PYTHON BASICS
面向对象编程

2020-04-28

STEVEN WANG

上节总结

	函数式编程	知识点	用处
	函数赋值变量	`f_var = f`	定义调用分离
	函数储存容器	`f_list = [f1, f2, …, fn]`	循环调用
高阶函数	函数当返回	闭包	装饰器
高阶函数	函数当参数	偏函数	柯里化

面向对象

- 基本概念
- 四大特点
- 具体案例

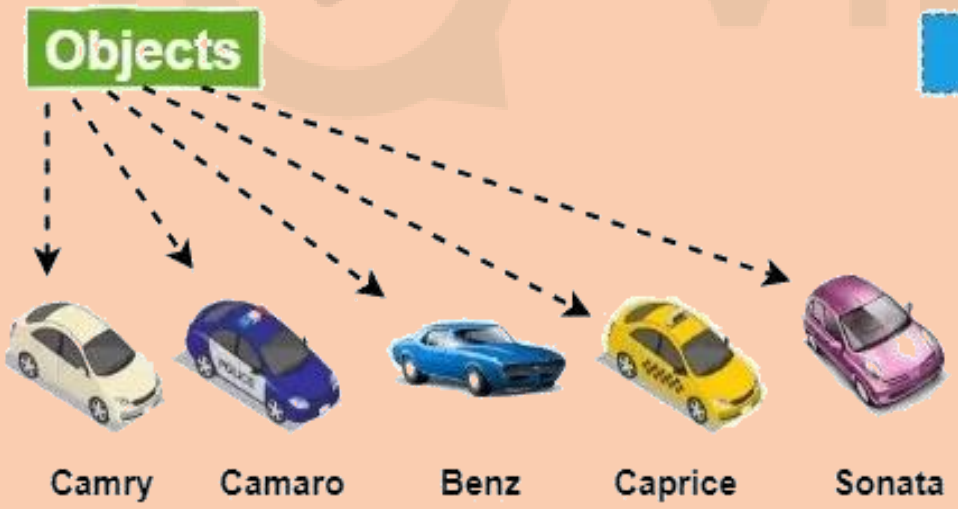
万物皆对象

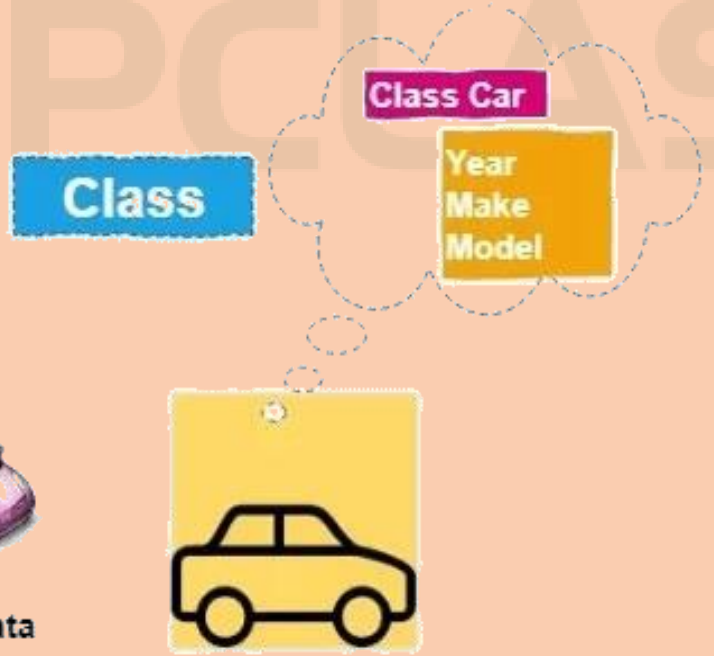
万物可分类

类和对象

类（class）是对某一类事物的描述，是抽象的；而对象（object）是类的一个实例，是具体的：

- 「人」是类，而「运动员」则是「人」的一个实例。
- 「金融产品」是类，而「外汇期权」则是「金融产品」的一个实例。





实例化

类 = 抽象描述

