**Python分析中国大陆各直辖市及各省省会的平均工资与平均房价**

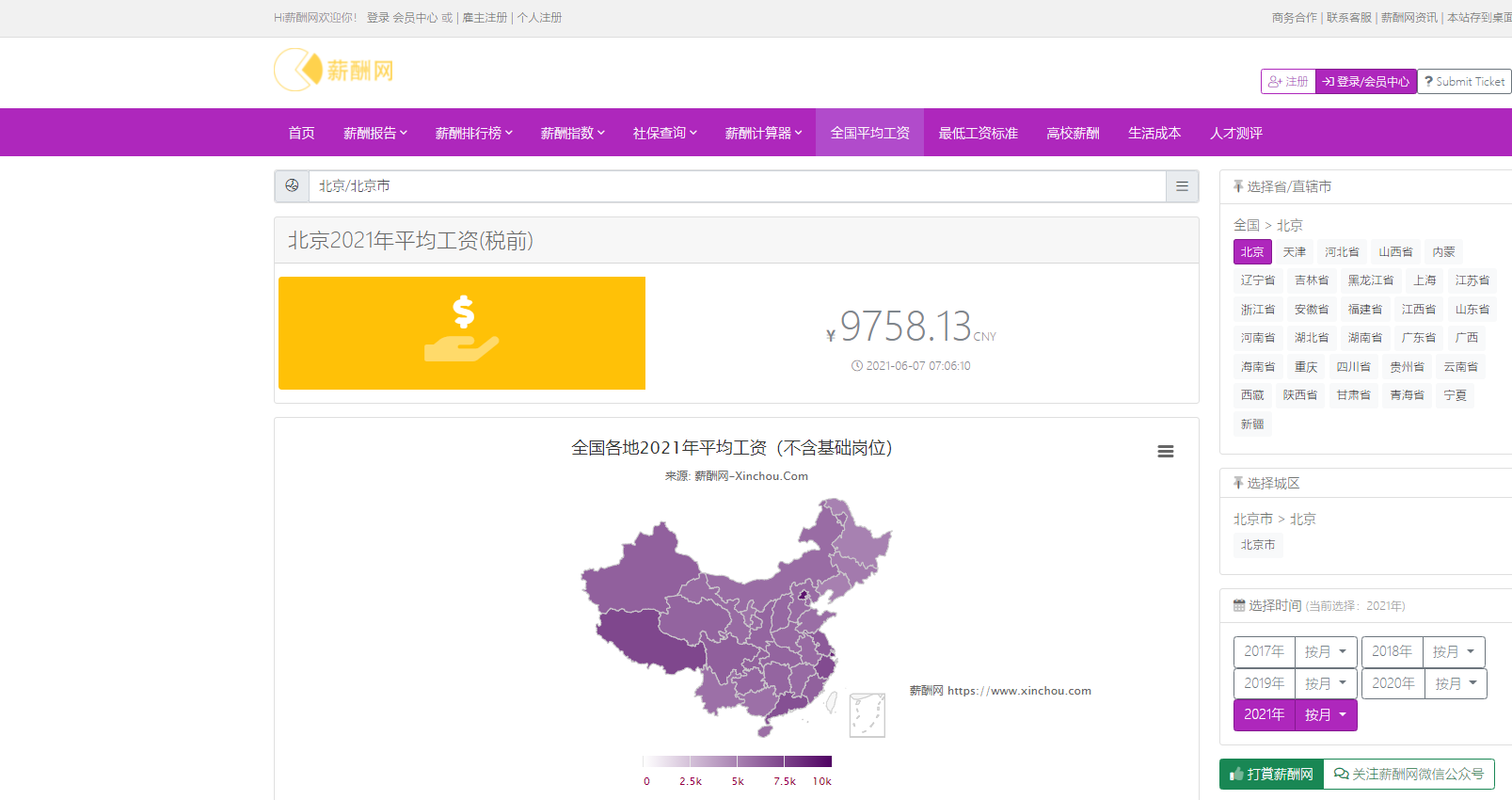
**-设计方案**

**01.研究目的**

近期网络上关于“躺平”的争论很多，但可以看出支持“躺平”的多是以90后为主的年轻人，反对“躺平”的是中老年人，而让年轻人“躺平”很大的一个原因便是高昂的房价。因此我想爬取相对客观的数据，看看各直辖市和各省省会的平均工资与房价是多少，分析房价是不是真的高到买不起，高到让年轻人躺平的地步。

