

Python的类

Python进阶

李靖远见/元宇宙实验中心
yyzlab.com.cn



目录

CONTENTS

01.

面向过程与面向对象

02.

类的定义与创建对象

03.

类的封装

04.

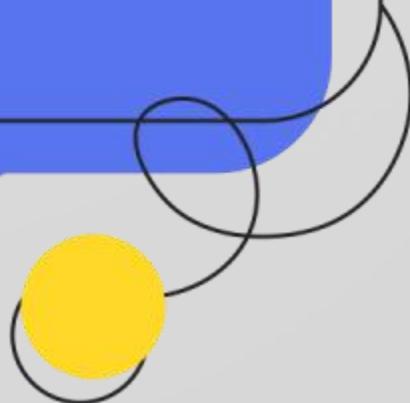
类的继承

05.

类的多态



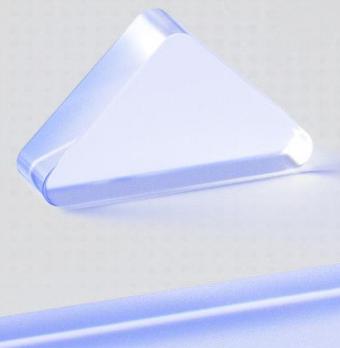
01 面向过程与面向对象





面向过程与面向对象

- 面向过程编程 (POP) : 是一种以过程为中心的编程思想，它将程序看作一系列的命令、函数或过程的集合，主要关注解决问题的步骤和过程。在面向过程编程中，程序被划分为一系列的函数，每个函数负责完成一个特定的任务。数据在整个程序都是自由流动的，任何函数都可以去访问和修改数据。
- 面向对象编程 (OOP) : 将程序看作一系列对象的集合，每个对象都是独立的个体，包含数据（属性）和操作这些数据的方法（行为）。这些对象之间通过发送消息（调用方法）来相互作用，共同解决某种问题。