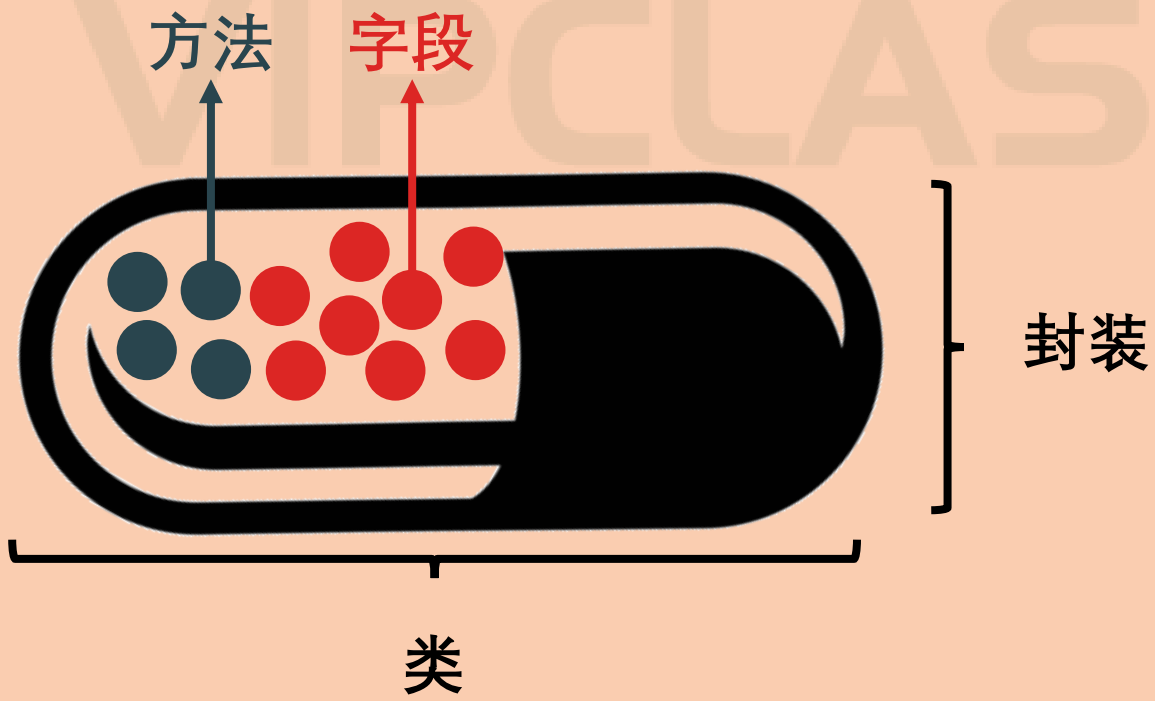
对象 = 具体实例

类和对象

之前介绍的变量 (variables) 和函数 (functions) 是零散的，而类和对象将它们做封装 (encapsulation)，在类和对象里也有

- 变量用来存储数据，这时变量又称字段 (fields)
- 函数用来操作数据，这时函数又称方法 (methods)

} 属性 (attributes)



对象初体验

整数

```
i = 1031

type(i)

i.numerator

i.bit_length()

i + 10

i * 1.2

i.__add__(10)

i.__mul__(1.2)
```

字符串

```
s = 'BABA'

type(s)

s[0]

s.lower()

'ALI' + s

s * 2

'ALI'.__add__(s)

s.__mul__(2)
```

