读取网页源代码：两个重要的库 requests 和 selenium

一、利用 requests 直接读网页

用法：

```
import requests
url = www.baidu.com
res = requests.get(url)
print(res.text)
```

1. 乱码（编码设置）：

查看编码 res.encoding;
查看网页编码;
把 res.encoding 改成网页编码
方法：res.encoding='utf-8';
res.text.encode('旧编码').decode('新编码')

```
url = 'https://www.baidu.com/s?tn=news&rtt=4&bsst=1&cl=2&wd=贵州茅台'
res = requests.get(url)
print(res.text)
```

2. 无法读取数据（浏览器伪装）：

返回的网页源代码没有数据，添加 headers 参数
headers={'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:87.0) Gecko/20100101 Firefox/87.0'}
res = request.get(url, headers=headers)

3. 假死（时长设置）

由于网络卡顿，导致无法获取数据，出现"假死"，陷入无限等待状态。

1. 设置 timeout 参数: res = request.get(url, headers=headers, timeout=10)
2. 但改方法并不完美，因为如果无法获取数据，则会返回异常（报错），导致程序终止，可用 try/except 结构处理异常

```
try:
    res = request.get(url, headers=headers, timeout=10)
except:
    print('Fail to getting data because of timeout')
```

3. 加上 try/catch 结构仍然不完美，因为失败后能保证程序继续运行，还是不能保证获取数据，因此加上 while 循环

```
ind_not_get_data = True
```

```
while ind_not_get_data:
    try:
        res = request.get(url, headers=headers, timeout=10)
        ind_not_get_data = False
    except:
        print('Fail to getting data because of timeout')
```

4. 访问网站过于频繁，导致 IP 被限制（代理设置）

对 requests.get 增加代理设置

```
proxy = ' 219.220.111.3:25764'   #需自己购买
proxies = {'http': 'http://' + proxy, 'https': 'https://' + proxy}
res = requests.get(url, headers=headers, timeout=10, proxies=proxies)
```

二、利用 Selenium 控制浏览器读取网页

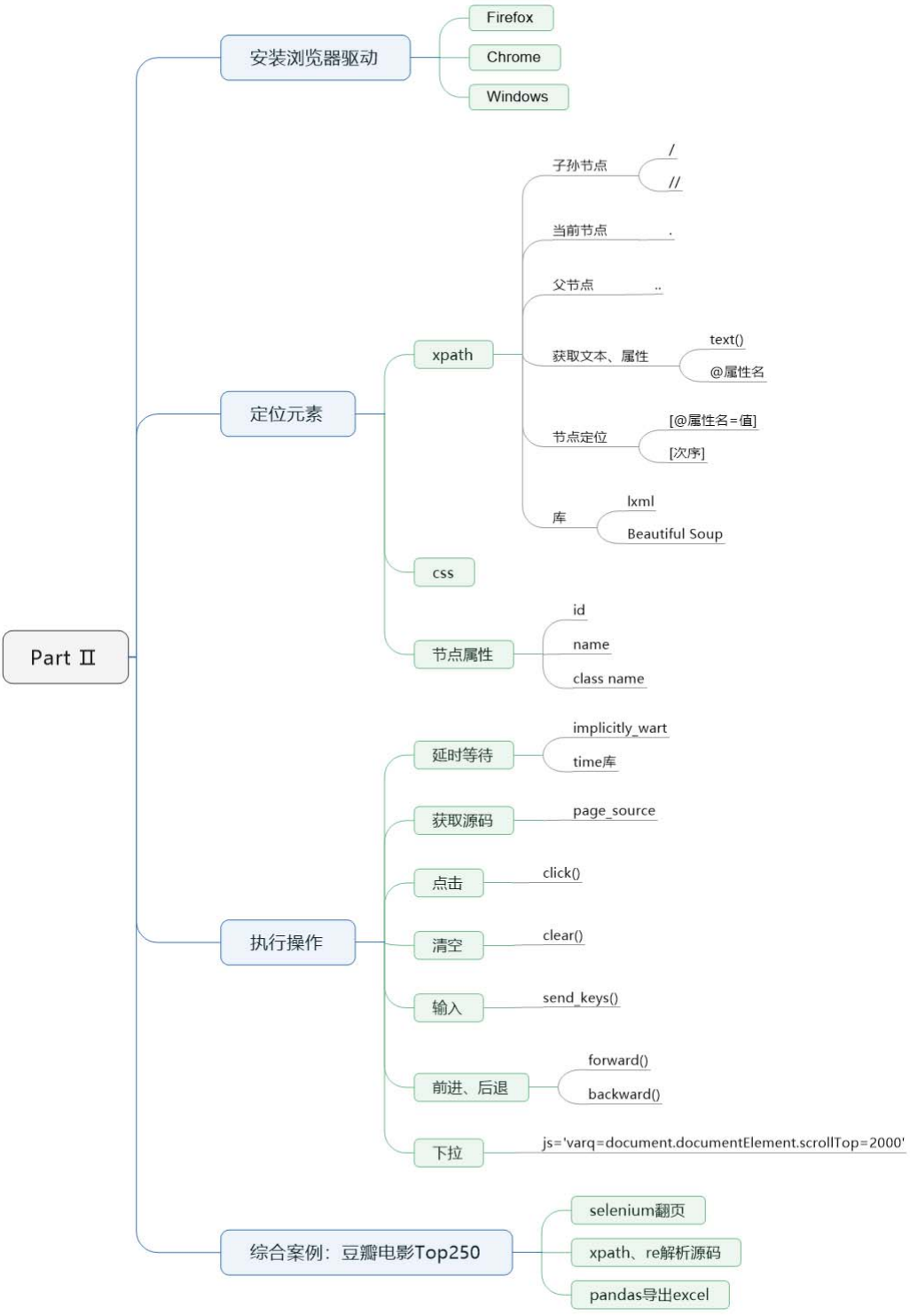
1. 安装浏览器驱动

Firefox 驱动：geckodriver
Chrome 驱动：chromedriver
windows 放置到 Anaconda3/Scripts

2. 定位元素：xpath，css，class name

3. 执行动作，模拟浏览行为（搜索、点击、翻页）

4. 获取网页源代码


Part Ⅱ
安装浏览器驱动
Firefox
Chrome
Windows
定位元素
xpath
子孙节点
/
//
当前节点
.
父节点
..
获取文本、属性
text()
@属性名
节点定位
[@属性名=值]
[次序]
库
lxml
Beautiful Soup
css
节点属性
id
name
class name
执行操作
延时等待
implicitly_wart
time库
获取源码
page_source
点击
click()
清空
clear()
输入
send_keys()
前进、后退
forward()
backward()
下拉
js='varq=document.documentElement.scrollTop=2000'
综合案例：豆瓣电影Top250
selenium翻页
xpath、re解析源码
pandas导出excel