大纲

07\_function

别将可变类型作为参数默认值

仅限关键字参数

函数返回  
1. 尽量只返回一种类型  
2. 谨慎返回None  
3. 早返回，多返回

常用函数模块：functools re

functools.partial() functools.lru\_cache() functools.wraps() functools.reduce()

re.search() re.match() re.sub()

函数与状态

全局变量 2. 闭包 3. 类

别写大复杂的函数  
1. 长度：65～200行

圈复杂度 <= 10 radon cc xx.py -s  
3. 一个函数只包含一层抽象

08\_decorator

装饰器：包装目标函数来修改其行为 --> 闭包原理

无参数：两层嵌套

装饰器

有参数：三层嵌套 <-- 可修改成可选考数

functools.wraps() 命令行工具集click

装饰时 先实例化, 再装饰

函数替换 类有括号@timer() func传入\_\_call\_\_()中

类装饰器

实例替换 类无括号@timer func传入\_\_init\_\_()中

直接装饰

wrapts库

装饰器特别适合用来实现以下功能：  
1. 运行时校验 @login\_required <-- django  
2. 注入额外参数 @patch <-- unittest.mock   
3. 缓存执行结果 @lru\_cache <-- functools

1. 注册函数 @app.route <-- flask  
   5. 替换为复杂对象 @staticmethod 描述符descriptor

09\_OOP

1. 继承 -- 里式替换原则
2. 实现 -- 实现抽象
3. 组合 -- 类实例作为属性

类实例的属性保存在\_\_dict\_\_中

@classmethod 类方法，类可以直接调用

@staticmethod 静态方法，无状态  
@property 方法变属性

鸭子类型的原限性：1. 缺乏标准 2. 进于隐式

isinstance()

抽象类的子类化机制：只关心结构，不关心真实继承关系  
抽象类通过\_\_subclasshook\_\_()钩子和.register()方法，实现了一种比继承更灵活、更松散的 子类化机制，并以此改变了isinstance()的行为

多重继承与MRO  
MRO: method resolution order，解决多重继承的方法优先级问题

mixin模式：把额外功能”混入”某个类，mixin是一种有约束的多重继承

针对事物的行为建模（而不是对事物本身建模），可以孵化出更好更灵活的模型设计

利用多态，消除条件判断语句

有序组织类方法

单例模式 vs 预邦定方法模式