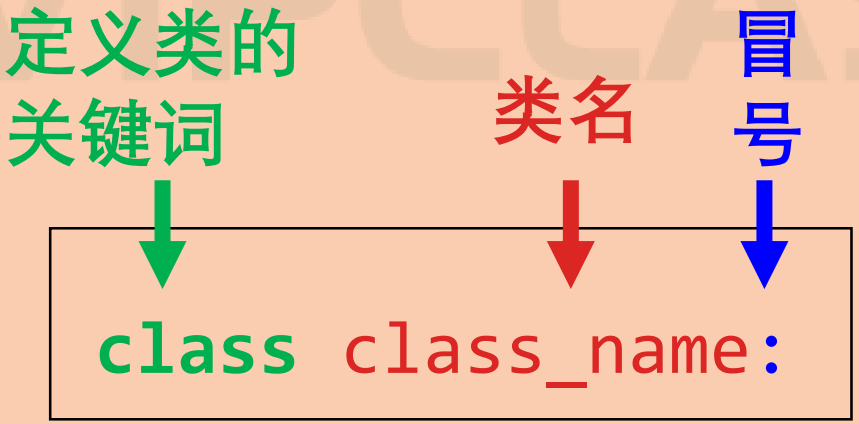
列表

```
l = [1, 2, 3]

type(l)

l[2]

l.append(4)

l + [5, 6]

l * 2

l.__add__([5, 6])

l.__mul__(2)
```