面向过程与面向对象

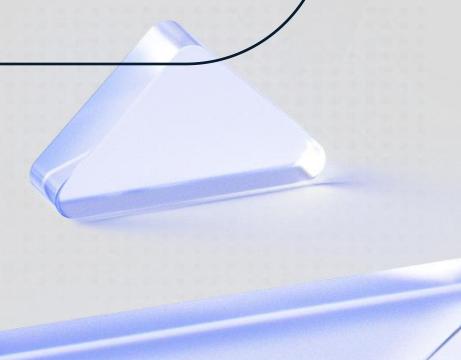
面向过程的优点：

- 简单直观
- 性能高效
- 代码简洁

面向过程的缺点：

- 不易维护
- 不易扩展
- 代码重用性低

华清远见/元宇宙实验室
yyzlab.com.cn





面向过程与面向对象

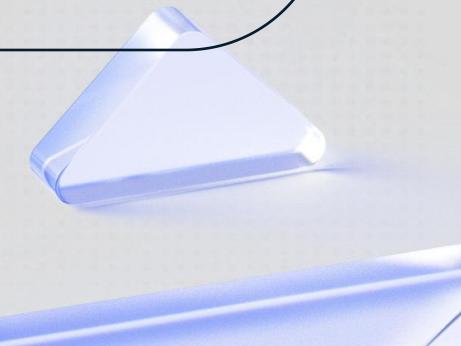
面向对象的优点：

- 模块化
- 安全性高
- 代码重用性高

面向对象的缺点：

- 学习难度大
- 性能开销大
- 调试困难

华清远见/元宇宙实验室
yyzlab.com.cn

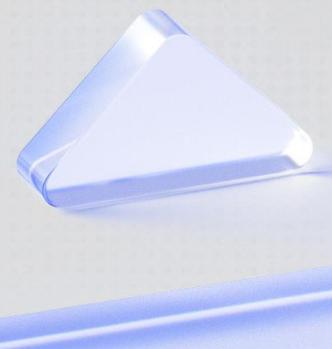




面向过程与面向对象

从本质上来说，面向过程与面向对象都是用来解决问题的，只是看待问题的角度不同：

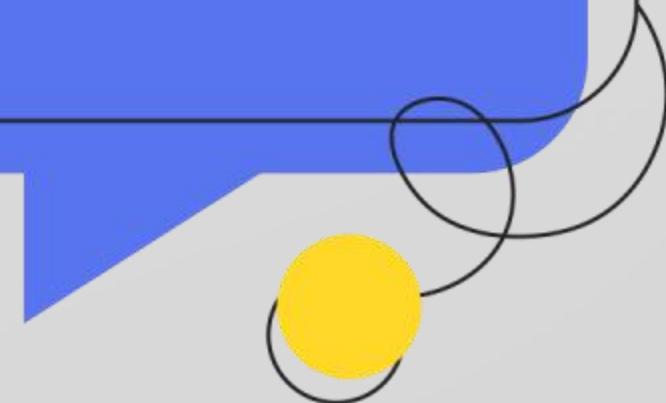
- 面向过程更多的是注重解决问题的步骤，编程时会详细规划和控制程序的流程，程序按照步骤执行下去，问题就能解决。
- 面向对象更多的是注重对象之间的交互，将问题抽象为类，然后创造对象，通过对对象调用方法，从而协同解决问题。





02 类的定义与创建 对象

华清远见 YYZlab.com.cn





类的定义与创建对象

类的定义：在Python中，使用class关键字进行定义一个类。

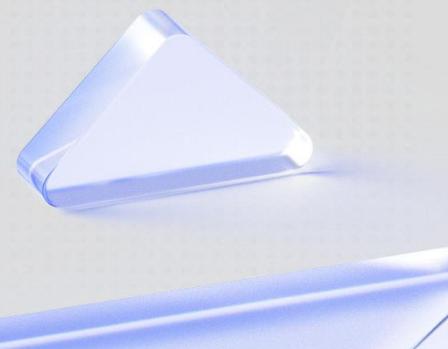
注意：类名需要使用大驼峰命名法。

定义格式为：

class 类名:

 pass

```
class MyClass:  
    pass
```





类的定义与创建对象

类的调用，也称类的实例化，实例化出来的内容就称为对象，该过程也叫做创建对象。

创建对象的格式为：

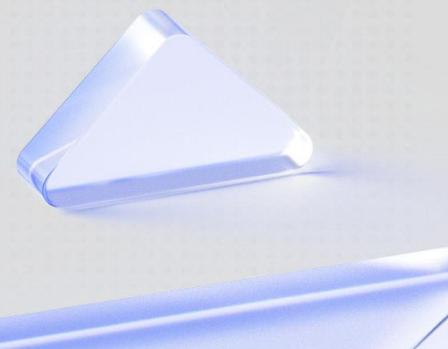
对象名 = 类名()

1个用法

```
class MyClass:  
    pass
```

```
myclass = MyClass()
```

华清远见元宇宙实训中心
yyzlab.com.cn



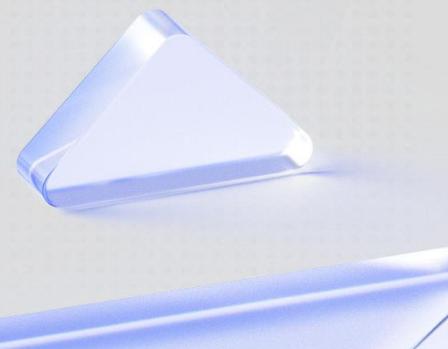


类的定义与创建对象

对象与类之间的关系：

- 对象拥有类所定义的全部的属性和行为。
- 对象的属性和行为可以单独进行增加、删除、修改。
- 对象不能单独创建，必须依托类的实例化，且一个类可以实例化无数的对象。
- 对象之间的属性和行为是不共享的。

华清远见教育实验中心
www.yjx.com.cn





类的定义与创建对象

在Python中，一切皆对象!!!!

```
num = 1  
my_str = 'hello'  
my_list = [1, 2, 3, 4, 5]  
my_tuple = (1, 2, 3)  
  
print(type(num))  
print(type(my_str))  
print(type(my_list))  
print(type(my_tuple))
```

```
<class 'int'>  
<class 'str'>  
<class 'list'>  
<class 'tuple'>
```



类的定义与创建对象

self：是一个参数，表示对象自身，里面存放着对象自身的地址。如果希望类中的方法可以被对象调用，那么第一个参数必须是self。其作用就是将实例对象与类的方法进行绑定，这样才能让每个对象都能调用“属于”自己的方法。

```
class Dog:  
    2个用法  
        def speak(self):  
            print(f'{self}在说话')  
  
dog1 = Dog()  
dog1.speak()  
dog2 = Dog()  
dog2.speak()
```

```
<__main__.Dog object at 0x000001E4FDAF2070>在说话  
<__main__.Dog object at 0x000001E4FDB18E50>在说话
```



