# 实例属性、类属性

在面向对象开发中，使用类创建出来的实例是一个对象，那么，类是否是一个对象呢?

## 实例属性

通过类创建的对象被称为 **实例对象**，对象属性又称为实例属性，记录对象各自的数据，不同对象的同名实例属性，记录的数据各自独立，互不干扰。

```
class Person(object):
    def __init__(self,name,age):
        # 这里的name和age都属于是实例属性，每个实例在创建时，都有自己的属性
        self.name = name
        self.age = age

# 每创建一个对象，这个对象就有自己的name和age属性
p1 = Person('张三',18)
p2 = Person("李四",20)
```

## 类属性

类属性就是类对象所拥有的属性，它被该类的所有实例对象所共有，类属性可以通过类对象或者实例对象访问。

```
class Dog:
    type = "狗"  # 类属性

dog1 = Dog()
dog2 = Dog()

# 不管是dog1、dog2还是Dog类，都可以访问到type属性
print(Dog.type)  # 结果：狗
print(dog1.type)  # 结果：狗
print(dog2.type)  # 结果：狗
```

## 使用场景：

1. 类的实例记录的某项数据始终保持一致时，则定义类属性。
2. /实例属性要求每个对象为其单独开辟一份内存空间来记录数据，而类属性为全类所共有 ，仅占用一份内存，更加节省内存空间。

## 注意点：

1> **尽量避免类属性和实例属性同名**。如果有同名实例属性，**实例对象会优先访问实例属性**。

```
class Dog(object):
    type = "狗"  # 类属性

    def __init__(self):
        self.type = "dog"  # 对象属性

# 创建对象
dog1 = Dog()

print(dog1.type)      # 结果为 “dog”    类属性和实例属性同名，使用 实例对象 访问的是 实例属性
```

2> **类属性只能通过类对象修改，不能通过实例对象修改**

```
lass Dog(object):
    type = "狗"  # 类属性

# 创建对象
dog1 = Dog()
dog1.type = "dog"   # 使用 实例对象 创建了对象属性type

print(dog1.type)      # 结果为 “dog”    类属性和实例属性同名，访问的是实例属性
print(Dog.type)       # 结果为 "狗"    访问类属性

# 只有使用类名才能修改类属性
Dog.type = "土狗"
print(Dog.type)  # 土狗
dog2 = Dog()
print(dog2.type)  # 土狗
```

3> **类属性也可以设置为私有**，前边添加两个下划线。 如:

```
class Dog(object):
    count = 0  # 公有的类属性
    __type = "狗"  # 私有的类属性

print(Dog.count)        # 正确
print(Dog.__type)       # 错误,私有属性，外部无法访问。
```