

.

|  |  |  |  |
| --- | --- | --- | --- |
| 写作  （40分） | 程序设计（60分） | 总分  （100分） | 指导教师签名 |
|  |  |  |  |

**《专业课程综合设计》**

**课程考核报告**

Python网络爬虫

|  |  |
| --- | --- |
| 学 院： | 人工智能学院 |
| 专业班级： | \*\*\*\*工程18秋\*班 |
| 姓 名： |  |
| 学 号： |  |
| 指导教师： | 张 三 |

**完 成 日 期 2021年 12月**

目录

[1、引言 1](#_Toc91517366)

[爬虫 1](#_Toc91517367)

[爬虫准备工作 2](#_Toc91517368)

[2、课程设计目的 2](#_Toc91517369)

[课程设计内容 2](#_Toc91517370)

[背景知识 3](#_Toc91517371)

[3、设计步骤与方法 3](#_Toc91517372)

[步骤1，数据的爬取 requests,re 3](#_Toc91517373)

[步骤2：数据处理pandas 9](#_Toc91517374)

[4、设计结果及分析 11](#_Toc91517375)

[5、问题及心得体会 11](#_Toc91517376)

[6、参考文献 12](#_Toc91517377)

# 1、引言

## 爬虫

网络爬虫（又被称为网页蜘蛛，网络机器人）就是模拟浏览器发送网络请求，接收请求响应，一种按照一定的规则，自动地抓取互联网信息的程序。  
原则上,只要是浏览器(客户端)能做的事情，爬虫都能够做。

为什么我们要使用爬虫

互联网大数据时代，给予我们的是生活的便利以及海量数据爆炸式的出现在网络中。

过去，我们通过书籍、报纸、电视、广播或许信息，这些信息数量有限，且是经过一定的筛选，信息相对而言比较有效，但是缺点则是信息面太过于狭窄了。不对称的信息传导，以致于我们视野受限，无法了解到更多的信息和知识。

互联网大数据时代，我们突然间，信息获取自由了，我们得到了海量的信息，但是大多数都是无效的垃圾信息。

例如新浪微博，一天产生数亿条的状态更新，而在百度搜索引擎中，随意搜一条——减肥100,000,000条信息。

在如此海量的信息碎片中，我们如何获取对自己有用的信息呢？

答案是筛选！

通过某项技术将相关的内容收集起来，在分析删选才能得到我们真正需要的信息。

这个信息收集分析整合的工作，可应用的范畴非常的广泛，无论是生活服务、出行旅行、金融投资、各类制造业的产品市场需求等等……都能够借助这个技术获取更精准有效的信息加以利用。

网络爬虫技术，虽说有个诡异的名字，让能第一反应是那种软软的蠕动的生物，但它却是一个可以在虚拟世界里，无往不前的利器。

## 爬虫准备工作

我们平时都说Python爬虫，其实这里可能有个误解，爬虫并不是Python独有的，可以做爬虫的语言有很多例如：PHP,JAVA,C#,C++,Python，选择Python做爬虫是因为Python相对来说比较简单，而且功能比较齐全。

# 2、课程设计目的

在即将面临毕业季找工作难的问题，恰逢这次课程设计，于是想要通过爬虫爬取前程无忧这个网站的招聘信息，并对其进行数据统计，找出适合自己的工作等

## 

## 课程设计内容

本次课设主要由两部分组成：

1. 首先是网站数据的爬取，并且将一些有用的信息存在一个CSV文件中，由于网站较大，所以我们只是单纯爬取符合本专业的职位，一共爬取了8000多条数据信息。
2. 然后是使用pandas库对这些数据进行分析出来，把岗位详细，公司名称，公司详情，薪资，工作地点，发布日期，更新日期，公司类型，行业，工作经验要求，学历，招聘人数等信息保存到CSV中

## 

## 背景知识

“前程无忧”(Nasdaq: JOBS)是国内一个集多种媒介资源优势的专业人力资源服务机构，创始人为甄荣辉。

它集合了传统媒体、网络媒体及先进的信息技术，加上一支经验丰富的专业顾问队伍，提供包括招聘猎头、培训测评和人事外包在内的全方位专业人力资源服务，全国25个城市设有服务机构。

2004年9月，前程无忧成为第一个在美国纳斯达克上市的中国人力资源服务企业，是中国最具影响力的人力资源服务供应商之一。

前程无忧成立于1999年，是一家网络招聘服务提供商，网站目标有两大部分：致力于为积极进取的白领阶层和专业人士提供更好的职业发展机会。同时，网站致力于为企业搜寻、招募到最优秀的人才。

