 max

函数参数 a 和 b

函数括号后面需要冒号:

函数体

```
1 def max(a, b):  
2     if a > b:  
3         return a  
4     return b  
5  
6
```

- **def 关键词开头**，之后是函数名和参数
- 参数在圆括号中间，括号后面紧接冒号
- **return语句结束函数**，返回值给调用方
- 如果没有return语句，**最终默认返回None**

```
1  def max(a, b):  
2      if a > b:  
3          return a  
4      return b  
5  
6
```

调用函数使用**函数名**然后在**括号内传入参数**

```
1  def max(a, b):  
2      if a > b:  
3          return a  
4      return b  
5  
6  
7  
8  max(10, 5)  
9
```


- 使用max函数, 求三个数a,b,c的最大值
- 求方程的根
 - <http://www.lintcode.com/en/problem/root-of-equation/>
 - <http://www.jiuzhang.com/solution/root-of-equation/>



类的定义与使用

什么是对象(object)？

世间万物皆为对象 - 对象是一种世界观

- 小狗
- 汽车
- 电脑



对象的属性

小狗：四条腿

汽车：四个轮子，方向盘

电脑：屏幕，键盘

优雅的腿，迷人的眼睛，动人的鼻子都是他的属性



【自挂单身狗】

类 Class 与 对象 Object



九章算法

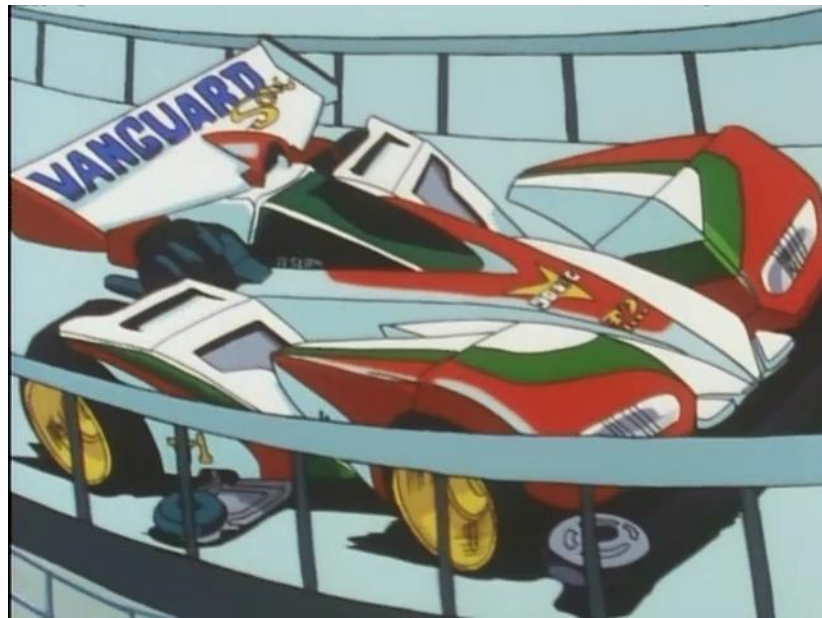
对象的行为

小狗: 汪汪叫

汽车: 加减速, 转弯

电脑: 运行程序, 播放视频

转弯:



class的定义



- 方法 `__init__()` 方法是特殊的方法，为类的构造函数或初始化方法，当创建了这个类的实例对象时就会调用该方法
- `self` 参数代表类的实例，`self` 在定义类的方法时是必须有的，但是调用是你看不到这个参数，因为它就是对象或者实例本身，而不是这个“类”

```
1 class ListNode():
2
3     def __init__(self, val):
4         self.val = val
5         self.next = None
6
7
8
```

- 如何访问类的属性？
- 如何调用类的方法？

类的成员变量与方法



九章算法

print(Dog)

print(dog)

提问: 以上两者有什么不同?

```
1  class Dog:
2
3      def __init__(self, name):
4          self.name = name
5
6      def speak(self):
7          print(self.name + ": Wow")
8
9
10 dog = Dog("Wang Cai")
11
12 dog.speak()
13
14
```

其他的访问类的属性的常用方法：

- `hasattr` 检查是否存在一个属性
- `getattr` 访问对象的属性
- `setattr` 设置一个属性, 如果属性不存在呢？

类的属性 - 单、双下划线、头尾双下划线的情况：

- `__xxx__`: 这是特殊方法, 类似 `__init__()`
- `_xxx`: 单下划线开头表示是 **protected** 类型, 保护类型只允许其本身与子类进行访问 (稍后在继承中讲解)
- `__xxx`: 双下划线表示私有类型(private)类型, 只能是允许这个类中进行访问。
- (对应Java或者Cpp的 **public, protected, private**)

类的常用内置属性：

- `__doc__` 类的文档字符串
 - 以Class Queue举例
- `__dict__` 类的属性展示
- `__name__` 类名

创建一个Vector类:

- “重载”__str__ 方法, 支持展示一个向量如[1,2]
- “重载”__add__ 方法, 支持向量的加和如[1,2] + [3,4]

