

day12

今日内容

- 函数中高级（闭包/高阶函数）
- 内置函数
- 内置模块（.py文件）

内容回顾

- 函数基础概念
 - 函数基本结构

```
def func(arg):  
    return arg;  
v1 = func(123)
```

- 参数
 - 写函数
 - def func(a1,a2):pass
 - def func(a1,a2=None):pass
 - def func(*args,**kwargs):pass
 - 执行函数
 - 位置参数在前/关键字参数在后。
- 函数小高级
 - 函数可以做变量

```
def func():  
    pass  
v1 = func  
v1()  
  
v2 = [func, func, func]  
v2[1]()
```

- 函数可以做参数

```
def func(arg):
    v2 = arg()

def show():
    pass

v1 = func(show)
# 注意返回值
```

- Python中以函数为作用域

```
# 第一题
for item in range(10):
    pass
print(item)

# 第二题
item = 10
def func():
    for item in range(10):
        pass
    print(item)
func()

# 第三题
item = 10
def func():
    item = 2
    def inner():
        print(item)
    for item in range(10):
        pass
    inner()
func()

# 第四题【新浪微博】
def func():
    for num in range(10):
        pass
    v4 = [lambda :num+10, lambda :num+100, lambda :num+100,]
    result1 = v4[1]()
    result2 = v4[2]()
    print(result1, result2)
func()

# 第五题【新浪微博】
def func():
    for num in range(10):
        pass
    v4 = [lambda :num+10, lambda :num+100, lambda :num+100,]
    result1 = v4[1]()
    num = 73
```

```
result2 = v4[2]()  
print(result1,result2)  
func()
```

◦ lambda表达式 (匿名函数)

```
v = [lambda x:x=100,]
```

◦ 内置

■ 输入输出

- print
- input

■ 强制转换

- int
- bool
- str
- list
- tuple
- dict
- set

■ 数学相关

- max
- min
- sum
- abs
- divmod
- float
- pow

```
v = pow(2,3)  
print(v)
```

■ round

```
v = round(1.127,2)  
print(v)
```

■ 进制

- bin
- oct
- int
- hex

- 其他
 - len
 - range
 - id
 - type
 - open

补充

1. 数据类型中的方法到底有没有返回值？

○ 无返回值

```
v = [11,22,33]
v.append(99) # 无返回值
```

○ 仅有返回值：

```
v = "alex"
result = v.split('l')

v = {'k1':'v2'}
result1 = v.get('k1')
result2 = v.keys()
```

○ 有返回+修改数据

```
v = [11,22,33]
result = v.pop()
```

○ 常用需要记住

- str
 - strip, 返回字符串
 - split, 返回列表
 - replace, 返回字符串
 - join, 返回字符串。
- list
 - append, 无
 - insert, 无
 - pop, 返回要删除的数据
 - remove, 无
 - find/index, 返回索引的位置。
- dict
 - get
 - keys
 - values
 - items

2. 函数内部的数据是否会混乱。

- 函数内部执行相互之间不会混乱
- 执行完毕 + 内部元素不被其他人使用 => 销毁

内容详细

1. 函数中高级

1.1 函数可以做返回值

```
def func():  
    print(123)  
  
def bar():  
    return func  
  
v = bar()  
  
v()
```

```
name = 'oldboy'  
def func():  
    print(name)  
  
def bar():  
    return func  
  
v = bar()  
  
v()
```

```
def bar():  
    def inner():  
        print(123)  
    return inner  
v = bar()  
v()
```

```
name = 'oldboy'  
def bar():  
    name = 'alex'  
    def inner():  
        print(name)  
    return inner  
v = bar()  
v()
```

```

name = 'oldboy'
def bar(name):
    def inner():
        print(name)
    return inner
v1 = bar('alex') # { name=alex, inner } # 闭包, 为函数创建一块区域 (内部变量供自己使用), 为他以后执行提供数据。
v2 = bar('eric') # { name=eric, inner }
v1()
v2()

```

练习题

```

# 第一题
name = 'alex'
def base():
    print(name)

def func():
    name = 'eric'
    base()

func() # {name=eric, }

```

```

# 第二题
name = 'alex'

def func():
    name = 'eric'
    def base():
        print(name)
    base()
func()

```

```

# 第三题
name = 'alex'

def func():
    name = 'eric'
    def base():
        print(name)
    return base
base = func()
base()