提供报纸招聘、网络招聘、招聘猎头、培训测评和人事外包在内的人力资源服务，最初全国包括香港在内的26个城市设有服务机构，已经增加至104个城市。

经过一个学期python课程的学习，我对python语言也有了一定程度的掌握，特别是python丰富的第三方库，可以使得我更加容易的完成对该网络招聘信息的批量爬取与分析。

# 3、设计步骤与方法

实验环境：

Windows 10

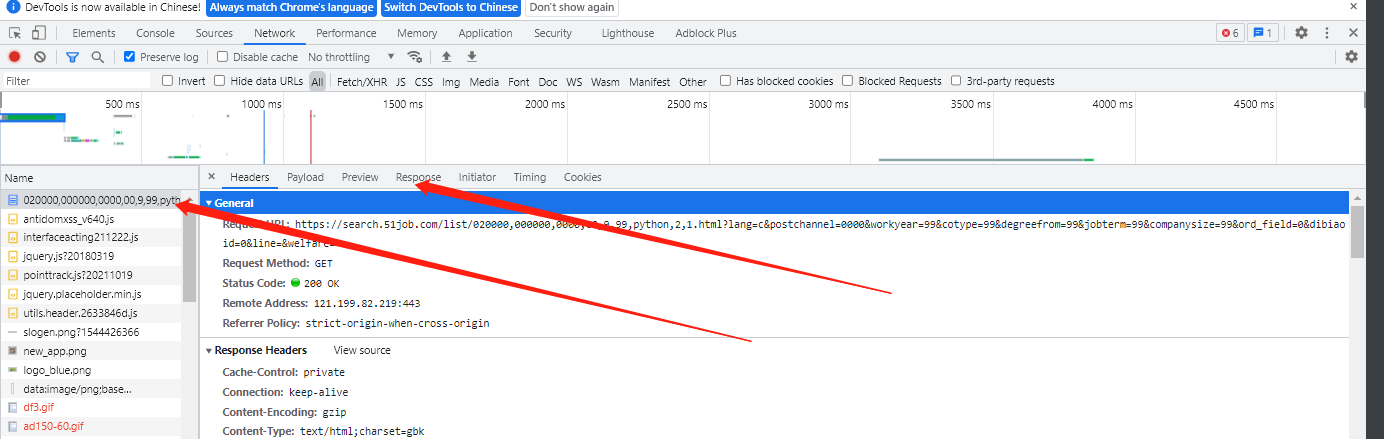
pycharm 2021.2

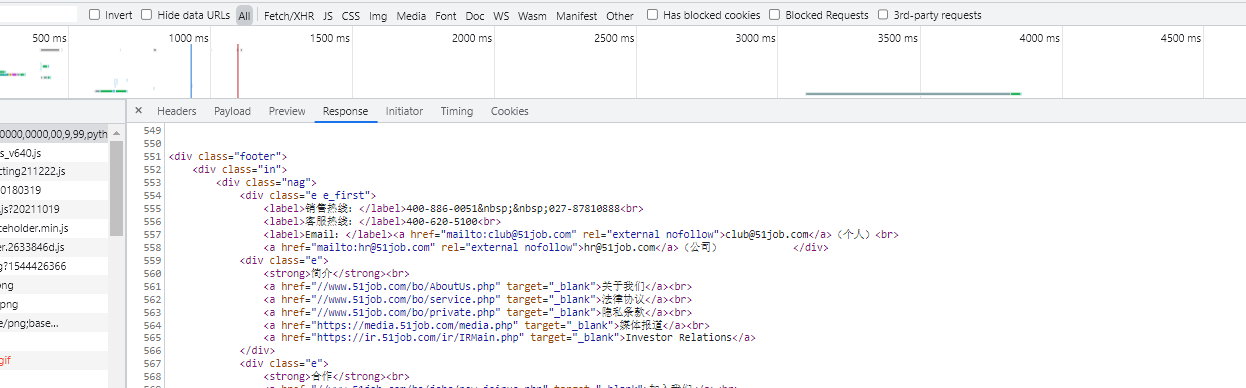
python 3.9

## 步骤1，数据的爬取 requests,re

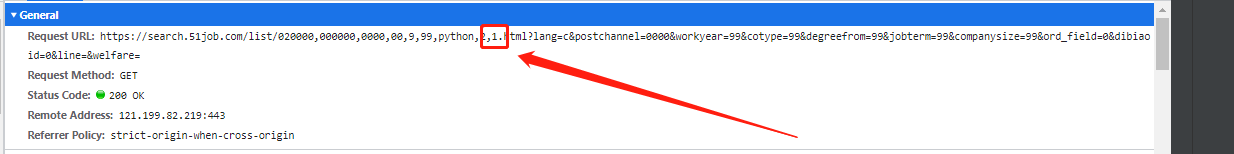
我们打开chromd 谷歌浏览器的开发人员工具查看该信息在网站时获取到的请求信息





