day16

1. 作业管理

```
第一行：10/25
按照自己情况进行说明：作业题 / 讲课内容。
```

2.今日内容

- 模块基础知识
- time/datetime
- json/picle
- shutil
- logging
- 其他

3.内容回顾和补充

3.1模块（类库）

- 内置
- 第三方
- 自定义

面试题：

- 列举常用内置模块：json / time / os/ sys

3.2 定义模块

定义模块时可以把一个py文件或一个文件夹（包）当作一个模块，以方便于以后其他py文件的调用。

对于包的定义：

- py2：文件见中必须有 _ _init _ _.py 。
- py3：不需要 _ *init* _.py .

推荐大家以后写代码时，都要加上此文件。

3.3 模块的调用

3.3.1 示例一

```python
# lizhongwei.py


#!/usr/bin/env python
# -*- coding:utf-8 -*-

def show():
    print('我司里种种')

def func():
    pass

print(456)
```

```python

# 导入模块，加载此模块中所有的值到内存。
import lizhongwei

print(123)
# 调用模块中的函数
lizhongwei.func()
```

```python
# 导入模块
from lizhongwei import func,show
from lizhongwei import func
from lizhongwei import show
from lizhongwei import *

func()
```

```python
# 导入模块
from lizhongwei import func as f

def func():
    print(123)
f()
```

导入模块：

- import 模块 模块.函数()
- from 模块 import 函数 函数() 【as起别名 / *】
- from 模块 import 函数 as 别名 别名()

3.3.2 示例二

```
lizohng
    - jd.py
    - pdd.py
    - tb.py
包.py
```

```
import lizhong.jd
lizhong.jd.f1()
```

