Requests库的7个主要方法

方法	说明
requests.request()	构造一个请求，支撑以下各方法的基础方法
requests.get()	获取HTML网页的主要方法，对应于HTTP的GET
requests.head()	获取HTML网页头信息的方法，对应于HTTP的HEAD
requests.post()	向HTML网页提交POST请求的方法，对应于HTTP的POST
requests.put()	向HTML网页提交PUT请求的方法，对应于HTTP的PUT
requests.patch()	向HTML网页提交局部修改请求，对应于HTTP的PATCH
requests.delete()	向HTML页面提交删除请求，对应于HTTP的DELETE

Response对象的属性（1）

属性	说明
r.status_code	HTTP请求的返回状态，200表示连接成功，404表示失败
r.text	HTTP响应内容的字符串形式，即，url对应的页面内容
r.encoding	从HTTP header中猜测的响应内容编码方式
r.apparent_encoding	从内容中分析出的响应内容编码方式（备选编码方式）
r.content	HTTP响应内容的二进制形式

```
>>> r = requests.get("http://www.baidu.com")
>>> r.status_code
200
>>> r.text
'<!DOCTYPE html>\r\n<!--STATUS OK--><html> <head><meta http-equiv=content-type c
ontent=text/html;charset=utf-8><meta http-equiv=X-UA-Compatible content=IE=Edge>
<meta content=always name=referrer><link rel=stylesheet type=text/css href=http:
//s1.bdstatic.com/r/www/cache/bdorz/baidu.min.css><title>ç\x99¾å°¦ä¸\x80ä¸\x8bi¼
>>> r.encoding
'ISO-8859-1'
>>> r.apparent_encoding
'utf-8'
>>> r.encoding = "utf-8"
>>> r.text
'<!DOCTYPE html>\r\n<!--STATUS OK--><html> <head><meta http-equiv=content-type c
ontent=text/html;charset=utf-8><meta http-equiv=X-UA-Compatible content=IE=Edge>
<meta content=always name=referrer><link rel=stylesheet type=text/css href=http:
//s1.bdstatic.com/r/www/cache/bdorz/baidu.min.css><title>百度一下，你就知道</titl
```

理解Response的编码

r.encoding	从HTTP header中猜测的响应内容编码方式
r.apparent_encoding	从内容中分析出的响应内容编码方式（备选编码方式）

r.encoding：如果header中不存在charset，则认为编码为ISO-8859-1

r.text根据r.encoding显示网页内容

r.apparent_encoding：根据网页内容分析出的编码方式

可以看作是r.encoding的备选

理解Requests库的异常

异常	说明
requests.ConnectionError	网络连接错误异常，如DNS查询失败、拒绝连接等
requests.HTTPError	HTTP错误异常
requests.URLRequired	URL缺失异常
requests.TooManyRedirects	超过最大重定向次数，产生重定向异常
requests.ConnectTimeout	连接远程服务器超时异常
requests.Timeout	请求URL超时，产生超时异常

理解Response的异常

r.raise_for_status()	如果不是200，产生异常 requests.HTTPError

```
r = requests.get(url)
```

r.raise_for_status()在方法内部判断r.status_code是否等于200，不需要增加额外的if语句，该语句便于利用try-except进行异常处理

爬取网页的通用代码框架

```
import requests

def getHTMLText(url):
    try:
        r = requests.get(url, timeout=30)
        r.raise_for_status() #如果状态不是200，引发HTTPError异常
        r.encoding = r.apparent_encoding
        return r.text
    except:
        return "产生异常"

if __name__ == "__main__":
    url = "http://www.baidu.com"
    print(getHTMLText(url))
```

爬取网页的通用代码框架

```
if __name__ == "__main__":
    url = "http://www.baidu.com"
    print(getHTMLText(url))
```

```
>>>
<!DOCTYPE html>
<!--STATUS OK--><html> <head><meta http-equiv=content-ty
rset=utf-8><meta http-equiv=X-UA-Compatible content=IE=E
 name=referrer><link rel=stylesheet type=text/css href=h
www/cache/bdorz/baidu.min.css><title>百度一下，你就知道</t
=#0000cc> <div id=wrapper> <div id=head> <div class=head
```

```
if __name__ == "__main__":
    url = "www.baidu.com"
    print(getHTMLText(url))
```

```
>>>
产生异常
```

如果出现 url 异常和网络异常，则输出产生异常