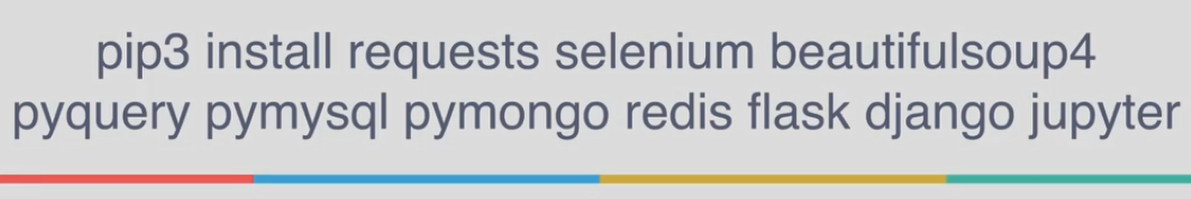
[Python3网络爬虫实战 1](#_Toc10120)

[安装常用库 1](#_Toc23010)

[爬虫基本原理 1](#_Toc30034)

# Python3网络爬虫实战

## 安装常用库



用于自动化测试

Chromedriver

Geckodriver

异步http请求库

Aiohttp

字符编码检测库

Cchardet

DNS解析库

Aiodns

光学字符识别

Tesserocr (tesseract软件的python API访问层，所以需要先下载tesseract,pip install tesserocr pillow)

APP爬取相关库的安装

Charles (作用域fiddler相似，好用，但是收费,安装代理证书，手机也是，然后手机代理电脑对外访问抓包)

Mitmproxy (与Fiddler和mitmproxy作用差不多，但是是命令行的内容)

Appium(手机浏览器的selenium,模拟点击，滑动，等操作)

爬虫框架

Pyspider

Scrapy

Scrapy-Splash (Scrapy中支持JavaScript的渲染工具)

Scrapy-Redis(分布式专用)

Scrapyd(部署与运行的Scrapy项目的工具)

Scrapy-Client(将Scrapy代码部署到远程的Scrapyd,通过集中式的远程管理)

Scrapy-API(通过API获取内容的Scrapyd的运行情况)

Scrapyrt(为scrapy提供一个调度的HTTP接口,通过HTTP来调度Scrapy任务)

Gerapy(Scrapy的分布式管理模块)

