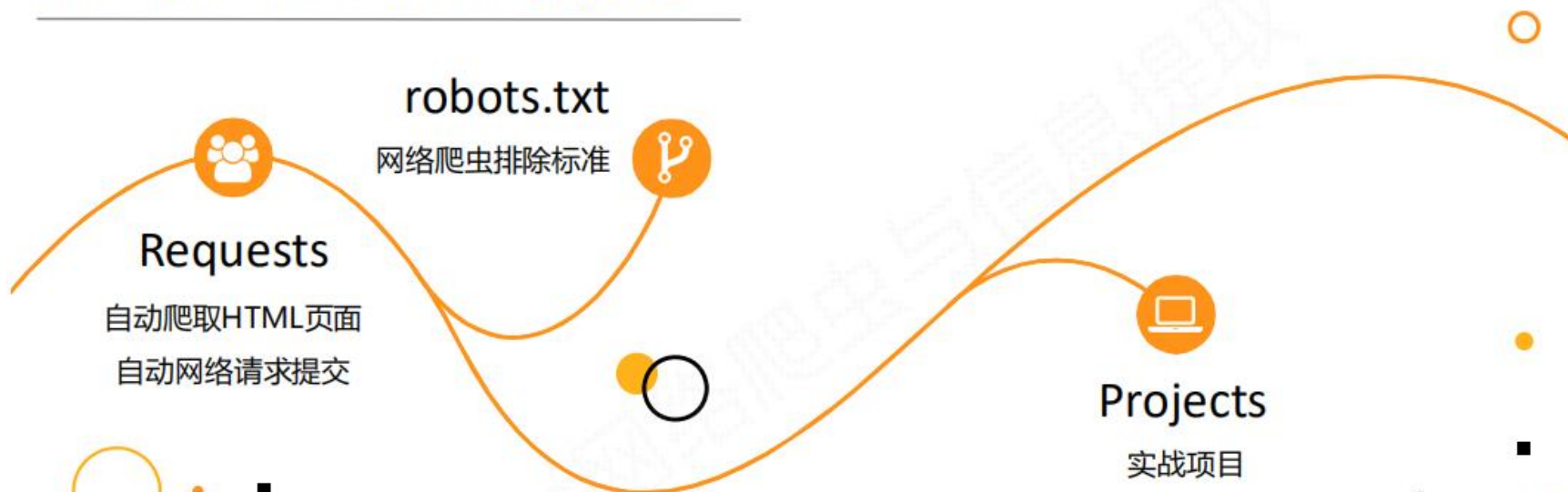


The Website is the API ...



掌握定向网络数据爬取和网页解析的基本能力

python
弹指之间 · 享受创新

Python网络爬虫与信息提取

04X -Tian

Requests库入门

The Website is the API ...



Requests

自动爬取HTML页面

自动网络请求提交

掌握定向网络数据爬取和网页解析的基本能力

Python网络爬虫与信息提取

python
弹指之间·享受创新



Requests库的安装



Star 22,840

Requests is an elegant and simple HTTP library for Python, built for human beings.

Stay Informed

Receive updates on new releases and upcoming projects.

Follow @kennethreitz

Requests: HTTP for Humans

Release v2.12.4. ([Installation](#))

<http://www.python-requests.org>

Requests is the only Non-GMO HTTP library for Python, safe for human consumption.

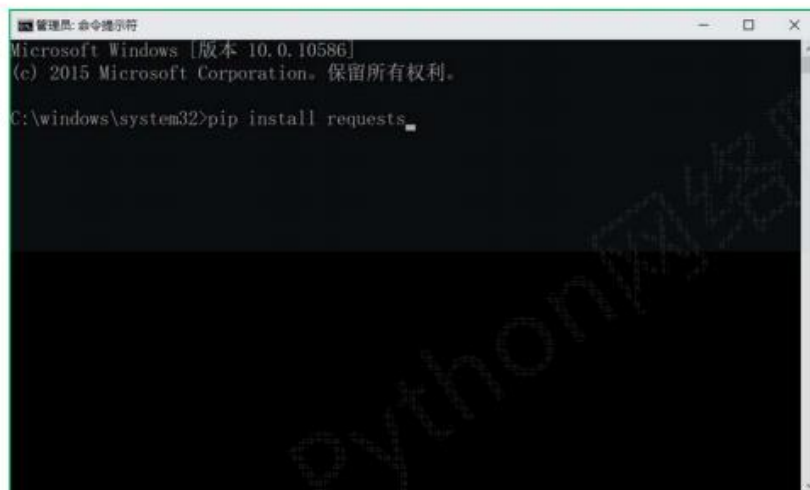
Warning: Recreational use of other HTTP libraries may result in dangerous side-effects, including: security vulnerabilities, verbose code, reinventing the wheel, constantly reading documentation, depression, headaches, or even death.