类的定义与创建对象

构造函数与析构函数：

构造函数`__init__()`：在创建对象时自动调用的一种函数，用来进行初始化属性，不是必须要定义的，可以在需要的时候再定义。

析构函数`__del__()`：在对象的引用清零时会自动调用的一种函数，一般用来进行释放资源的操作。不推荐使用，因为Python有自己的垃圾回收机制。

类的定义与创建对象

类属性：在类定义时所定义的属性，用来说明类本身。

实例属性：属于对象本身的属性，存在于`__init__`函数中，在对象创建时会赋值给该对象。

实例方法：第一个参数传入`self`参数，没有实例化前，无法调用，实例化后，也多使用对象来调用而不是用类来调用。

静态方法：与类里的数据不共享，有自己的参数，类和对象都可以调用。

类方法：第一个参数传入`cls`参数，表示类本身，类和对象都可以调用，用于访问类本身的属性。



03 类的封装

华清远见/广州实验室中心
yyzlab.com.cn

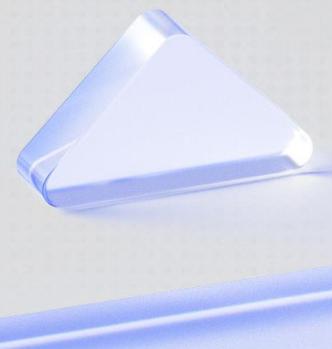




类的封装

类的封装：

就是将数据和函数封装到一个类中，并且通过设置访问权限可以将类的属性和方法隐藏在类的内部，可以防止类的属性和方法被外部直接访问和修改，提高了代码的安全性和可维护性。

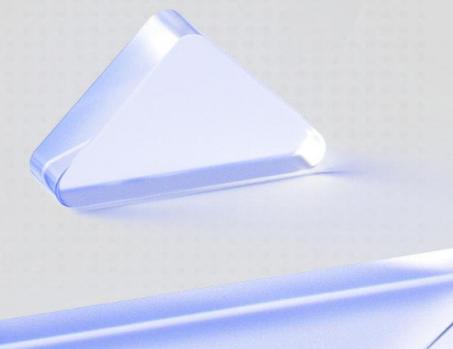




类的封装

在Python，可以通过给属性名和方法名添加下划线(_)设置权限，又根据下划线的个数分为不同的类型：

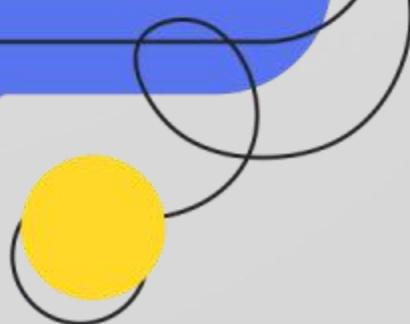
- 单下划线前缀：以单下划线为开头命名的属性或方法，表示该属性或方法是内部使用的，但这是人为规定的，事实上程序是可以访问的。
- 双下划线前缀：真正的设置私有权限的方法，通过在属性名或行为名之前添加双下划线就可以将该属性或行为定义为私有的，且不可被对象访问。
- 双下划线前后缀：表示Python中的特殊的属性或方法，是Python内置好的内容，有特殊的意义和用途，不推荐自己定义。





04 类的继承

华清远见/广州实验室中心
www.yyzlab.com.cn





类的继承

在Python中，一个类可以被继承，也可以去继承别的类，其中被继承的类被称为父类或基类，继承的类叫做子类。在继承过程中，子类会继承父类的所有的属性和行为，并且一个子类可以有多个父类，一个父类可以有多个子类。

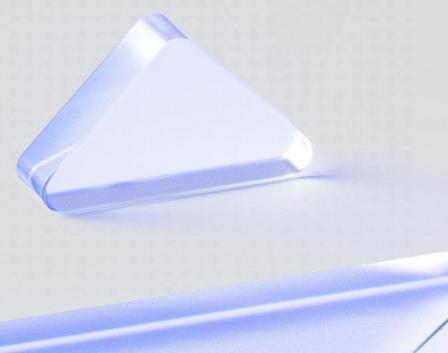
格式为：

```
class Person(要继承的类):
```

```
    pass
```