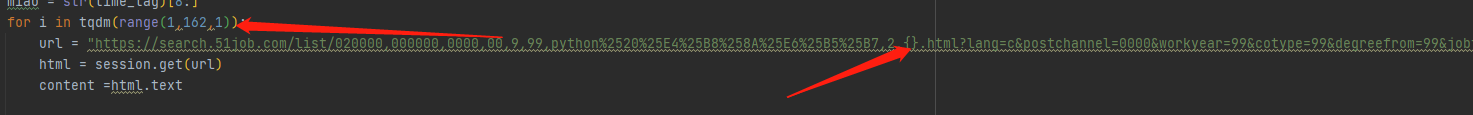


发现其实它是一个静态网站



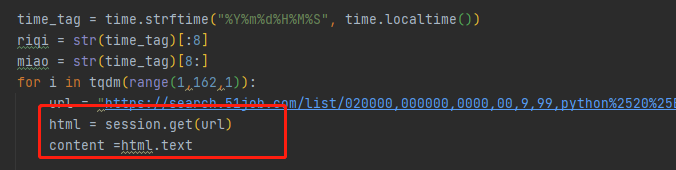
再通过翻页去查询它们的规律发现，其实每翻一页，这个数字就会加1，那么就可以构建出整个网站的页数，去批量爬取所有页数的招聘信息