Behold, the power of Requests:

```
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
u'{"type": "User"...}'
>>> r.json()
{'u'private_gists': 419, u'total_private_repos': 77, ...}
```

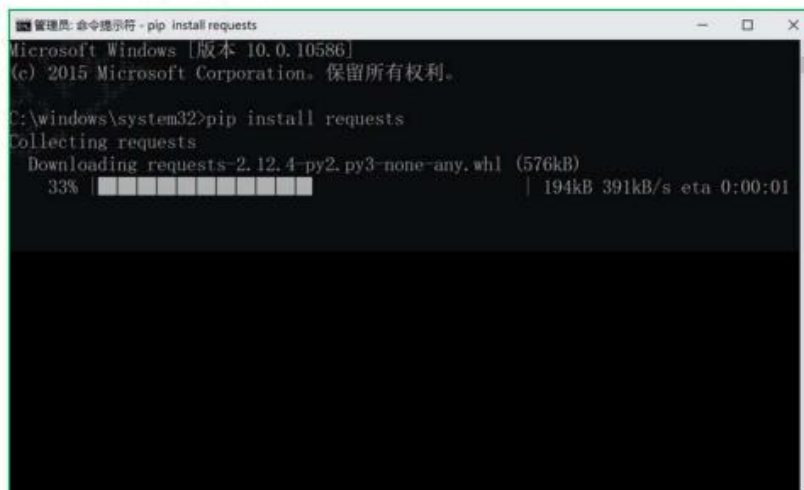
Requests库的安装

Win平台：“以管理员身份运行” cmd , 执行 `pip install requests`



```
管理员: 命令提示符
Microsoft Windows [版本 10.0.10586]
(c) 2015 Microsoft Corporation。保留所有权利。

C:\windows\system32>pip install requests_
```



```
管理员: 命令提示符 - pip install requests
Microsoft Windows [版本 10.0.10586]
(c) 2015 Microsoft Corporation。保留所有权利。


C:\windows\system32>pip install requests
Collecting requests
  Downloading requests-2.12.4-py2.py3-none-any.whl (576kB)
    33% |#####| 194kB 391kB/s eta 0:00:01
```

Requests库的安装小测

```
>>> import requests
>>> r = requests.get("http://www.baidu.com")
>>> print(r.status_code)
200
>>> r.text
'<!DOCTYPE html>\r\n<!--STATUS OK--><html> <head><meta http-equiv
=text/html; charset=utf-8><meta http-equiv=X-UA-Compatible content
t=always name=referrer><link rel=stylesheet type=text/css href=ht
r/www/cache/bdorz/baidu.min.css><title>ç\x99%â°|ä.\x80ä.\x8bï¼\x8
\x93</title></head> <body link=#0000cc> <div id=wrapper> <div id=
>>> import requests
>>> url = "http://www.baidu.com"
>>> r = requests.get(url)
>>> print(r.status_code)
200
>>> r.text
>>> type(r)
<class 'requests.models.Response'>
```


Requests库的7个主要方法

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



Requests库的get()方法

requests.get()

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

返回一个包含服务器
资源的Response对象

Response

构造一个向服务器请求
资源的Request对象

Request

```
requests.get(url, params=None, **kwargs)
```

- **url** : 拟获取页面的url链接
- **params** : url中的额外参数，字典或字节流格式，可选
- ****kwargs**: 12个控制访问的参数

requests.get(url, params=None, **kwargs)

```
def get(url, params=None, **kwargs):  
    """Sends a GET request.  
  
    :param url: URL for the new :class:`Request` object.  
    :param params: (optional) Dictionary or bytes to be sent in the query string for the  
    :param \**kwargs: Optional arguments that ``request`` takes.  
    :return: :class:`Response <Response>` object  
    :rtype: requests.Response  
    """  
  
    kwargs.setdefault('allow_redirects', True)  
    return request('get', url, params=params, **kwargs)
```

