

面向对象 Object Oriented

详细说明

面向对象

类和对象

面向对象

分析出解决问题的步骤，然后逐步实现。

公式

程序 = 算法 + 数据结构

优点

所有环节、细节自己掌控

缺点

考虑所有细节，工作量大

例如：婚礼筹备

发请柬（选照片、措词、制作）

宴席（场地、找厨师、准备桌椅餐具、计划菜品、购买食材）

婚礼仪式（定婚礼仪式流程、请主持人）

面向对象

找出解决问题的人，然后分配职责。

公式

程序 = 对象 + 交互

优点

思想层面

可模拟现实情景，更接近于人类思维

有利于梳理归纳、分析解决问题

技术层面

高复用

对重复的代码进行封装，提高开发效率

高扩展

增加新的功能，不修改以前的代码

高维护

代码可读性好，逻辑清晰，结构规范

学习曲线陡峭

基础知识

1.类：一个抽象的概念，即生活中的“类别”。（抽象）

2.对象：类的具体实例，即归属于某个类别的“个体”。（具体的）

3.类是创建对象的“模板”。

4.类与类行为不同，对象与对象数据不同。

5.类和对象内存图

数据成员：名词类型的状态。

方法成员：动词类型的行为

定义类

1.代码
class 类名:
def __init__(self,参数列表):
self.实例变量 = 参数
方法成员

#类名所有单词首字母大写,多个单词不用下划线隔开。
(实例变量 指CPU寻址)

说明

类名所有单词首字母大写。

__init__ 也叫构造函数，创建对象时被调用，也可以省略

self 变量绑定的是被创建的对象，名称可以随意

创建对象(实例化)

变量 = 类名(参数列表)

实例成员

语法

(1)定义：对象.变量名

(2)调用：对象.变量名

说明

(1)首次通过对象赋值为创建，再次赋值为修改。

(2)通常在构造函数(__init__)中创建。

(3)每个对象存储一份，通过对象地址访问。

作用

描述某个对象的数据。

__dict__：对象的属性，用于存储自身实例变量的字典。

w01 = Wife()

w01.name = "丽丽"

w01 = Wife("丽丽",24)

print(w01.name)

语法

类变量

语法

(1)定义：在类中，方法外定义变量。

(2)调用：类名.变量名

说明

存储在类中。

只有一份，被所有对象共享。

作用

描述所有对象的共有数据。

类变量内存图

类成员与实例成员访问方式

类变量的应用

类成员

语法

定义 @classmethod
def 方法名称(cls,参数列表):
方法体

调用 类名.方法名(参数列表)

不建议通过对象访问类方法

说明

至少有一个形参，第一个形参用于绑定类，一般命名为'cls'

使用@classmethod修饰的目的是调用类方法时可以隐式传递类

类方法中不能访问实例成员，实例方法中可以访问类成员。

作用

操作类变量

静态方法

定义

@staticmethod
def 方法名称(参数列表):
方法体

调用

类名.方法名(参数列表)

不建议通过对象访问静态方法

说明

使用@staticmethod修饰的目的是该方法不需要隐式传参数。

静态方法不能访问实例成员和类成员

作用

定义常用的工具函数。