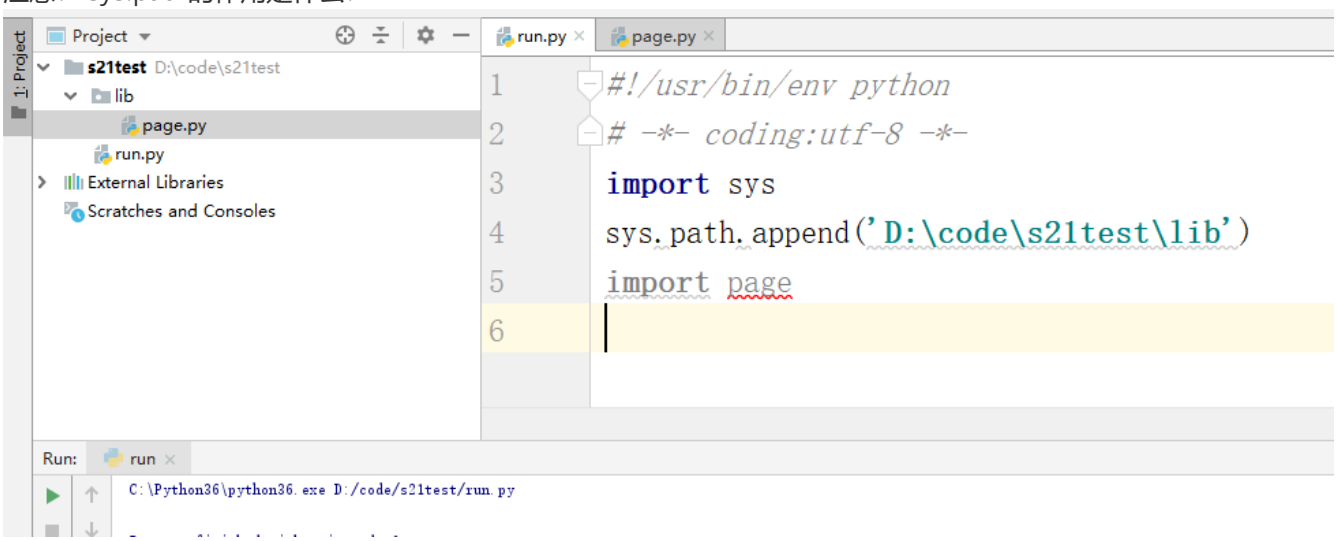
```
from lizhong import jd
jd.f1()
```

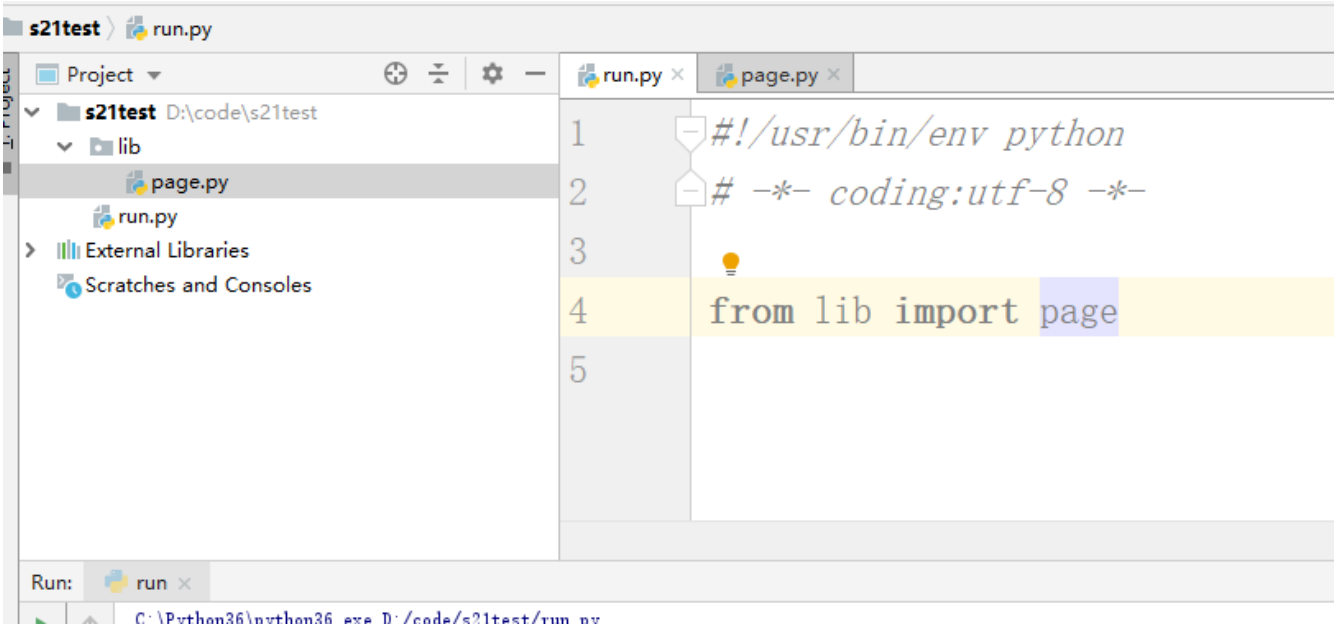
```
from lizhong.jd import f1
f1()
```

总结

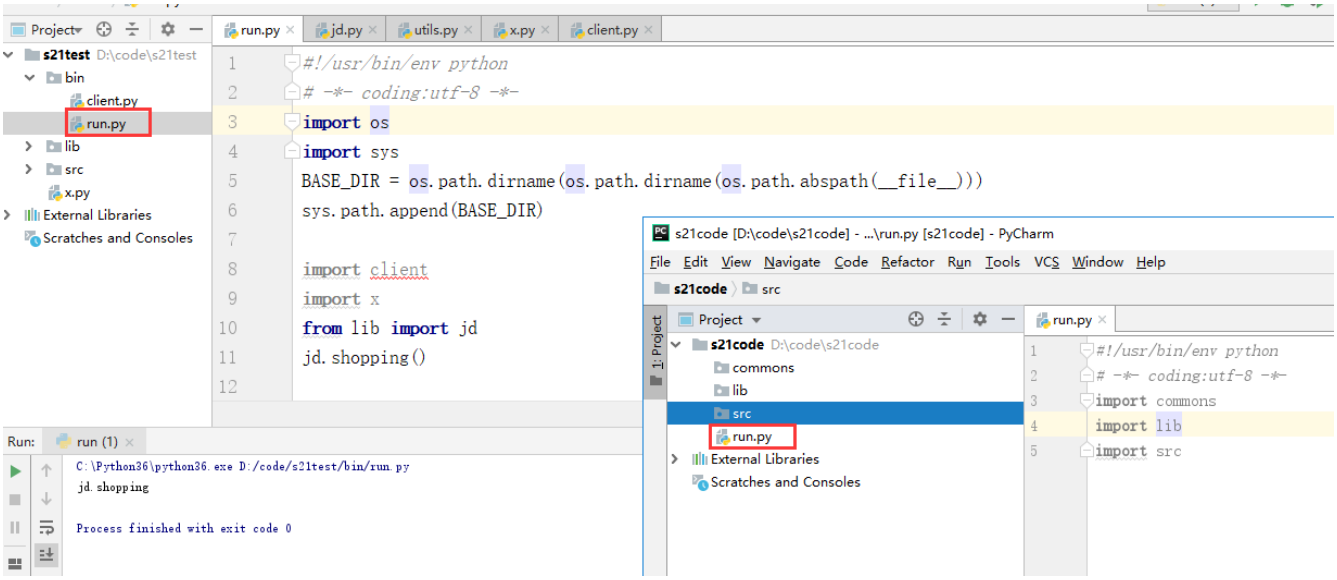
- 模块和要执行的py文件在同一目录 且 需要 模块中的很多功能时，推荐用： import 模块
- 其他推荐：from 模块 import 模块 模块.函数()
- 其他推荐：from 模块.模块 import 函数 函数()

注意： sys.path的作用是什么?





练习题



3.4 内置模块

- os
- sys
- time
- json
  - dumps
  - loads
  - 注意:
    - 字典或列表中如有中文，序列化时想要保留中文显示:

```
v = {'k1':'alex','k2':'李杰'}

import json
val = json.dumps(v,ensure_ascii=False)
print(val)
```

- dump

```
import json

v = {'k1':'alex','k2':'李杰'}

f = open('x.txt',mode='w',encoding='utf-8')
val = json.dump(v,f)
print(val)
f.close()
```

- load

```
import json

v = {'k1':'alex','k2':'李杰'}

f = open('x.txt',mode='r',encoding='utf-8')

data = json.load(f)
f.close()

print(data,type(data))
```

- haslib
- random
- getpass
- shutil
- copy

4. 今日内容

4.1 json和pickle

- json，优点：所有语言通用；缺点：只能序列化基本的数据类型 list/dict/int...
- pickle，优点：python中所有的东西都能被他序列化（socket对象）；缺点：序列化的内容只有python认识。

```
import pickle

# ################### dumps/loads #####################
"""
v = {1,2,3,4}
```

```
val = pickle.dumps(v)
print(val)
data = pickle.loads(val)
print(data,type(data))
"""

"""
def f1():
    print('f1')

v1 = pickle.dumps(f1)
print(v1)
v2 = pickle.loads(v1)
v2()
"""

# ################### dump/load ######################
# v = {1,2,3,4}
# f = open('x.txt',mode='wb')
# val = pickle.dump(v,f)
# f.close()

# f = open('x.txt',mode='rb')
# data = pickle.load(f)
# f.close()
# print(data)
```

4.2 shutil 模块

```
import shutil

# 删除目录
# shutil.rmtree('test')

# 重命名
# shutil.move('test','ttt')

# 压缩文件
# shutil.make_archive('zzh','zip','D:\code\s21day16\lizhong')

# 解压文件
# shutil.unpack_archive('zzh.zip',extract_dir=r'D:\code\xxxxxx\xxxx',format='zip')
```

