

类

类的定义

- 属性(变量)
- 函数

```
1 class MyClass:
2     def __init__(self):
3         self.name = 'itcast'
4
5     def say_hello(self):
6         print('hello')
```

!!!note

`class` 为关键字，用来声明一个类

```
1 `__init__` 是构造函数，构造函数也是函数，是一个具体实例对象创建时默认调用的函数。
2
3 `self` 表示当前创建实例对象本身
4
5 类中的函数，有个特点，默认第一个参数都是 `self`
```

对象

- 类是一种模板模型
- 对象是这个类的具体实现

!!!tip

狗和旺财，哪一个类？哪一个对象？

```
1 狗是一种类型，属于模板
2
3 旺财是狗的实现，属于具体的，具体的就是对象
```

类的使用

类的使用，其实就是将类具体化，获得对象，然后使用对象的属性和方法

```
1 class Car:
2     def __init__(self):
3         self.speed = 10
4         self.x = 0
5
6     def move(self):
7         self.x += self.speed
```

!!!note

`self.speed` 是属性，用来记录数据的

```
1  `self.x` 是属性，用来记录数据的
2
3  `move` 是函数，是一种行为，行为的变化会产生数据的变化
4
5  整个对象，其实就是维护状态数据的。
```

面向对象

提到面向对象，就不得不提到面向过程。

面向对象和面向过程其实都是一种编程思想，就是解决问题的思路。

面向过程

面向过程，全称POP(Process Oriented Programming)。

- 面向过程强调的是过程
- 完成这个过程是，站在单一执行者的角度去完成这个过程

例如：中午想吃西红柿炒鸡蛋怎么办呢？

如果用面向过程的思维，可以分解为一下过程：

```
1  第一步，到超市买西红柿和鸡蛋
2  第二步，把西红柿洗干净切好
3  第三步，把鸡蛋炒好
4  第四步，把西红柿炒好
5  第五步，把鸡蛋和西红柿混在一起炒好
6  第六步，把炒好西红柿炒蛋端出来
```

面向对象

面向对象，全称OOP(Object Oriented Programming)

- 面向对象不再关注具体的过程细节
- 面向对象注重的是关系管理和调度

同样是西红柿炒鸡蛋，面向对象会这么做：

```
1  采购员去超市采购
2  洗菜员负责洗菜
3  切菜员负责切成
4  厨师负责烹饪
5  上菜员负责上菜
```

每一条都很模糊，但是每一条也明确了要干什么，面向对象就是将这些角色的调度做好。

探讨

如果以西红柿炒蛋为例，面向过程的做法，类似于一个小老板开了一个小餐厅，前前后后都由自己打点。

而面向对象的做法，类似于一个连锁饭店，分工明确，可复制性比较强。

其实我们的编程是面向过程和面向对象的叠加。调度管理部分我们采用的是面向对象，细节流程部分采用的是面向过程。

例如：西红柿炒鸡蛋

- 1 采购员去超市采购
- 2 采购员怎么采购？具体化的流程，1，2，3...
- 3 洗菜员负责洗菜
- 4 洗菜员怎么洗菜？具体化的流程，1，2，3...
- 5 切菜员负责切成
- 6 切菜员怎么切菜？具体化的流程，1，2，3...
- 7 厨师负责烹饪
- 8 厨师怎么烹饪？具体化的流程，1，2，3...
- 9 上菜员负责上菜
- 10 上菜员怎么上菜？具体化的流程，1，2，3...