

## python\_继承

python中继承这里主要写3点，一个是继承调用关系，一个是方法的复写，最后一个是继承后实例方法的使用等

### 1、继承，实例化等，通过实例进行巩固练习

```
class animal: #父类

    def __init__(self,leg,hand):

        self.leg=leg

        self.hand=hand

    def run(self):

        print(self.leg)

    def fly(self):

        print(self.hand)

class bird(animal): #继承animal父类

    def high(self):

        self.fly() #子类中可调用父类方法

        print("high")

        print(self.leg) #子类中可调用父类属性

t=bird('leg','hand')

t.fly() #子类实例对象可调用本类方法

t.high() #子类实例对象可调用父类方法

t.run() #子类实例对象可调用父类方法

print(t.hand) #子类实例对象可调用父类属性

print(t.leg) #子类实例对象可调用父类属性
```

### 2、子类复写父类方法

```
class animal: #父类
```

```
    def __init__(self, leg, hand):
```

```
        self.leg=leg
```

```
        self.hand=hand
```

```
    def run(self):
```

```
        print(self.leg)
```

```
    def fly(self):
```

```
        print(self.hand)
```

```
class bird(animal): #继承animal父类
```

```
    def fly(self): #继承父类方法功能之外在新增功能
```

```
        super().fly() #继承父类方法
```

```
        print("fly") #子类方法新功能
```

```
    def run(self): #完全复写父类方法
```

```
        print("run---run")
```

```
t=bird('leg', 'hand')
```

```
t.fly() #调用子类方法
```

```
t.run() #调用子类方法
```

### 3、多继承

```
class A(object):#新式类
```

```
    def say(self):
```

```
        print('A')
```

```
class B(A):
```

```
    def say(self):
```

```
        print('B')
```

```
class C(A):
```

```
    def say(self):
```

```
        print('c')
```

```
class D(C, B):
```

```
    pass
```

```
d = D()
```

```
print(D.mro()) #复杂多继承通过mro, 查看继承先后顺序, python是根据这个顺序进行运行的
```

```
d.say()
```

