

Lab06-装饰器

1 基本信息

姓名	李懋
学号	2213189
实验题目	Python 装饰器
完成时间	2024.11.1

2 实验目的

- 复习并练习 Python 装饰器

3 实验内容

- 参考资料：菜鸟教程 <https://www.runoob.com/w3cnote/python-func-decorators.html>

3.1 问答题

- 见云文档：<https://nankai.feishu.cn/docx/RTJTdY2JWoBPulxt7z3cj42Vncf>

3.2 编程题

3.2.1 基础题

- 实现一个**装饰器**，通过一次调用使函数重复执行5次

```
1 def repeat(num_times):
2     def decorator(func):
3         def wrapper(*args, **kwargs):
4             for _ in range(num_times):
5                 result = func(*args, **kwargs)
6                 return result
7         return wrapper
8     return decorator
9
10 # 使用装饰器
11 @repeat(5)
12 def foo():
13     print("this is foo")
14
15 # 调用 foo 函数
16 foo()
```

Run 实现一个装饰器 ×

D:\Applications\编程软件\Python\python.exe D:\desktop\...

this is foo
this is foo
this is foo
this is foo
this is foo

Process finished with exit code 0

- 改写后

```
1 def repeat(num_times):
2     def decorator(func):
3         def wrapper(*args, **kwargs):
4             for _ in range(num_times):
5                 result = func(*args, **kwargs)
6                 return result
7         return wrapper
8     return decorator
9
10 # 使用装饰器
11 @repeat(5)
12 def foo():
13     print("this is foo")
14
15 # 调用 foo 函数
16 foo()
```

Run 改写后 ×

D:\Applications\编程软件\Python\python.exe D:\desktop\大三上-课程资料\3-

this is foo
this is foo
this is foo
this is foo
this is foo

Process finished with exit code 0

3.2.2 进阶题

1. 在通讯录的基础上，进行扩展

1. 通过编写装饰器，为每个**功能模块函数**的每次调用设置认证。要求：

- 每次调用函数的时候需要验证用户是否登录。如果未登录要求用户进行登录，并提示用户输入用户名和密码（与文件中的信息对比），成功后调用函数；如果已经登录则直接调用函数

2. 在上述的基础上，编写装饰器，为**每个函数**添加记录调用的功能

- 每次调用函数的时候，都将调用的**函数名**和**调用时间**记录在**文本文件**当中



2. 涉及到的知识点与问题：

- 文件的读取
 - 如何将文本文件读取为其他数据结构，比如dict？可能需要结合使用高阶函数？
- 装饰器的嵌套
 - 既想要验证登录，又想要记录日记，就需要多层装饰器嵌套。这时候装饰器先运行哪一个（参考问答题第四题）？如果需要记录装饰器函数的名称需要怎么办？
 - [Python functools.wraps 详解jiang_huixin的博客-CSDN博客@functools.wraps](#)
 - 或许交换装饰器的顺序可以改善这种情况？

```
===== 联系人管理系统 =====

a: 添加联系人
d: 删除联系人
c: 修改联系人
f: 查找联系人
s: 显示所有联系人

请输入您的操作：a
请登录以继续操作。
用户名：Jack
密码：123456
欢迎回来，Jack！
请输入姓名：limao
请输入QQ号码：12345
请输入电话号码：110
请输入电子邮件：123@mail

成功添加：limao !
```

```
===== 联系人管理系统 =====

a: 添加联系人
d: 删除联系人
c: 修改联系人
f: 查找联系人
s: 显示所有联系人

请输入您的操作：c

已修改，最新的列表为

No.   Name    QQ      Phone   E-mail
1     limao   12345   110     123@mail

===== 联系人管理系统 =====

a: 添加联系人
d: 删除联系人
c: 修改联系人
f: 查找联系人
s: 显示所有联系人
```

3.2.3 附加题

- 编写下载网页内容的函数，要求：
 - 用户传入一个url，函数返回下载页面的结果（存储在html格式文件中）
 - 编写装饰器，实现缓存网页内容功能
 - 当传入url时，如果缓存文件内有值（文件大小不为0），就优先从缓存文件中读取网页的内容。否则就去下载页面，之后缓存在文件中。
- 例如：
 - 用户输入百度网址，程序发现没有下载过，于是将百度网页进行下载并保存为缓存文件“baidu.html”，并打开了这个文件。当用户再次输入百度的网址时，程序发现本地已经有“baidu.html”，则不需要再次从服务器对网页进行下载了，只需要打开这个本地文件即可。

```
D:\Applications\编程软件\Python-3.10.11\python.exe D:\desktop\大三上-课程资料\3-Python\python_programming_basics\WKU-WBH\Lab\Lab06\编程题\3-附加题.py
请输入要下载的网页 URL: https://www.icourse163.org/
Downloading from https://www.icourse163.org/
<!DOCTYPE html>
<html xmlns="//www.w3.org/1999/xhtml" xml:lang="zh" lang="zh">
<head>
<title>
中国大学MOOC_优质在线课程学习平台
</title>
<script>
Object.defineProperty(window, 'handledomooRPC', {
  enumerable: false,
  configurable: false,
  writable: false,
  value: null
});
</script>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
<link rel="shortcut icon" href="//edu-image.nosdn.127.net/32a8dd2a-b9aa-4ec9-abd5-66cd8751befb.png?imageView&quality=100" />
<meta http-equiv="x-dns-prefetch-control" content="on"/>
<meta http-equiv="Content-Security-Policy" content="upgrade-insecure-requests">
<meta itemprop="name" content="中国大学MOOC_优质在线课程学习平台"/>
<link rel="dns-prefetch" href="//s.stu.126.net"/>
<link rel="dns-prefetch" href="//icourse163.org/" />
<link rel="dns-prefetch" href="//mc.stu.126.net"/>
<link rel="dns-prefetch" href="//mooc.study.163.com"/>
```

4 实验心得

1. 多查看菜鸟教程，很完整，多上手实现代码
2. 认真做问答题，期末可能有原题（相似的）

5 源代码

- 见附件
- Github仓库