Requests库的2个重要对象

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

Response

Request

Response对象包含爬虫返回的内容

Response对象

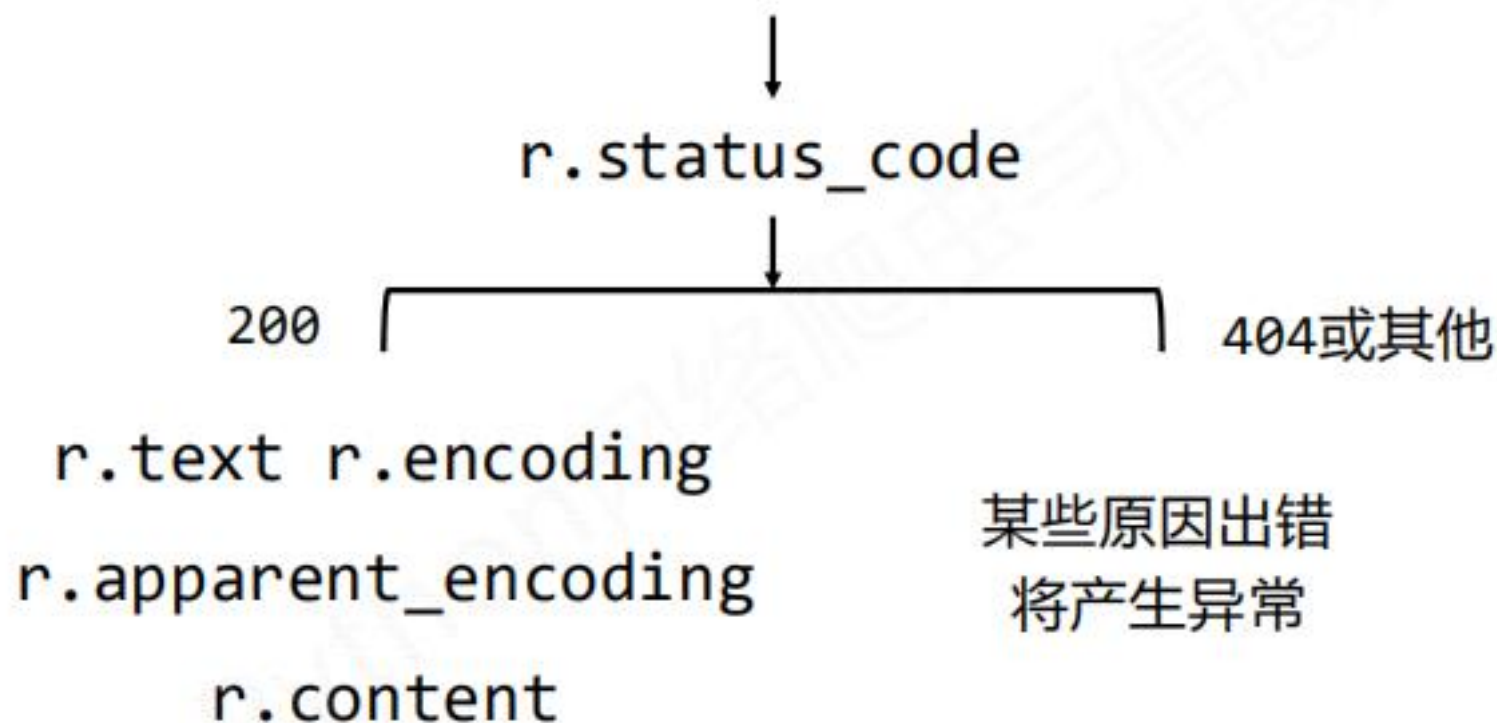
```
>>> import requests
>>> r = requests.get("http://www.baidu.com")
>>> print(r.status_code)
200
>>> type(r)
<class 'requests.models.Response'>
>>> r.headers
{'Cache-Control': 'private, no-cache, no-store, proxy-revalidate,
ection': 'Keep-Alive', 'Transfer-Encoding': 'chunked', 'Server':
```

