

# requests 是什么？

[人类唯一的一个非转基因的 Python HTTP 库，人类可以安全享用。](#)

## 安装与调用 requests

```
1 | pip install requests # 在 CMD 中运行
2 | import requests      # 在 python 编辑器中调用
```

## requests 库的七个常用用法

| 方法                  | 说明                                     |
|---------------------|--|
| requests.requests() | 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，对应与 HTTP 的 GET         |
| request.head()      | 获取 HTML 网页头的方法，对应与 HTTP 的 HEAD         |
| requests.post       | 向 HTML 网页提交 POST 请求的方法 对应于 HTTP 的 POST |
| request.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 网页提交删除请求，对应于 HTTP 的DELETE       |

## Response 对象的属性

| 属性                  | 解释                         |
|---------------------|----------------------------|
| r.status_code       | HTTP 请求的返回状态，200表示连接成功     |
| r.text              | 返回对象的文本内容                  |
| r.content           | 猜测返回对象的二进制形式               |
| 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）   |

```
1 | import requests #导入Requests库
2 |
3 | def get_html_text(url):
4 |     try:
5 |         r = requests.get(url,timeout=20) # 设置超时
```

```
6         #使用get方法发送请求，返回包含网页数据的Response并存储到Response对象r中
7         r.raise_for_status()
8         # 如果 HTTP 请求返回了不成功的状态码， Response.raise_for_status() 会抛出一个
        HTTPError异常
9         r.encoding = r.apparent_encoding
10        return r.text
11    except:
12        # 异常处理
13        return '产生异常'
14    if __name__ == '__main__':
15        url = ''
16        print(get_html_text(url))
```