**02.爬取数据的来源**

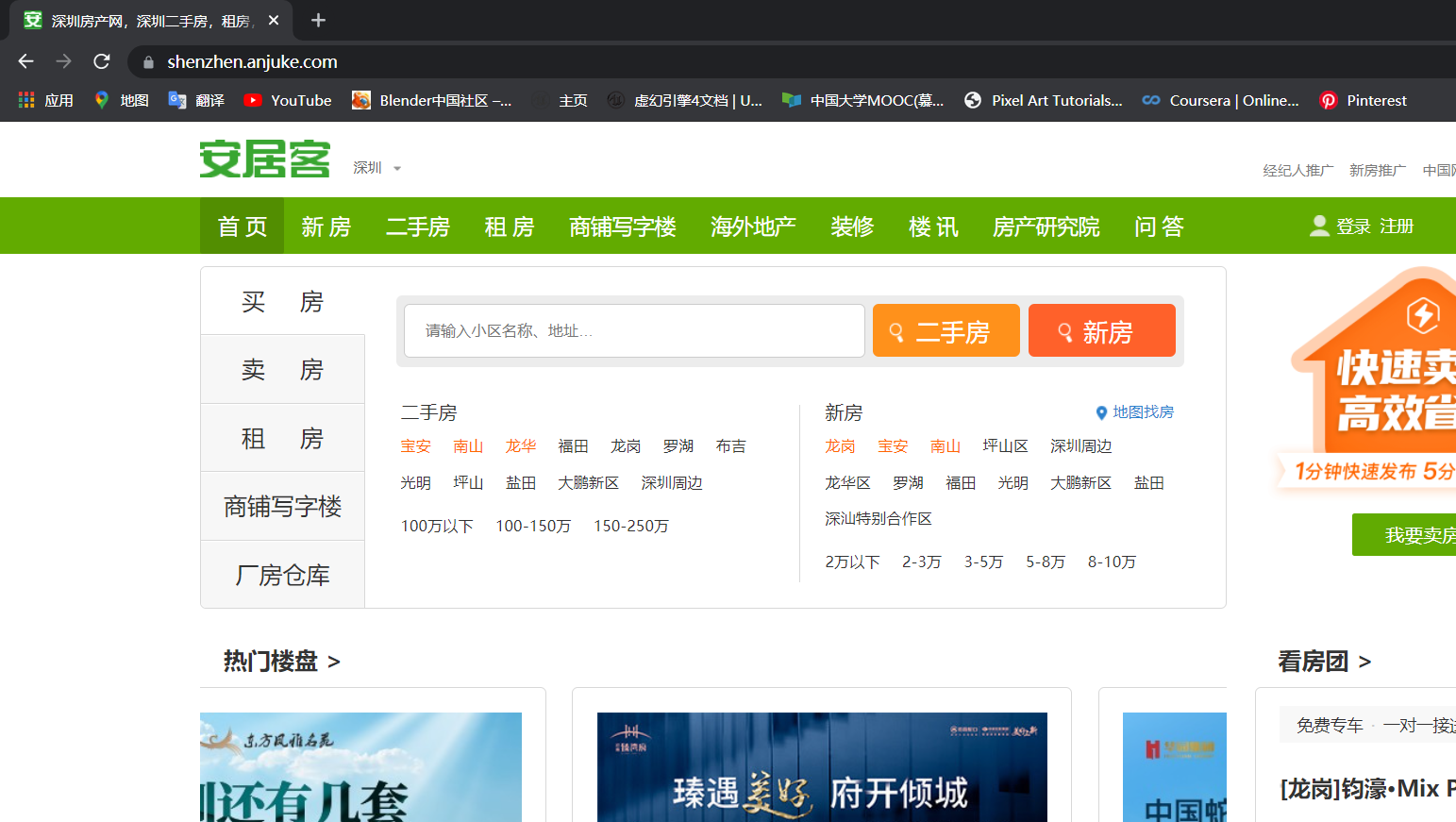
爬取平均工资的网站：薪酬网



爬取平均房价的网站：安居客



很诡异的一件事是在安居客上查询不到深圳的平均房价，但深圳既不是直辖市也不是省会城市故我不再深究





**03.爬取数据中遇到的问题**

在大数据越来越重要，爬虫技术越来越先进的今天，反爬虫的手段也愈加高明。在爬取薪酬网的数据时，即使伪装了头部信息、设置了休眠时间依然会被识别为爬虫；而在爬取安居客数据时，它不仅会设置验证信息，网页还会动态变化。比如某一时刻，福州网页的网址为<https://fuzhou.anjuke.com/market/>而过了某个时间段后其网址就变为<https://fz.anjuke.com/market/。