Response对象包含服务器返回的所有信息，也包含请求的Request信息

Response对象的属性（1）

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

Response对象的属性



```
>>> 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\x99%å°|ä,\x80ä,\x8bï¼
>>> 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

<code>r.encoding</code>	从HTTP header中猜测的响应内容编码方式
<code>r.apparent_encoding</code>	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

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



爬取网页的通用代码框架

理解Requests库的异常

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



Exception

网络连接有风险，异常处理很重要

理解Requests库的异常

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

理解Response的异常

<code>r.raise_for_status()</code>	如果不是200，产生异常 <code>requests.HTTPError</code>
-----------------------------------	--

```
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))
```

```
if __name__ == "__main__":  
    url = "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
```

```
>>>  
产生异常
```

The Website is the API ...



Requests

自动爬取HTML页面
自动网络请求提交

robots.txt

网络爬虫排除标准



掌握定向网络数据爬取和网页解析的基本能力

Python网络爬虫与信息提取

python
弹指之间 · 享受创新

04X -Tian



网络爬虫的尺寸

小规模，数据量小

爬取速度不敏感

Requests库

>90%

爬取网页 玩转网页

中规模，数据规模较大

爬取速度敏感

Scrapy库

爬取网站 爬取系列网站

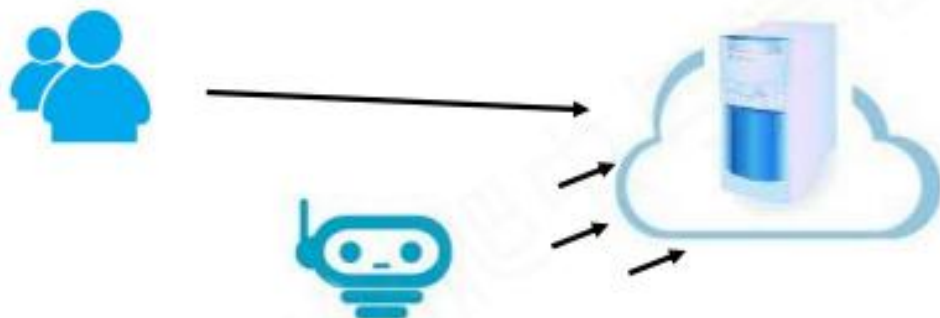
大规模，搜索引擎

爬取速度关键

定制开发

爬取全网

网络爬虫的“性能骚扰”



Web服务器默认接收人类访问

受限于编写水平和目的，网络爬虫将会为Web服务器带来巨大的资源开销

网络爬虫的法律风险



服务器上的数据有产权归属

网络爬虫获取数据后牟利将带来法律风险

网络爬虫的隐私泄露



网络爬虫可能具备突破简单访问控制的能力，获得被保护数据
从而泄露个人隐私

网络爬虫引发的问题

性能骚扰

法律风险

隐私泄露



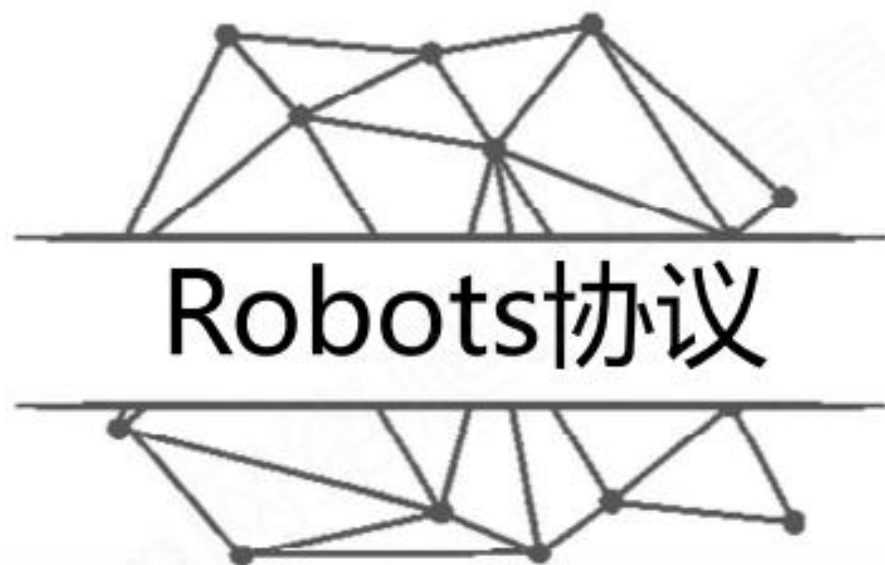
网络爬虫的限制

- 来源审查：判断User-Agent进行限制

检查来访HTTP协议头的User-Agent域，只响应浏览器或友好爬虫的访问

- 发布公告：Robots协议

告知所有爬虫网站的爬取策略，要求爬虫遵守



Robots协议

Robots协议

Robots Exclusion Standard , 网络爬虫排除标准

作用：

网站告知网络爬虫哪些页面可以抓取，哪些不行

形式：

在网站根目录下的robots.txt文件

案例：京东的Robots协议

<https://www.jd.com/robots.txt>