## 爬虫基本原理

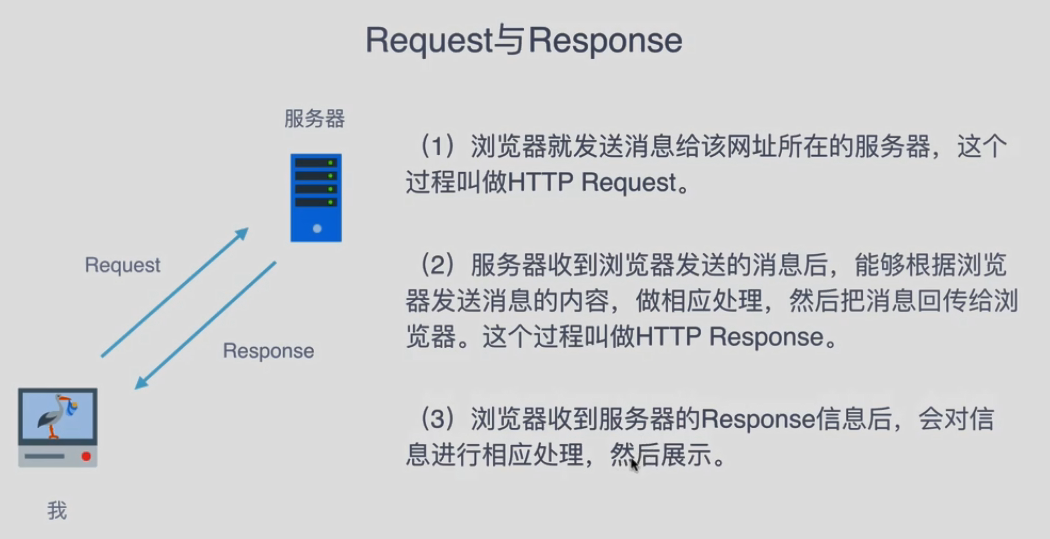
### 什么是爬虫

**请求**网站并**提取**数据的**自动化**程序

### 爬虫基本流程



### Request与Response的概念



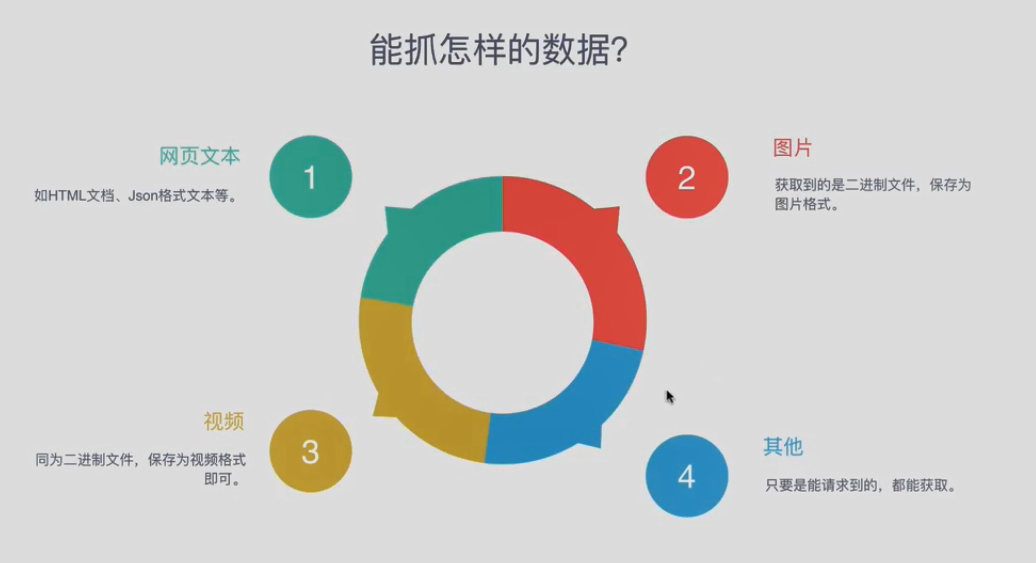
### Request包含的内容



### Response中包含的内容



### 能抓怎样的数据



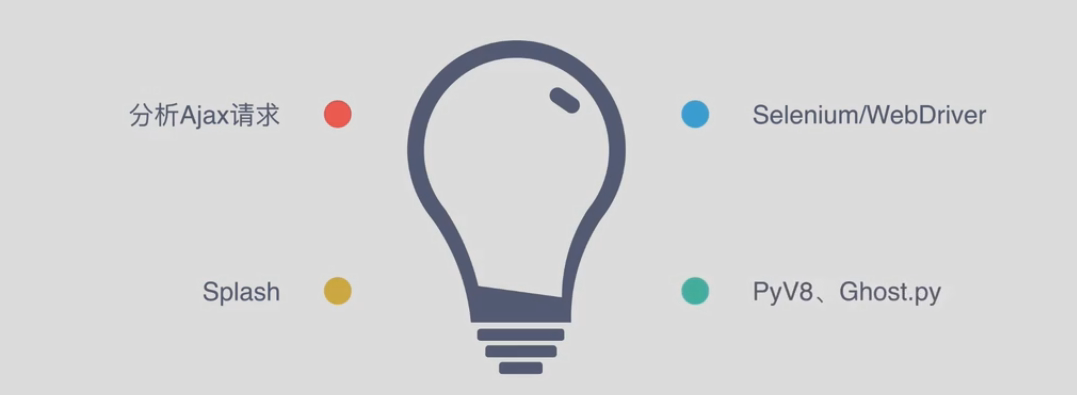
### 解析方式



### 为什么我抓到的和浏览器看到的不一样?

因为页面都是经过JavaScript后期加载渲染的，并不是单纯request请求回来的第一次内容

### 怎样解决JavaScript渲染的问题



使用selenium获取的页面原代码不是request当前URL的原代码，而是审查元素一样的，你能看到的内容组织起来的源代码

### 可以怎样保存数据?