构建页数的代码如下：



每次这个传递一个参数的时候，对应的url也会更新，这样就可以构造出全部的url方便我们爬取

因为上面说到的这个网站是一个静态网站，所以内容都是全部存放到HTML页面的，那么我们获取HTML的源码，那么对应的就是获取招聘信息的内容

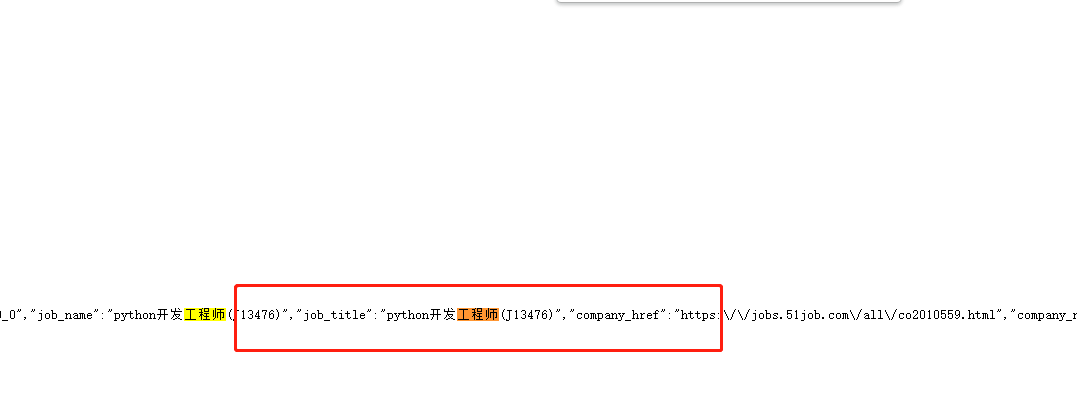


通过这两句便是获取每一页的html源码信息

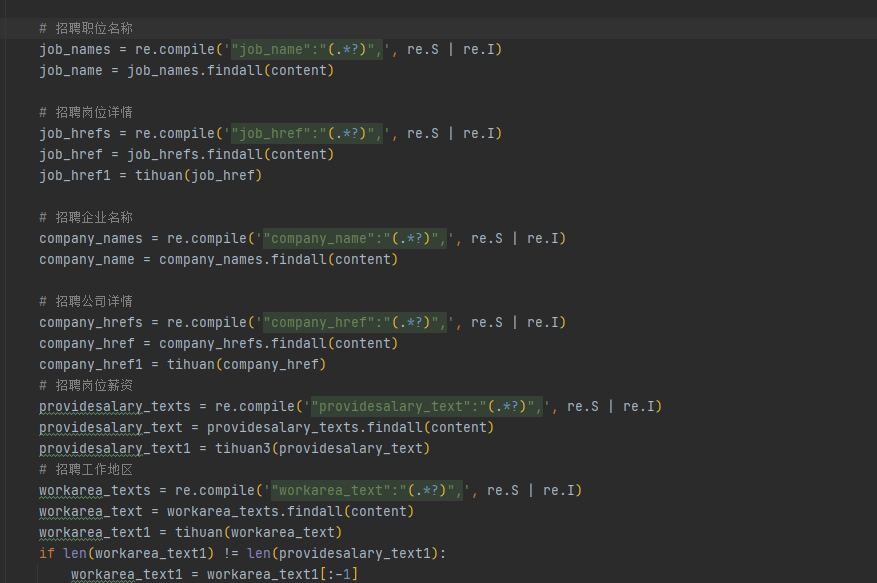
因为他是静态的，在获取源码之后我们还得用re来进行定位

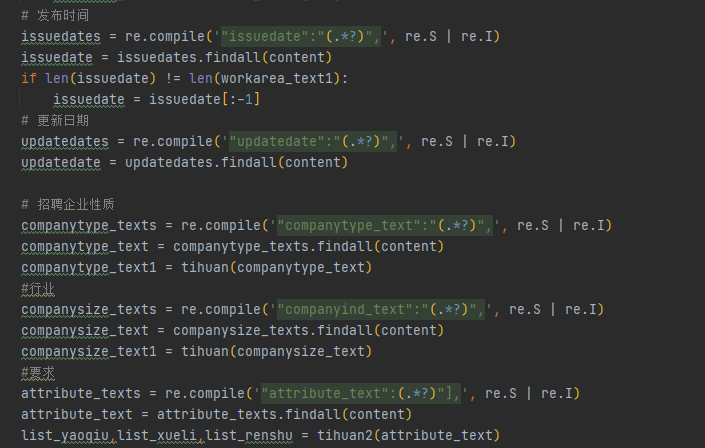
正则表达式（英语：Regular Expression，常简写为regex、regexp或RE），又称正则表示式、正则表示法、规则表达式、常规表示法，是计算机科学的一个概念。

正则表达式使用单个字符串来描述、匹配一系列匹配某个句法规则的字符串。在很多文本编辑器里，正则表达式通常被用来检索、替换那些匹配某个模式的文本。   
   
许多程序设计语言都支持利用正则表达式进行字符串操作。例如，在Perl中就内建了一个功能强大的正则表达式引擎。正则表达式这个概念最初是由Unix中的工具软件（例如sed和grep）普及开的。