```
User-agent: *  
Disallow: /?*  
Disallow: /pop/*.html  
Disallow: /pinpai/*.html?*  
User-agent: EtaoSpider  
Disallow: /  
User-agent: HuihuiSpider  
Disallow: /  
User-agent: GwdangSpider  
Disallow: /  
User-agent: WochachaSpider  
Disallow: /
```

注释，*代表所有，/代表根目录

```
User-agent: *  
Disallow: /
```

Robots协议基本语法

- **#***代表所有,**/**代表根目录
- **User-agent:* #user-agent**代表来源
- **Allow:/ #**代表运行爬取的内容
- **Disallow:/ #**代表不可爬取的目录,如果是/
后面没有写内容,便是其对应的访问者不可爬取所有内容

案例：真实的Robots协议

<http://www.baidu.com/robots.txt>

<http://news.sina.com.cn/robots.txt>

<http://www.qq.com/robots.txt>

<http://news.qq.com/robots.txt>

<http://www.moe.edu.cn/robots.txt> （无robots协议）

<https://www.baidu.com/robots.txt>



Robots协议的遵守方式

Robots协议的使用

网络爬虫：

自动或人工识别robots.txt，再进行内容爬取

约束性：

Robots协议是建议但非约束性，网络爬虫可以不遵守，但存在法律风险

对Robots协议的理解

访问量很小：可以遵守

访问量较大：建议遵守

非商业且偶尔：建议遵守

商业利益：必须遵守

必须遵守

爬取网页 玩转网页

爬取网站 爬取系列网站

爬取全网

原则：类人行为可不参考Robots协议



单元小结

网络爬虫 “盗亦有道”

注释，*代表所有，/代表根目录

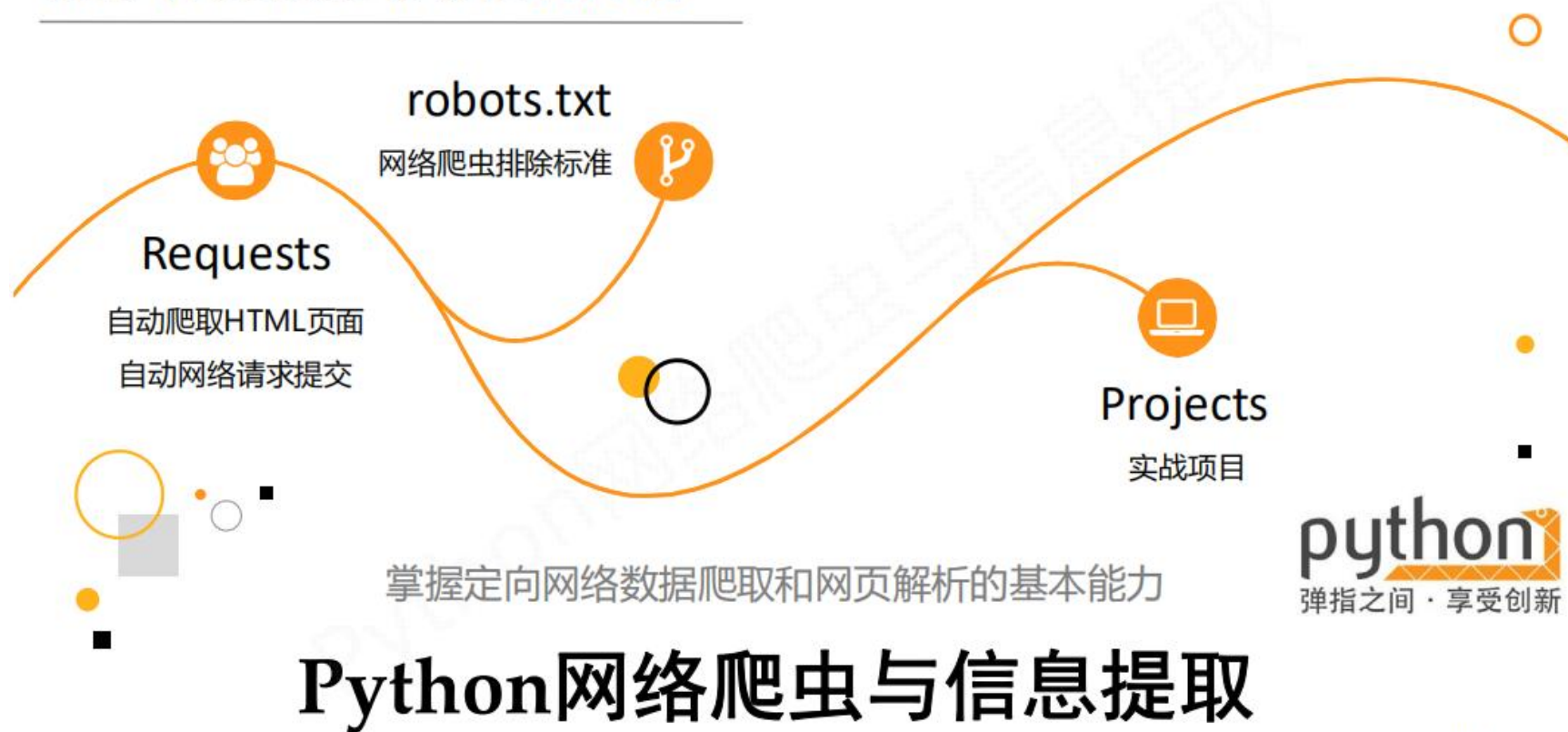
User-agent: *

Disallow: /

Robots协议基本语法

Robots协议的使用原则

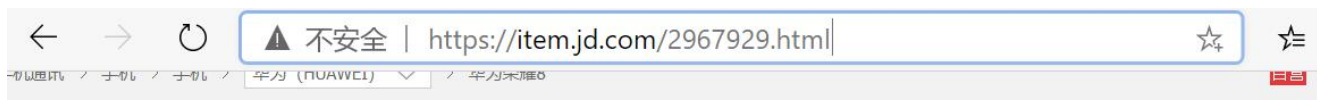
The Website is the API ...





实例1：京东商品页面的爬取

<https://item.jd.com/2967929.html>