定义类

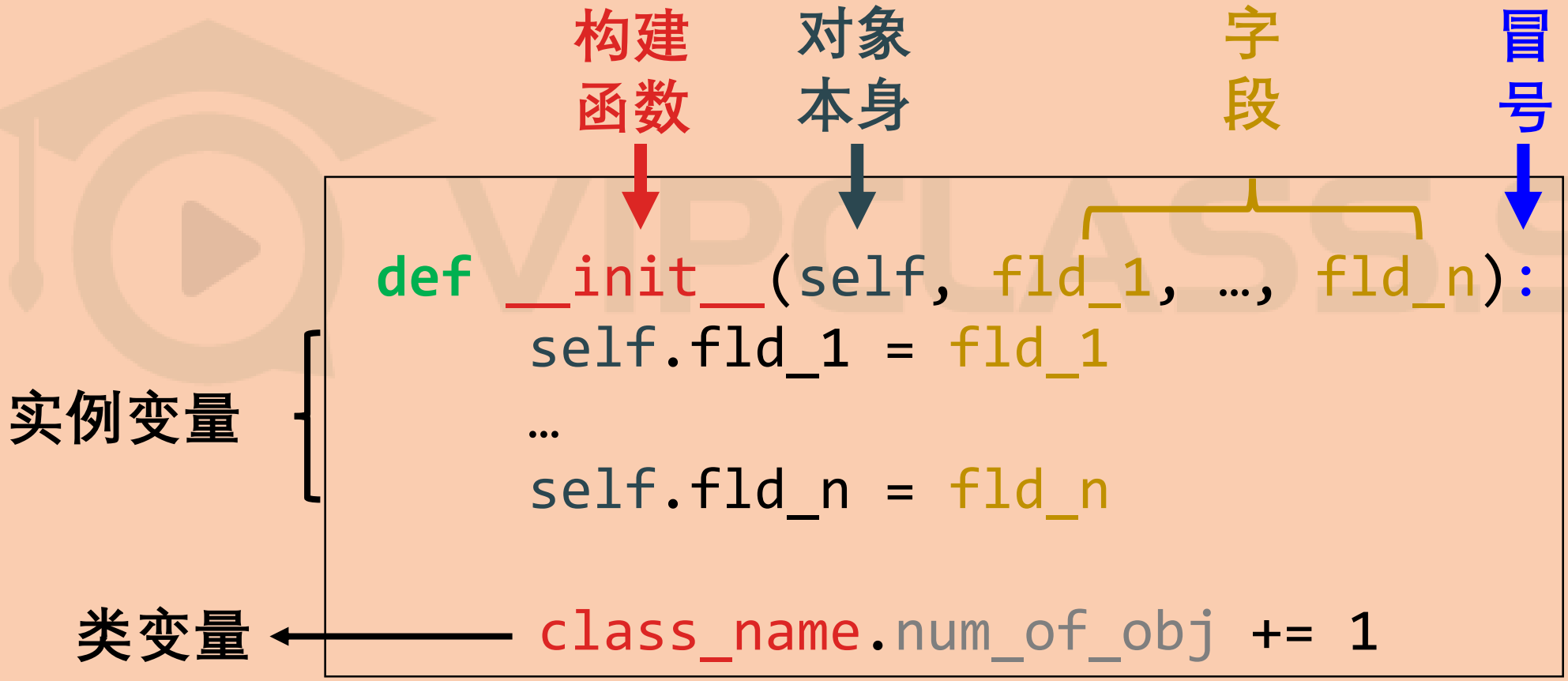
类的名称通常使用 camel case + 首个字母大写的惯例。

- Dog
- EuropeanOption



实例变量、类变量

用 __init__() 方法来构建对象，第一个参数永远是 self，表示创建对象本身，后面的参数都是字段名。



实例方法、类方法、静态方法


```python
class MyClass:

    def method(self):
        return '实例方法', self

    @classmethod
    def classmethod(cls):
        return '类方法', cls

    @staticmethod
    def staticmethod():
        return '静态方法'
```

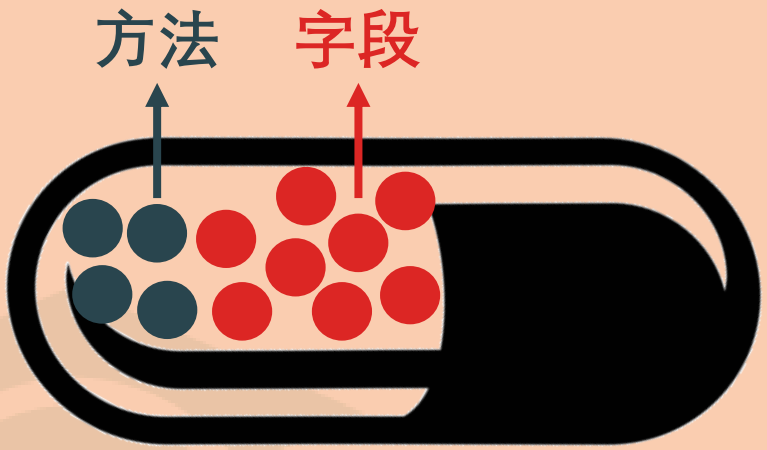
```python
class 斗地主:

    def start_game(self):
        print('欢迎', self.name)

    @classmethod
    def top_score(cls):
        print('恭喜', cls.top_score)

    @staticmethod
    def help():
        print('欢乐豆不足，请充值')
```