示例

```
import os
import shutil
from datetime import datetime
ctime = datetime.now().strftime('%Y-%m-%d-%H-%M-%S')

# 1.压缩lizhongwei文件夹 zip
# 2.放到到 code 目录（默认不存在）
```

```
# 3.将文件解压到D:\x1目录中。

if not os.path.exists('code'):
    os.makedirs('code')
shutil.make_archive(os.path.join('code',ctime),'zip','D:\code\s21day16\lizhongwei')

file_path = os.path.join('code',ctime) + '.zip'
shutil.unpack_archive(file_path,r'D:\x1','zip')
```

4.3 time&datetime

> UTC/GMT：世界时间
>
> 本地时间：本地时区的时间。

4.3.1 time模块

- time.time()，时间戳：1970-1-1 00:00
- time.sleep(10)，等待秒数。
- time.timezone

```
# https://login.wx.qq.com/cgi-bin/mmwebwx-bin/login?
loginicon=true&uuid=4ZwIFHM6iw==&tip=1&r=-781028520&_=1555559189206
```

4.3.2 datetime模块

```
#!/usr/bin/env python
# -*- coding:utf-8 -*-
import time
from datetime import datetime,timezone,timedelta

# ######################### 获取datetime格式时间 ##############################
"""
v1 = datetime.now() # 当前本地时间
print(v1)
tz = timezone(timedelta(hours=7)) # 当前东7区时间
v2 = datetime.now(tz)
print(v2)
v3 = datetime.utcnow() # 当前UTC时间
print(v3)
"""

# ######################### 把datetime格式转换成字符串 ##############################
# v1 = datetime.now()
# print(v1,type(v1))
# val = v1.strftime("%Y-%m-%d %H:%M:%S")
# print(val)

# ######################### 字符串转成datetime ##############################
# v1 = datetime.strptime('2011-11-11','%Y-%m-%d')
# print(v1,type(v1))
```

```
# ######################### datetime时间的加减 ##############################
# v1 = datetime.strptime('2011-11-11','%Y-%m-%d')
# v2 = v1 - timedelta(days=140)
# date = v2.strftime('%Y-%m-%d')
# print(date)

# ######################### 时间戳和datetime关系 ##############################
# ctime = time.time()
# print(ctime)
# v1 = datetime.fromtimestamp(ctime)
# print(v1)

# v1 = datetime.now()
# val = v1.timestamp()
# print(val)
```

4.4 logging [欠]

4.5 异常处理

```
try:
    val = input('请输入数字:')
    num = int(val)
except Exception as e:
    print('操作异常')
```

```
# import requests
#
# try:
#     ret = requests.get('http://www.google.com')
#     print(ret.text)
# except Exception as e:
#     print('请求异常')
```

```
def func(a):
    try:
        return a.strip()
    except Exception as e:
        pass
    return False

v = func('alex')
if not v:
    print('函数执行失败')
else:
    print('结果是',v)
```

练习题

```python
# 1. 写函数，函数接受一个列表，请将列表中的元素每个都 +100
def func(arg):
    result = []
    for item in arg:
        if item.isdecimal():
            result.append(int(item) + 100)
    return result

# 2. 写函数去，接受一个列表。列表中都是url，请访问每个地址并获取结果。
import requests
def func(url_list):
    result = []
    try:
        for url in url_list:
            response = requests.get(url)
            result.append(response.text)
    except Exception as e:
        pass
    return result

def func2(url_list):
    result = []
    for url in url_list:
        try:
            response = requests.get(url)
            result.append(response.text)
        except Exception as e:
            pass
    return result

func(['http://www.baidu.com','http://www.google.com','http://www.bing.com'])
```

总结

函数高级 5*

- 嵌套
- 装饰器

模块分类和定义 4*

- 内置模块
  - os
  - sys
  - ...
- 第三方
  - requests
  - xlrd
- 自定义模块
  - 文件

    - 文件夹 + init.py【包】

导入模块

- sys.path
- 导入
    - import
    - from xx.xxx import xx

异常处理

```
try:
    pass
except Exception as e:
    pass
```