解析

1.xpath

```
xpath使用：
    注意：提前安装xpath插件
     （1）打开chrome浏览器
     （2）点击右上角小圆点
     （3）更多工具
     （4）扩展程序
     （5）拖拽xpath插件到扩展程序中
     （6）如果crx文件失效，需要将后缀修改zip
     （7）再次拖拽
     （8）关闭浏览器重新打开
     （9）ctrl + shift + x
     （10）出现小黑框
    1.安装lxml库
            pip install lxml -i https://pypi.douban.com/simple
    2.导入lxml.etree
            from lxml import etree
    3.etree.parse()   解析本地文件
            html_tree = etree.parse('XX.html')
    4.etree.HTML()    服务器响应文件
            html_tree = etree.HTML(response.read().decode('utf-8')
    4.html_tree.xpath(xpath路径)
```

```
xpath基本语法：
    1.路径查询
        //：查找所有子孙节点，不考虑层级关系
        / ：找直接子节点
    2.谓词查询
        //div[@id]
        //div[@id="maincontent"]
    3.属性查询
        //@class
    4.模糊查询
        //div[contains(@id, "he")]
        //div[starts-with(@id, "he")]
    5.内容查询
        //div/h1/text()
    6.逻辑运算
        //div[@id="head" and @class="s_down"]
        //title | //price
```

应用案例： 1.站长素材图片抓取并且下载（http://sc.chinaz.com/tupian/shuaigetupian.html）--》懒加载

2.JsonPath

```
jsonpath的安装及使用方式：
            pip安装：
                    pip install jsonpath
            jsonpath的使用：
                   obj = json.load(open('json文件', 'r', encoding='utf-8'))
                   ret = jsonpath.jsonpath(obj, 'jsonpath语法')
```

教程连接（http://blog.csdn.net/luxideyao/article/details/77802389）

案例练习:淘票票

作业： 1.股票信息提取（http://quote.stockstar.com/）

2.boos直聘

3.中华英才

4.汽车之家

3.BeautifulSoup

1.基本简介

```
1.BeautifulSoup简称：
     bs4
2.什么是BeatifulSoup?
    BeautifulSoup，和lxml一样，是一个html的解析器，主要功能也是解析和提取数据
3.优缺点?
    缺点：效率没有lxml的效率高
    优点：接口设计人性化，使用方便
```

2.安装以及创建

```
1.安装
    pip install bs4
2.导入
    from bs4 import BeautifulSoup
3.创建对象
    服务器响应的文件生成对象
        soup = BeautifulSoup(response.read().decode(), 'lxml')
    本地文件生成对象
        soup = BeautifulSoup(open('1.html'), 'lxml')
        注意：默认打开文件的编码格式gbk所以需要指定打开编码格式
```

3.节点定位

```
1.根据标签名查找节点
        soup.a 【注】只能找到第一个a
            soup.a.name
            soup.a.attrs
2.函数
        (1).find(返回一个对象)
                find('a')：只找到第一个a标签
```

```
            find('a', title='名字')
            find('a', class_='名字')
    (2).find_all(返回一个列表)
            find_all('a')  查找到所有的a
            find_all(['a', 'span'])  返回所有的a和span
            find_all('a', limit=2)  只找前两个a
    (3).select(根据选择器得到节点对象)【推荐】
            1.element
                eg:p
            2..class
                eg:.firstname
            3.#id
                eg:#firstname
            4.属性选择器
                [attribute]
                    eg:li = soup.select('li[class]')
                [attribute=value]
                    eg:li = soup.select('li[class="hengheng1"]')
            5.层级选择器
                element element
                    div p
                element>element
                    div>p
                element,element
                    div,p
                          eg:soup = soup.select('a,span')
```

4.节点信息

```
(1).获取节点内容：适用于标签中嵌套标签的结构
        obj.string
        obj.get_text()【推荐】
(2).节点的属性
        tag.name 获取标签名
            eg:tag = find('li)
               print(tag.name)
        tag.attrs将属性值作为一个字典返回
(3).获取节点属性
        obj.attrs.get('title')【常用】
        obj.get('title')
        obj['title']
```

应用实例： 1.股票信息提取（http://quote.stockstar.com/）

2.中华英才网-旧版

3 .腾讯公司招聘需求抓取（https://hr.tencent.com/index.php）