我只好先把这31>个城市的平均工资和平均房价的网页先爬取下来再爬取相关的数据。

**04.爬取相关数据**

爬取平均工资：

Average\_wage\_get.py，输出data1

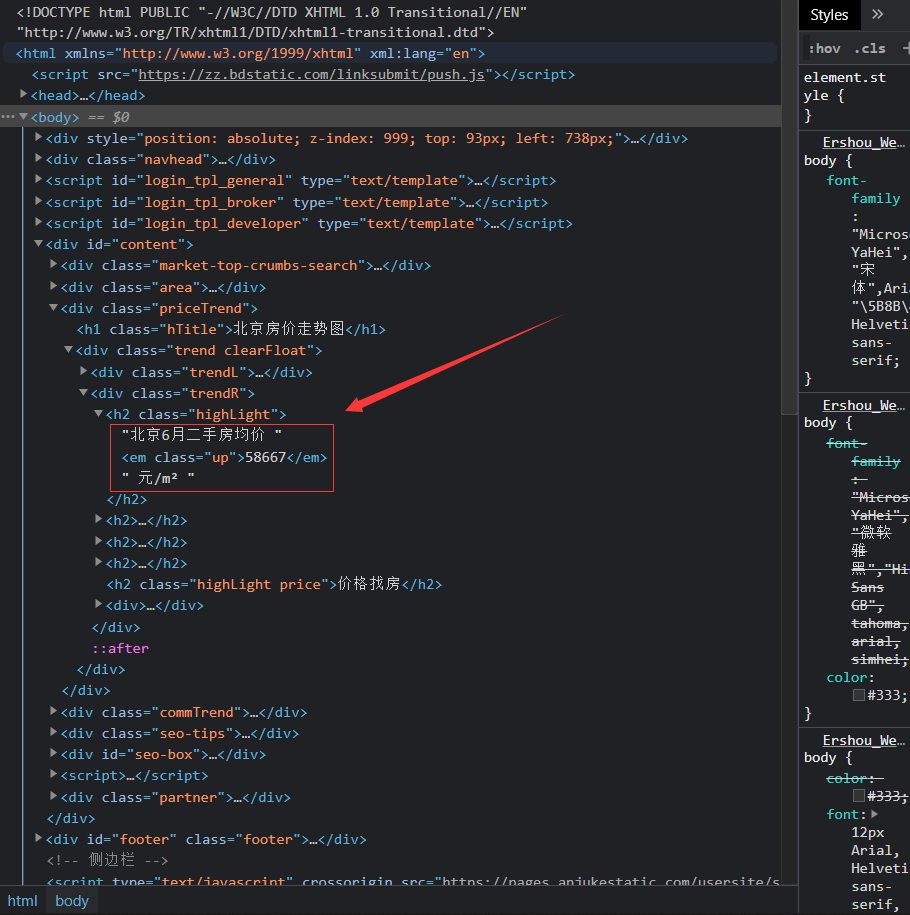


代码：

text = get\_text\_from\_saved\_html(city\_lst[i]) # 从保存的html里获取网页内容  
soup = BeautifulSoup(text, 'html.parser')  
city\_name = '' # 创建一个空字符串用来存放城市字符串  
wage\_str = '' # 创建一个空字符串用来存放房价字符串  
  