关注 分享 对比

举报

荣耀8 4GB+64GB 全网通4G手机 魅海蓝

该商品已下柜，欢迎挑选其他商品！

相似商品推荐



Apple iPhone 12 (A2404) 128GB
绿色 支持移动联通电信5G 双卡双
¥6799.00



Apple iPhone 12 Pro Max (A2412)
256GB 海蓝色 支持移动联通电信
¥10099.00



Apple iPhone 12 Pro (A2408)
128GB 海蓝色 支持移动联通电信
¥8499.00



Redmi Note 9 Pro 5G 一亿像素 骁龙750G 33W快充 120Hz刷新率 静
¥1999.00

```
>>> import requests
```

```
>>> url = "https://item.jd.com/100016034380.html"
```

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

```
>>> print(r.status_code)
```

```
200
```

```
>>> r.encoding
```

```
'UTF-8'
```

```
>>> r.text[:1000]
```

```
"<script>window.location.href='https://passport.jd.com/new/login.aspx?
ReturnUrl=http%3A%2F%2Fitem.jd.com%2F100016034380.html'</sc
ript>"
```

```
>>> r.encoding=r.apparent_encoding
```

```
>>> r.text[:1000]
```

全代码

```
import requests
url = "https://item.jd.com/2967929.html"
try:
    r = requests.get(url)
    r.raise_for_status()
    r.encoding = r.apparent_encoding
    print(r.text[:1000])
except:
    print("爬取失败")
```



实例3：百度/360搜索关键词提交



<http://www.baidu.com>

百度一下



<http://www.so.com>

搜一下

搜索引擎关键词提交接口

百度的关键词接口：

`http://www.baidu.com/s?wd=keyword`

360的关键词接口：

`http://www.so.com/s?q=keyword`

