主要技术：urllib,selenium

在爬取辽宁博物馆的数据中，我首先分析是如何获取数据的。首先，辽宁博物馆的数据没有图片，没有隐藏链接，只有文字，共两万六千多条数据如下：



需要通过不断翻页来获取下一页的数据。在打开F12后，我通过翻页发现是通过更新下一页的接口来获取数据，因此选取的获取接口的技术为urllib，接口格式为https://www.lnmuseum.com.cn/singleMuseum/szwwkt/list?\_t=1712718600&year=&category=&name=&currentPage=1&size=15，在翻页中接口连接其中的变量是list?\_t和Page，Page简单，不断++，list?\_t在我研究中发现来自于timestamp，是加载完数据后的时间戳，因此需要如下代码获取加载当前页面的时间戳

    timestamp = driver.execute\_script("return window.performance.timing.loadEventEnd;")/1000

并且要除以1000，因为格式与int(time.time())获取的当前时间戳相同。

在了解完获取接口的技术后，我需要进行自动化测试，模拟浏览器自动点击下一页的行为，在点击后立马获取当前时间戳，更新list?\_t，要进行自动化测试，因此需要selenium来实现，选用的浏览器为chrome，这部分代码如下：

service = Service(executable\_path='E:\\vscode1\\python\\urllib\_use\\chromedriver.exe')

options = webdriver.ChromeOptions()

options.add\_experimental\_option("detach",True)

options.add\_argument('-ignore-certificate-errors')

options.add\_argument('-ignore -ssl-errors')

options.add\_argument('--disable-gpu')

driver=webdriver.Chrome(service=service,options=options)

url="https://www.lnmuseum.com.cn/#/collect/digital-culture"

driver.get(url)

button=driver.find\_element(By.CLASS\_NAME,"btn-next")

会自动打开辽宁博物馆页面，button是我通过class\_name来找到翻页按钮，在后续代码中使用button.click()就可以翻页了

下面代码是使用urllib的部分：

    \_url="https://www.lnmuseum.com.cn/singleMuseum/szwwkt/list?\_t="+str(timestamp)+"&year=&category=&name=&currentPage="+str(page)+"&size=15"

    headers={

        "user-agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36"

    }

    request=urllib.request.Request(url=\_url,headers=headers)

    response=urllib.request.urlopen(request)

    content=response.read().decode('utf-8')

爬取数据获取content，再把content读写到json文件中，再用json转格式

最后再写入csv文件里：

    data\_list=content\_dict['result']

    csv\_file='museum.csv'

    fieldnames = list(data\_list[0].keys())

    with open(csv\_file, mode='a', newline='', encoding='utf-8') as file:

        writer = csv.DictWriter(file, fieldnames=fieldnames)

        # 写入列名

        if flag==0:

            writer.writeheader()

            flag=1

        # 写入数据

        for list\_data in data\_list:

            writer.writerow(list\_data)

    time.sleep(2)

这里最后用time.sleep(2)是爬爬取太快给ip封了，但最后发现爬的很慢很慢。