面向对象

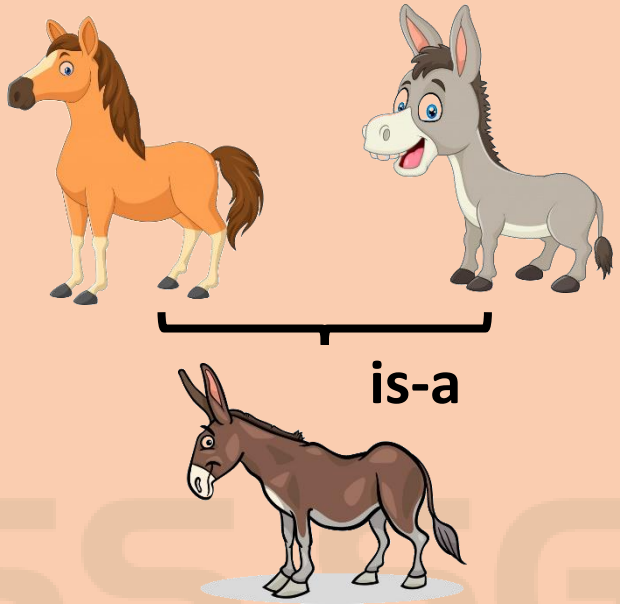
- 基本概念
- 四大特点
- 具体案例

四大特点

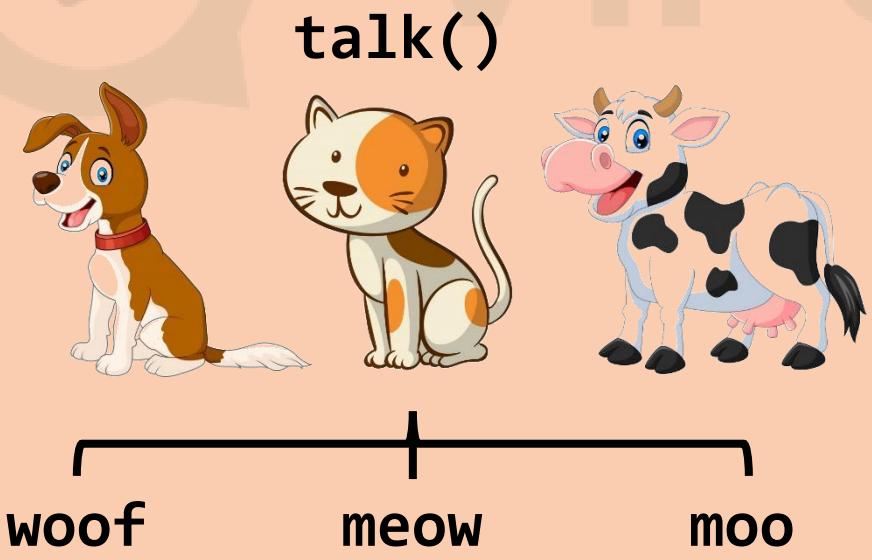
封装



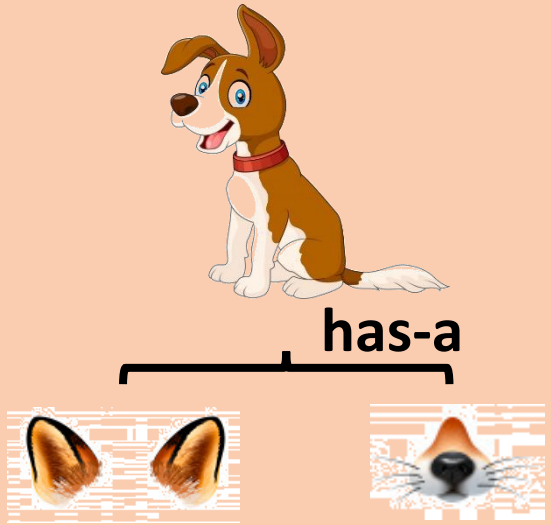
继承



多态



组合

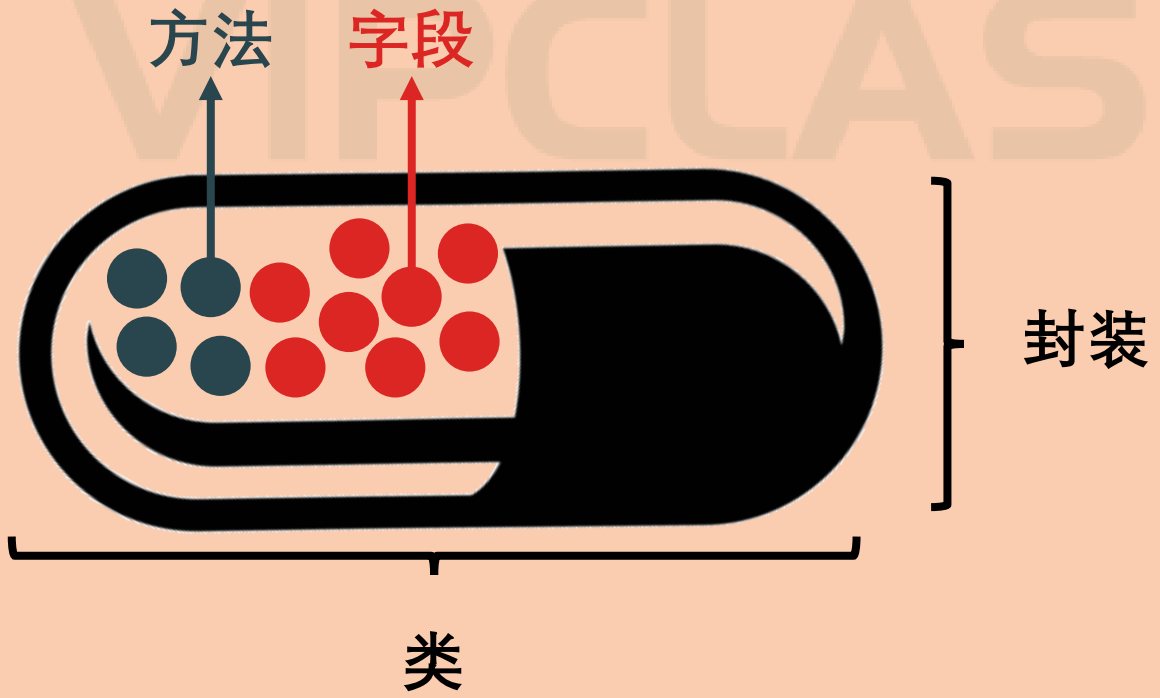


封装

封装 (encapsulation) 包含两个概念：

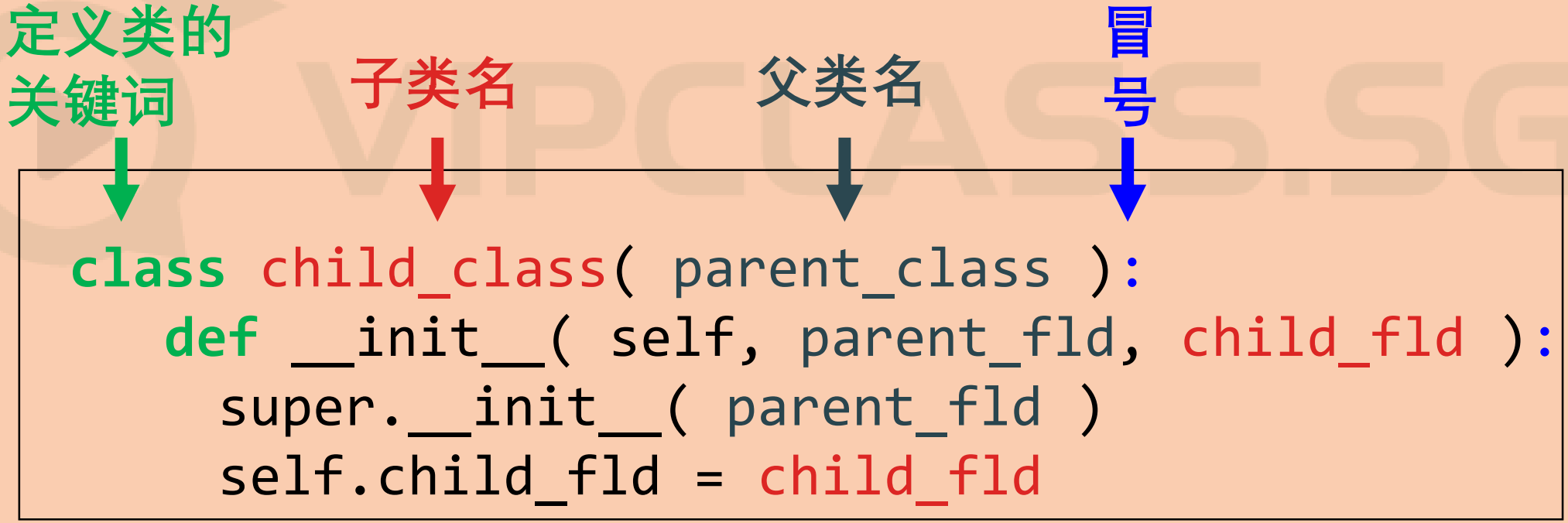
1. 类将字段和方法集中在一起。

2. 隐藏类的内部数据，避免外部代码直接进行访问。



继承

