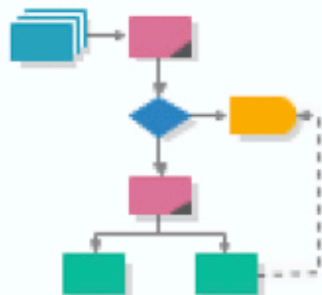


# 图解

# Python

可能是迄今为止

最易懂的Python视频教程



---

```
"""继承的概述"""
```

```
"""
```

## 一、为什么需要继承

在下面的程序中，由于狗和鸟都具有吃饭和喝水这两个功能，所以，类对象**Dog**和**Bird**中都定义了**eat()**和**drink()**这两个方法，并且这两个方法的代码是完全相同的，这样就导致了程序中存在重复的代码。

```
class Dog(object):  
    def eat(self):  
        print("吃饭")
```

```
    def drink(self):  
        print("喝水")
```

```
    def swim(self):  
        print("游泳")
```

```
class Bird(object):  
    def eat(self):  
        print("吃饭")
```

```
    def drink(self):  
        print("喝水")
```

```
    def fly(self):  
        print("飞翔")
```

```
"""
```

---

.....

## 二、什么是继承

当几个类对象中有共同的属性和方法时，就可以把这些属性和方法抽象并提取到一个基类中，每个类对象特有的属性和方法还是在本类对象中定义，这样，只需要让每个类对象都继承这个基类，就可以访问基类中的属性和方法了。继承基类的每个类对象被称为派生类。基类也被称为父类或超类，派生类也被称为子类。

Python中的所有类对象都继承自一个统一的基类：`object`。这就是为什么我们在定义类对象时要在类名后面添加(`object`)。

除了封装，继承也是面向对象编程的三大特征之一。继承是实现代码复用的重要手段。

.....

```
class Animal(object):  
    def eat(self):  
        print("吃饭")
```

```
    def drink(self):  
        print("喝水")
```

```
class Dog(Animal):  
    def swim(self):  
        print("游泳")
```

```
class Bird(Animal):  
    def fly(self):  
        print("飞翔")
```

---

```
dog = Dog()  
dog.eat()  
dog.drink()  
dog.swim()
```

```
bird = Bird()  
bird.eat()  
bird.drink()  
bird.fly()
```