数据分析及实践实验二实验报告

PB18061443 江昊霖

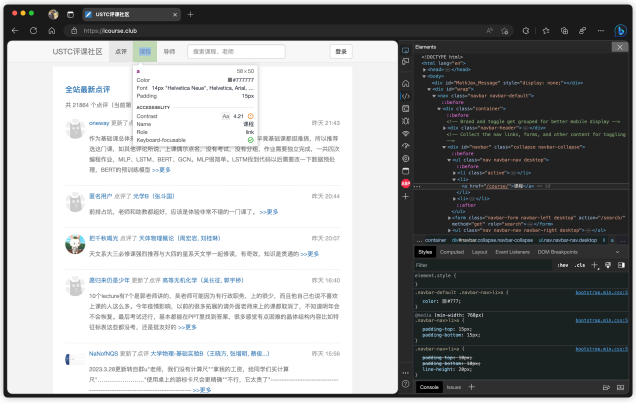
2023 年 5 月 3 日

1 实验目的

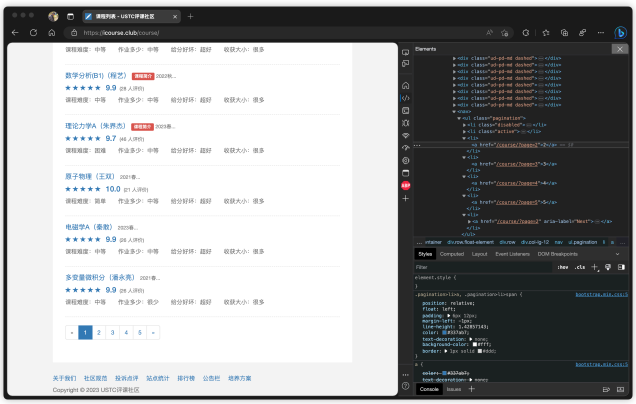
给定评课社区网站，需要设计一个网站遍历策略，爬取至少 200 个课程的详细信息，记录于 json 格式的文件中.

2 实验步骤

由给定的 url 进入课程列表网站爬取课程信息。



通过观察，课程列表的页面规律为：`https://icourse.club/course/?page={pagenumber}`，每页为 10 个课程。



递归进入每个课程页面后爬取信息.

3 代码

由于在进行请求时需要请求头，使用 fake_useragent 生成 user-agent.

3.1 获取课程列表网页链接

```
def courseUrl(url: str) -> str:
    """get course list page suffix from given page

    Args:
        url (str): given url

    Returns:
        str: course list page
    """

    headers = {'User-Agent': UserAgent().random}
    ret = Request(url, headers=headers)
    res = urlopen(ret)
    data = res.read().decode('utf-8')
    soup = bf(data, features="lxml")

    suffix = soup.find('a', string=re.compile('.*? 课程。*?'))['href']
    return url + suffix

def getCoursesID(url: str) -> tuple[list, list]:
    """get courses' links and courses' names

    Args:
        url (str): courses list page

    Returns:
        tuple[list, list]: courses' links and names
```

```
    """

    headers = {'User-Agent': UserAgent().random}
    ret = Request(url, headers=headers)
    res = urlopen(ret)
    data = res.read().decode('utf-8')
    soup = bf(data, features="lxml")

    links = soup.find_all(class_="px16")
    courses = []

    for index, link in enumerate(links):
        courses.append(link.text)
        links[index] = link['href']

    return links, courses

def courseInfo(courseID: str, courseName: str, serial: int) -> list:
    """get courses info

    Args:
        courseID (str): the courses id to course page
        courseName (str): course's name and teacher
        serial (int): serial of course

    Returns:
        info(list): info of the course
    """

    url = URL+courseID

    headers = {'User-Agent': UserAgent().random}
    ret = Request(url, headers=headers)
    res = urlopen(ret)
    data = res.read().decode('utf-8')
    soup = bf(data, features="lxml")

    info = []
    info.append(str(serial))
    info.append(courseName)
```

```
    block1 = soup.find_all(class_="right-mg-md", limit=5)
    del (block1[0])
    for name in block1:
        info.append(name.string.split(': ')[-1])

    links = soup.find_all('strong', limit=6)

    for link in links:
        info.append(link.nextSibling)

    return info
```