编程中的继承 (inheritance) 与基因继承类似。在创建类时，该类也可以从另一个类那里继承字段和方法。被继承的类，称为父类 (parent class)；继承的类则被称为子类 (child class)。子类和父类有 “is-a” 这样的关系。



多态

多态 (polymorphism) 指的是“为不同的对象提供相同的函数或方法”。想想 len() 是不是可以用在字符串、元组和列表这些不同对象上?

无多态

```
shapes = [trl, sql, crl]

for shape in shapes:
    if type(shape) == "Triangle":
        shape.draw_triangle()
    if type(shape) == "Square":
        shape.draw_square()
    if type(shape) == "Circle":
        shape.draw_cirlce()
```

多态

```
shapes = [trl, sql, crl]

for shape in shapes:
    shape.draw()
```

通过统一多态的接口，可以随意向 shapes 列表中添加新图形，不需再添加额外的代码即可画出对应图形。

组合

组合 (composition) 将一个对象作为变量保存在另一个对象中，可以模拟“has-a”关系。例如可用组合来表达狗和其主人之间的关系 (狗有主人)。

合成

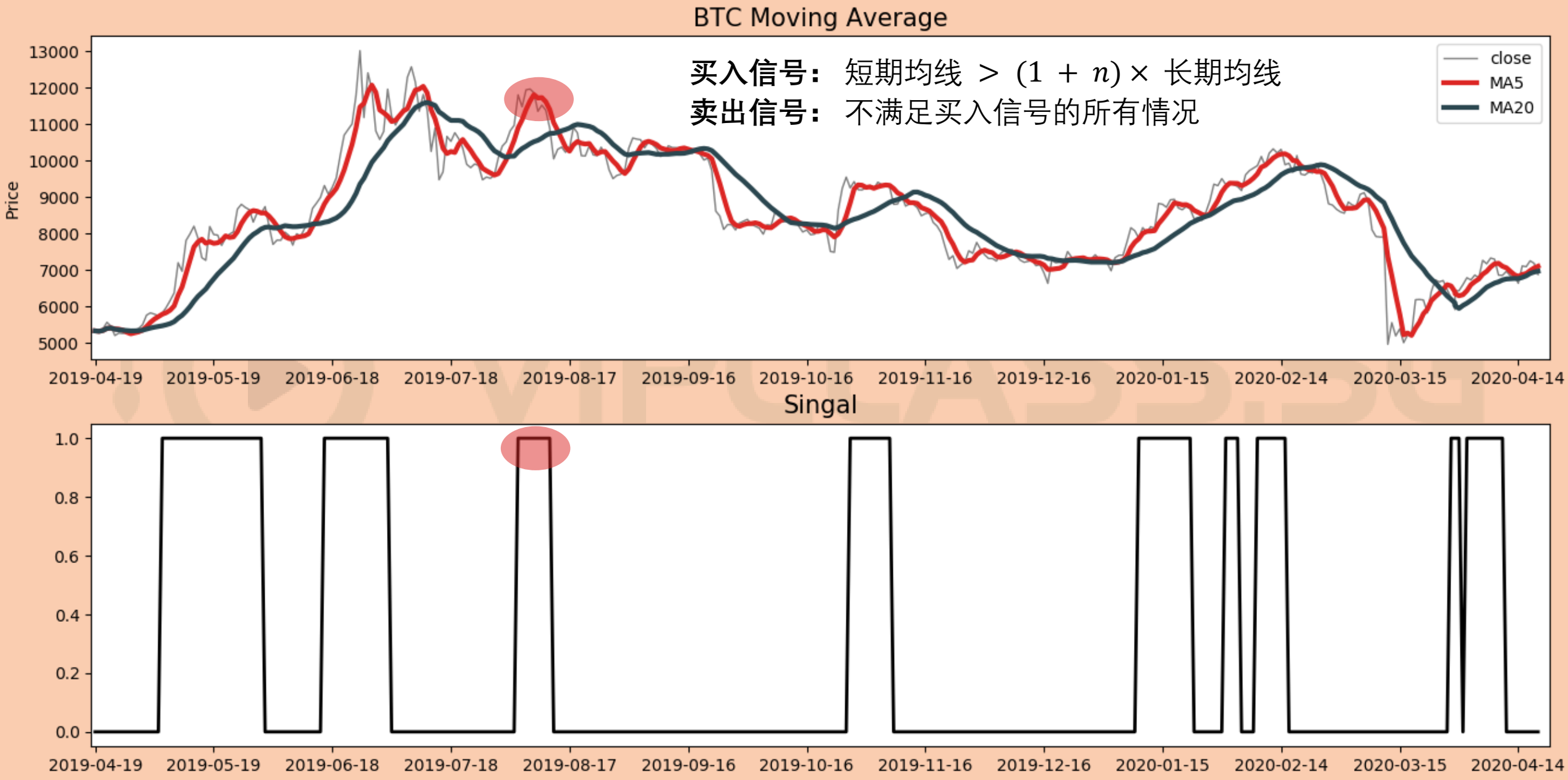
成份

```
class composite:
    def __init__( self, fld, composition ):
        self.fld = fld
        self.composition = composition
```