sf1 = soup.find(class\_="card-header") # 查找城市名所在位置  
st1 = sf1.text.replace(" ", "").strip() # 去掉空格、换行符等  
flag = 0  
while not st1[flag].isdigit():  
 city\_name += st1[flag]  
 flag += 1  
city\_name.replace('市', '')  
city.append(city\_name)  
  
sf2 = soup.find(class\_="text-center justify-content-between") # 查找平均工资所在位置  
st2 = sf2.text.replace(" ", "").strip() # 去掉空格、换行符等  
flag = 0  
while st2[flag] != "C":  
 wage\_str += st2[flag]  
 flag += 1  
wage\_num = eval(wage\_str) # 将字符串转换为数字  
wage.append(wage\_num)

爬取平均房价：

Average\_housing\_price\_get.py，输出data2



代码：

city = [] # 创建一个列表用来存放城市  
housing\_price = [] # 创建一个列表用来存放房价  
  
for i in range(len(city\_lst)): # 遍历传入的城市列表，查找保存的网页中城市名及对应房价  
 text = get\_text\_from\_saved\_html(city\_lst[i]) # 从保存的html里获取网页内容  
  
 soup = BeautifulSoup(text, 'html.parser')  
 sf = soup.find(class\_="highLight") # 根据浏览器检查网页内容确定了所要寻找的信息所在位置  
 city\_name = '' # 创建一个空字符串用来存放城市字符串  
 housing\_price\_str = '' # 创建一个空字符串用来存放房价字符串  
  
 st = sf.text.replace(" ", "").strip() # 去除空格、换行符等  
 flag = 0 # 标记位置  
 while not st[flag].isdigit(): # 获取城市字符串  
 city\_name += st[flag]  
 flag += 1  
 city\_name.replace('市', '')  
 city.append(city\_name)  
  
 for j in range(-9, -4): # 获取房价字符串  
 housing\_price\_str += st[j]  
 housing\_price\_num = eval(housing\_price\_str) # 将房价由字符串型转换为数字型，为后续计算做准备  
 housing\_price.append(housing\_price\_num)

**05.数据处理**

从Average\_wage\_get.py导入data1，从Average\_housing\_price\_get.py，输出data1，在data\_handle.py里用pandas输出类型为DataFrame的data保存到.csv和.xlsx文件

代码：

city\_lst = [] # 省会直辖市列表  
average\_wage\_lst = [] # 平均工资列表  
average\_housing\_price\_lst = [] # 平均房价列表  
time\_lst = [] # 不吃不喝买房所需时间  
data = dict() # 创建一个数据字典用来汇总所有的数据  
  
  
for i in data2.keys(): # 从数据2获取键放入城市列表中  
 city\_lst.append(i)  
for i in data1.values(): # 从数据1获取值放入平均工资列表中  
 average\_wage\_lst.append(i)  
for i in data2.values(): # 从数据2获取值放入平均房价列表中  
 average\_housing\_price\_lst.append(i)  
for i in range(len(city\_lst)): # 计算每个城市在不吃不喝(即工资完全用来买房)的情况下用平均工资买房要花多长时间  
 months = round(average\_housing\_price\_lst[i]\*100/average\_wage\_lst[i]) + 1 # 因为房贷少交一个月都不行所以四舍五入+1  
 years = round(months/12) + 1 # 同上  
 time\_lst.append(years)  
  
data["city"] = city\_lst # 城市字典  
data["wage"] = average\_wage\_lst # 平均工资字典  
data["housing\_price"] = average\_housing\_price\_lst # 平均房价字典  
data["time"] = time\_lst # 买房花费时间字典  
  
df = pd.DataFrame(data) # 创建DataFrame,输出为.csv和.xlsx文件  
df.to\_csv('大陆各省会直辖市平均工资及平均房价.csv', encoding="utf-8", index=False)  
df.to\_excel('大陆各省会直辖市平均工资及平均房价.xlsx')

**06.数据展示**

在Data\_Show.py中读取大陆各省会直辖市平均工资及平均房价.xlsx，用matplotlib.pyplot模块作图。