华清园见元实验室

华清园见元实验室
yyzlab.com.cn





类的继承

在Python3.x中，所有的类都有一个共同的父类，叫做object类，即使你在自定义自己的类时，也会默认继承object类，子类会拥有object类的所有的属性和方法。

```
class Person():
    pass
```

```
class Person(object):
    pass
```

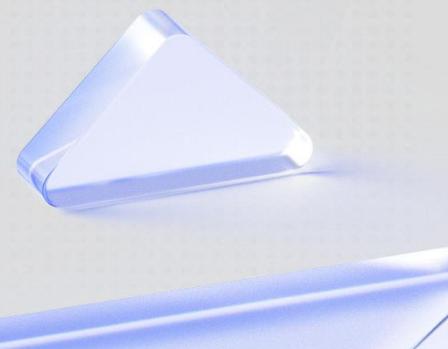




类的继承

object类：

| 名称 | 作用 |
|-------------------------|-----------------------------|
| <code>__new__()</code> | 创建对象 |
| <code>__init__()</code> | 对对象进行初始化 |
| <code>__eq__()</code> | 定义比较操作符 <code>==</code> 的行为 |
| <code>__iter__()</code> | 使对象支持迭代 |
| <code>__next__()</code> | 在迭代中返回下一个值 |
| <code>__class__</code> | 对象所属的类 |
| <code>__doc__</code> | 对象的文档字符串 |
| <code>__name__</code> | 类的名称 |





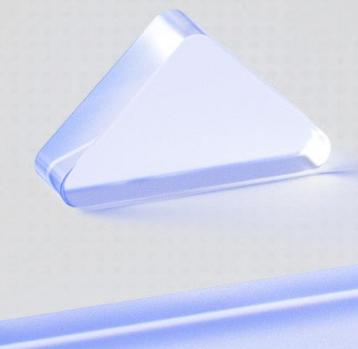
类的继承

单继承：一个类只有一个父类，就是单继承。当子类中存在与父类相同的属性或方法时，子类的属性或方法会覆盖掉父类的属性或方法。

多继承：一个类有多个父类，就是多继承。多继承可以提供更多的功能，但是也可能导致继承冲突，比如子类和多个父类有相同的属性和方法。那么此时在使用该属性或方法时，其顺序为：

子类>从左到右第一个父类>第二个父类>第三个>...

对于复杂的继承关系，使用子类的mro()方法获取其继承顺序。





05 类的多态

华清远见/广州实验室中心
www.yyzlab.com.cn

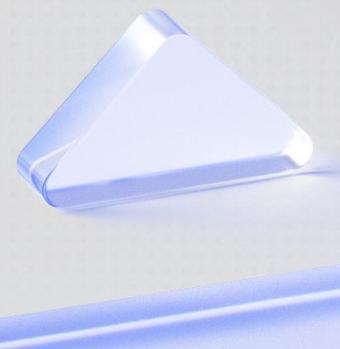




类的多态

多态：指允许不同对象对同一个方法做出不同的响应，即同一个方法可以根据不同对象的类型而表现出不同的行为，可用于设计更加灵活的代码，提高代码的重复使用。

比如 `len()` 函数，对于字符串来说就是统计字符串的长度，对于列表就是统计列表元素的个数，可以根据括号里的数据类型的不同，进行不同的操作。

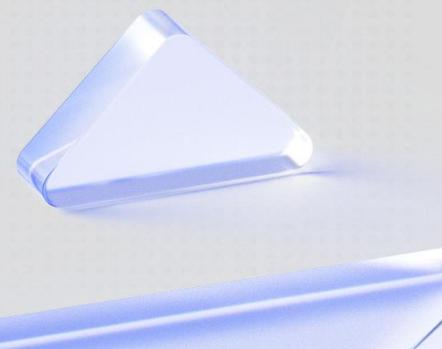




类的多态

在类的继承中所用到的方法的重写就是一种实现多态的方式。通过重写父类中的属性和方法，使其拥有自己独特的属性和方法，这就是多态的一种表现。

华清远见 YYZLAB.COM.CN



科技赋能·智引未来



技术领先
品质保障



超多干货
实时更新



海量视频
贴身学习