面向对象

- 基本概念
- 四大特点
- 具体案例

双均线交易


BTC Moving Average
买入信号： 短期均线 ＞ (1 ＋ n) × 长期均线
卖出信号： 不满足买入信号的所有情况
close
MA5
MA20
Price
13000
12000
11000
10000
9000
8000
7000
6000
5000
2019-04-19
2019-05-19
2019-06-18
2019-07-18
2019-08-17
2019-09-16
2019-10-16
2019-11-16
2019-12-16
2020-01-15
2020-02-14
2020-03-15
2020-04-14
Singal
1.0
0.8
0.6
0.4
0.2
0.0

期权定价

期：未来　　认购期权：拥有未来买入某个东西的权利 (锁定买入价)

权：权利　　认沽期权：拥有未来卖出某个东西的权利 (锁定卖出价)

期权类别　标的价格　年限　　行权价格　标准正态变量累积分布函数

$$V(S,K,r,q,T,\sigma,\omega)=\omega\cdot[Se^{-qT}\cdot\Phi(\omega\cdot d_+)-Ke^{-rT}\cdot\Phi(\omega\cdot d_-)]$$

分红率　　利率

$$d_\pm=\frac{1}{\sigma\sqrt{T}}\ln\left(\frac{Se^{(r-q)T}}{K}\right)\pm\sigma\sqrt{T}$$

波动率

总结

特点	含义	伪代码	用处
封装	将属性打包在一起	`o.fld, o.mtd()`	紧凑
继承	使用父类重复属性 (is-a)	`super.__init__()`	简洁
多态	将多对象的方法统一	`o1.mtd(), o2.mtd()`	一致
组合	对象中有其他对象 (has-a)	`self.__init__(o)`	实际

下节预告：格式化和正则


VIPCLASS.SG
small class, world class
终身学习
快乐学习
王圣元 Steven Wang
微信公众号：王的机器