requests模块的学习

使用事前

- pip install reqeusts

发送get，post请求，获取响应

- response = requests.get(url) #发送get请求，请求url地址对应的响应
- response = reqeusts.post(url,data={请求体的字典}) ##发送post请求

response的方法

- response.text
  - 该方式往往会出现乱码，出现乱码使用response.encoding="utf-8"
- response.content.decode()
  - 把响应的二进制字节流转化为str类型
- resposne.request.url #发送请求的url地址
- response.url #response响应的url地址
- resposne.request.headers #请求头
- resposne.headers #响应请求

获取网页源码的正确打开方式(通过下面三种方式一定能够获取到网页的正确解码之后的字符串)

- 1. response.content.decode()
- 2. response.content.decode("gbk")
- 3. response.text

发送带header的请求

- 为了模拟浏览器，获取和浏览器一模一样的内容

```
headers = {
"User-Agent":"Mozilla/5.0 (iPhone; CPU iPhone OS 10_3 like Ma
"Referer": "http://fanyi.baidu.com/?aldtype=16047"}

response = requests.get(url,headers=headers)
```

使用超时参数

- requests.get(url,headers=headers,timeout=3) #3秒内必须返回响应，否则会报错

retrying模块的学习

- pip install retrying

```
from retrying import retry

@retry(stop_max_attempt_number=3)
def fun1():
  print("this is func1")
  raise ValueError("this is test error")
```

处理cookie相关的请求

- 人人网{"email":"mr_mao_hacker@163.com", "password":"alarmchime"}
- 直接携带cookie请求url地址
    - 1. cookie放在headers中

```
headers= {"User-Agent":"....","Cookie":"cookie 字符串"
```

    - 2. cookie字典传给cookies参数
        - requests.get(url,cookies=cookie_dict)
- 先发送post请求，获取cookie，带上cookie请求登录后的页面
    - 1. seesion = requests.session()#session具有的方法和requests一样
    - 2. session.post(url,data,headers) #服务器设置在本地的cookie会笨哦存在session
    - 3. session.get(url) #会带上之前保存在session中的cookie,能够请求成功