```
>>> import requests
>>> kv = {'wd': 'Python'}
>>> r = requests.get("http://www.baidu.com/s", params=kv)
>>> r.status_code
200
>>> r.request.url
'http://www.baidu.com/s?wd=Python'
>>> len(r.text)
302829
```

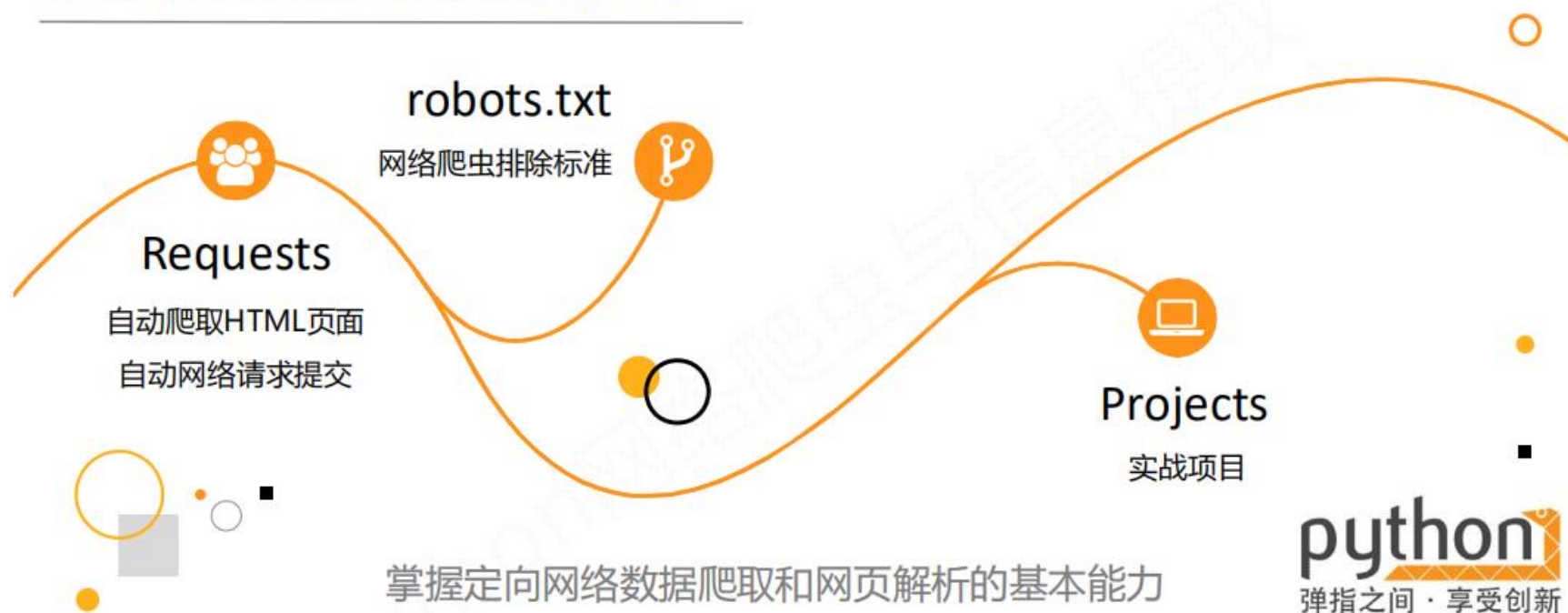

百度搜索全代码

```
import requests
keyword = "Python"
try:
    kv = {'wd':keyword}
    r = requests.get("http://www.baidu.com/s",params=kv)
    print(r.request.url)
    r.raise_for_status()
    print(len(r.text))
except:
    print("爬取失败")
```

```
>>> import requests
>>> kv = {'q': 'Python'}
>>> r = requests.get('http://www.so.com/s', params=kv)
>>> r.status_code
200
>>> r.request.url
'https://www.so.com/s?q=Python'
>>> len(r.text)
228253
```



The Website is the API ...



Python网络爬虫与信息提取

04X -Tian