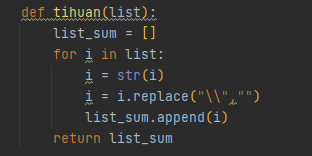
具体的代码如下：





其中因为有很多数据在获取的图中并不干净，所以我们还得去构造函数对获取到的这些数据一定程度的清洗

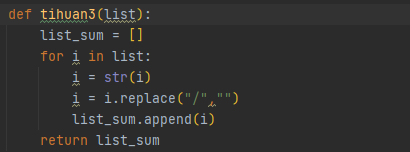
对于获取URL的



对于其他信息的

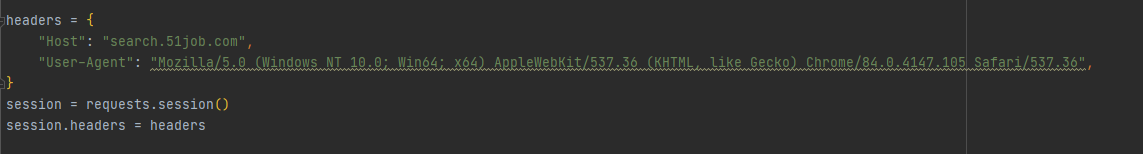


对于获取薪资的



最后将内容用for循环依次写入csv文件中，

关于反爬虫，我们这边是通过构建headers



来伪造成浏览器去爬取对应的内容

然后再用session来构建cookie，减轻爬取数据的时候对服务器造成的压力

最后每爬取一页信息



便用time.sleep缓存一秒，防止爬取过快被系统检测出是爬虫程序

## 步骤2：数据处理pandas

在获取全部信息之后

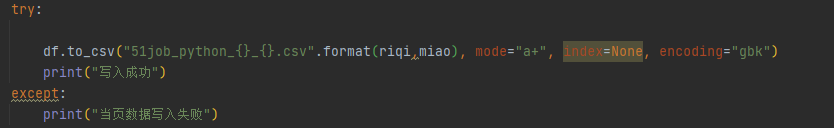
我们先是创建一个dataframe用来保存数据



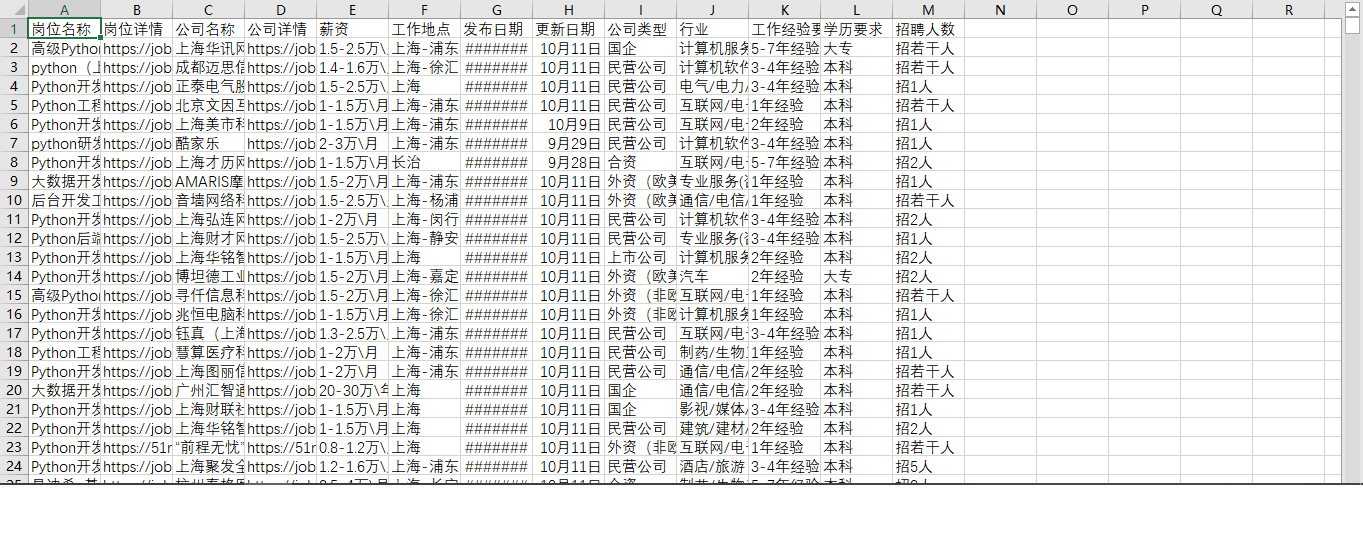
然后再依次把对应的数据内容传入到dataframe中



再用防错机制把获取到的数据一一保存到CSV中



最后数据的样式如下：



# 4、设计结果及分析

通过上面的数据集可以清晰的认知关于招聘python岗位的要求，任职条件，招聘人数，以及有多少岗位数量等这些非常重要的信息，也清晰的认知自己的能力不足，以及和别人之间的差异，还是有多虚心学习，争取在毕业的时候能找到一份自己满意的工作

# 5、问题及心得体会

本次课程完成的过程中，遇到了许多问题，大部分问题都通过自己的debug或者网络教程解决。

关于前程无忧爬取的教程网上也有很多，不过以较为老旧，经过了解，前程无忧在去年就关闭了这些网络教程所使用的API，现在访问这些URL只能看到500，所以只能通过实际访问中查看网络请求寻找爬取实际数据的入口。

本次的课程设计巩固和加深了我对python爬取信息知识的理解，提高了综合运用所学知识的能力，增强了根据课程需要选学参考资料，查阅手册、图表和文献资料的自学能力。

同时，在本次课程中，我通过实际项目，更加熟悉python的使用，并且体会到它的强大之处。首先它易于扩展，在pypi商店和GitHub上，有许多免费开源的包提供我们直接使用，这大大减少了我们的工作量，并且方便我们实行现水平下难以实行的功能。其次，它的数据结构类型很丰富，列表，字典，dataframe等数据结构大大方便了我们的使用，不过，在数据处理中，也体会到python的运行速度逊于C、C++的方面，所以也学会了通过优化算法减少时间复杂度来给程序提速。

# 6、参考文献

[正则表达式 - 语法](https://www.runoob.com/regexp/regexp-syntax.html)

[Python 爬虫介绍](https://www.runoob.com/w3cnote/python-spider-intro.html)