

day004装饰器-偏函数-文件读写-异常处理

概念：是一个闭包，把一个函数当做参数返回一个替代版的函数，本质上就是一个返回函数的函数

装饰器

```
def outer(func):
    def inner(*args,**kwargs):#可以传任意参数
        #添加修饰的功能
        print("&&&&&&&&&")
        func(*args,**kwargs)
    return inner

@outer
def say(name,age):
    print("my name is %s,I'm %d years old" % (name,age))

通用装饰器 say("tracy",18)
```

#函数的参数理论上是无限制的，但实际上不要超过6-7个
#say函数不允许修改，所以使用outer作为装饰器来添加一个print

偏函数

对参数上默认值的控制

```
def int2(str,base = 2):
    return int(str,base)
print(int2("1010"))
```

import functools

int3= functools.partial(int,base = 2) #把一个参数固定住，形成一个新的函数

print(int3("1111")) #partial返回一个参数，将int返回，base=2固定，用int3保存

变量的作用域

作用域：变量可以使用的范围

程序的变量并不是在所有位置都能使用的，访问的权限决定于变量的位置

局部作用域，函数作用域，内建作用域

try...except...finally

```
try:
    语句t
except 错误码 as e:
    语句1
except 错误码 as e:
    语句2
```

```
except 错误码 as e:
    语句n
finally:
    语句f
```

作用：语句t无论是否有错误都执行最后的语句f

断言

```
def func(num,div):
    assert div!= 0,'div不能为0'
    return num/div
print(func(10,0))
```

文件读写

文件打开

open(path,flag[, encoding][, errors])

r: 以只读的方式打开文件，文件的描述符放在文件的开头

rb: 以二进制格式打开一个文件用于只读，文件的描述符放在文件的开头

r+: 打开一个文件用于读写，文件的描述符放在文件的开头

w: 打开一个文件只用于写入，如果该文件已经存在会覆盖，如果不存在则创建新文件

wb: 打开一个文件只用于写入二进制，如果该文件已经存在会覆盖，如果不存在则创建新文件

w+: 打开一个文件用于读写，如果该文件已经存在会覆盖，如果不存在则创建新文件

a: 打开一个文件用于追加，如果该文件已经存在，文件描述符会放在文件末尾

a+: 打开一个文件用于读写，如果该文件已经存在，文件描述符会放在文件末尾

encoding:编码方式utf-8, jbk

errors: 错误处理

f.read() 读取全部内容

f.readline() 读取整行，包括\n字符

f.readlines() 读取所有行并返回列表

f.seek(0) 若文件描述符已经到了结尾，可修改文件描述符的位置

with open(path,"r",encoding="tuf-8") as e:

文件写

f.write(str)

f.flush 刷新缓冲区，直接把内部缓冲区的数据立刻写入文件，而不是被动的等待自动刷新缓冲区写入

#使用\n,也可以刷新缓冲区

#f.write("I also like play basketbal\n")

list, tuple, dict, set的操作

import pickle #数据持久性模块

#写操作 pickle.dump()

myList = (1,2,3,4,5,"hello master!","该")

path = r"G:\python_learn\pycharm_learn\pythonlearn\day004\write.txt"

f = open(path,"wb")

pickle.dump(myList,f)

f.close()

#读取 pickle.load()

f1 = open(path,"rb")

tempList = pickle.load(f1)

print(tempList)

f1.close

函数也是一种数据类型

f = sum #这是让f等于sum，使得f有了sum同样的功能

a=f (2,4) #就可以执行，print (a)