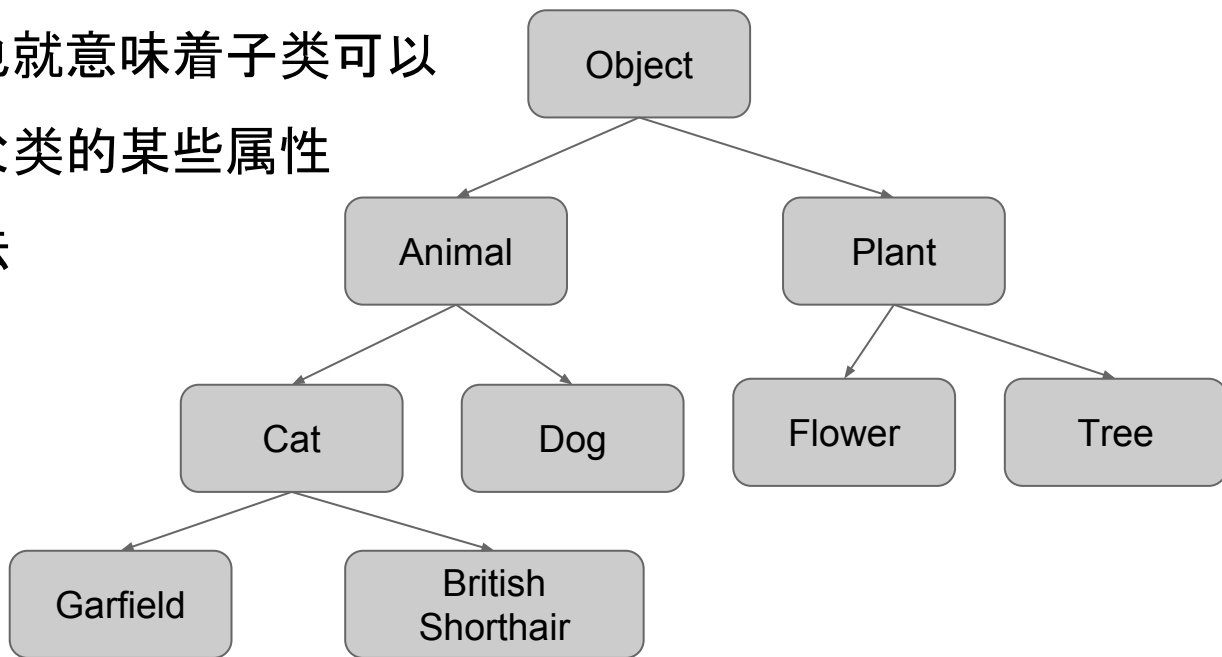
重载 => 覆盖?

类的继承与多态

Inheritance & Polymorphism

子类继承父类：

继承也就意味着子类可以
继承父类的某些属性
和方法



- 创建父类Animal
- 创建子类Cat & Dog, 继承与Animal
 - 子类继承父类的属性和方法
 - 子类使用父类的protected的属性
 - 子类重载父类的方法
 - 子类添加父类没有的属性和方法

提问:Animal这个父类中, 哪些属性和方法是可以被子类继承和使用的?

name ?

__color ?

get_name ?

__get_color ?

__get_color ?

speak() ?

```
1 class Animal:
2
3     def __init__(self, name):
4         self.name = name
5         self._color = "red"
6
7     def get_name(self):
8         return self.name
9
10    def __get_color(self):
11        return self._color
12
13    def __get_color(self):
14        return self._color
15
16    def speak(self):
17        print "LoL"
18
```

Isinstance(obj, type) 可以用来判断对象obj是否是type这种类型

如右的继承关系，四个选项，哪些是True：

A isinstance(dog, Dog)

B isinstance(dog, Animal)

C isinstance(animal, Animal)

D isinstance(animal, Dog)

```
1 class Animal:
2     ...
3
4 class Dog(Animal):
5     ...
6
7
8
```



Python多态的争议性

更多OOD相关内容会在OOD课程当中详细讲解

异常Exception

- 异常是什么？
 - 异常相当于一个事件，该事件会在程序执行过程中发生，影响了程序的正常执行。
 - 程序语句无法正常处理时就会发生异常
 - 异常是对象，表示一个错误。

提问：如果我们不捕捉这个异常，程序将会发生什么？



try/except语句

```
1 try:
2 ...
3 except:
4 ...
5 else:
6 ...
7
```

代码块1:

```
1 a, b = None, None
2 try:
3     a = 2
4 except:
5     a = 100
6 else:
7     b = 200
8
9 print a, b
```

代码块2:

```
1 a, b = None, None
2 try:
3     a[1] = 2
4 except:
5     a = 100
6 else:
7     b = 200
8
9 print a, b
10
```

提问环节:

代码块1和代码块2的

输出结果分别是什么？

可以在except后面指定一些异常如except IOError

提供了一下标准异常如：

- ValueError
- IOError
- RuntimeError
- TypeError
- NameError
- 更多Error: <https://docs.python.org/2/tutorial/errors.html>

- 无论是否发生异常都将执行finally最后的代码
- 做一些无论处理结果如何都必须要做的一些收尾工作
 - 比如最终都需要Close文件

```
1  try:
2      ...
3  finally:
4      ...
5
6
```




扫描二维码关注微信/微博
获取最新面试题及权威解答

微信: [ninechapter](#)

微博: <http://www.weibo.com/ninechapter>

官网: www.jiuzhang.com



谢谢大家