```

注意：函数在何时被谁创建？面试题

```
info = []

def func():
    print(item)

for item in range(10):
    info.append(func)

info[0]()
```

```
info = []

def func(i):
    def inner():
        print(i)
    return inner

for item in range(10):
    info.append(func(item))

info[0]()
info[1]()
info[4]()
```

1.2 闭包

```
def func(name):
    def inner():
        print(name)
    return inner

v1 = func('alex')
v1()
v2 = func('eric')
v2()
```

1.3 高阶函数

- 把函数当作参数传递
- 把函数当作返回值

注意：对函数进行赋值

1.4 总结

- 函数执行的流程分析（函数到底是谁创建的？）
- 闭包概念：为函数创建一块区域并为其维护自己数据，以后执行时方便调用。【应用场景：装饰器 / SQLAlchemy源码】

2. 内置函数

- 编码相关

- chr, 将十进制数字转换成 unicode 编码中的对应字符串。

```
v = chr(99)
print(v)
```

- ord, 根据字符在unicode编码中找到其对应的十进制。

```
num = ord('中')
```

- 应用:

```
import random

def get_random_code(length=6):
    data = []
    for i in range(length):
        v = random.randint(65,90)
        data.append(chr(v))

    return ''.join(data)

code = get_random_code()
print(code)
```

```
import random # 导入一个模块

v = random.randint(起始,终止) # 得到一个随机数
```

- 高级一点的内置函数

- map, 循环每个元素 (第二个参数), 然后让每个元素执行函数 (第一个参数), 将每个函数执行的结果保存到新的列表中, 并返回。

```
v1 = [11,22,33,44]
result = map(lambda x:x+100,v1)
print(list(result)) # 特殊
```

- filter

```
v1 = [11,22,33,'asd',44,'xf']

def func(x):
    if type(x) == int:
        return True
    return False
result = filter(func,v1) # [11,]
print(list(result))
```



```
result = filter(lambda x: True if type(x) == int else False ,v1)
print(list(result))

result = filter(lambda x: type(x) == int ,v1)
print(list(result))
```

- reduce

```
import functools
v1 = ['wo', 'hao', 'e']

def func(x,y):
    return x+y
result = functools.reduce(func,v1)
print(result)

result = functools.reduce(lambda x,y:x+y,v1)
print(result)
```

- 面试题：
 - 常用的内置函数有哪些？
 - filter/map/reduce是什么意思？
 - 什么是匿名函数？

```
def func():
    pass

v = [lambda x:x+100,]
```

3. 模块

将指定的“字符串”进行加密。

```
import hashlib

def get_md5(data):
    obj = hashlib.md5()
    obj.update(data.encode('utf-8'))
    result = obj.hexdigest()
    return result

val = get_md5('123')
print(val)
```

加盐

```
import hashlib

def get_md5(data):
    obj = hashlib.md5("sidrsicxwersdfsaersdfsfresdy54436jgfsjdxff123ad".encode('utf-8'))
    obj.update(data.encode('utf-8'))
    result = obj.hexdigest()
    return result

val = get_md5('123')
print(val)
```

应用:

```
import hashlib
USER_LIST = []

def get_md5(data):
    obj = hashlib.md5("12:;idrsicxwersdfsaersdfsfresdy54436jgfsjdxff123ad".encode('utf-8'))
    obj.update(data.encode('utf-8'))
    result = obj.hexdigest()
    return result

def register():
    print('*****用户注册*****')
    while True:
        user = input('请输入用户名:')
        if user == 'N':
            return
        pwd = input('请输入密码:')
        temp = {'username':user, 'password':get_md5(pwd)}
        USER_LIST.append(temp)

def login():
    print('*****用户登陆*****')
    user = input('请输入用户名:')
    pwd = input('请输入密码:')

    for item in USER_LIST:
        if item['username'] == user and item['password'] == get_md5(pwd):
            return True

register()
result = login()
if result:
    print('登陆成功')
else:
    print('登陆失败')
```

赠送：密码不显示（只能在终端运行）

```
import getpass

pwd = getpass.getpass('请输入密码：')
if pwd == '123':
    print('输入正确')
```

总结

- 自定义函数
 - 基本函数结构（98%）
 - 高级
 - 参数
 - 闭包
- 内置函数
- 模块
 - random
 - hashlib
 - getpass

本周作业

1. 学习笔记：md文件
2. 思维导图：png文件
3. 本周每天的作业（考试题）
4. 今天作业（码云）

