		网络通信要素	网络: 网络是将具有独立功能的多台计算机通过 通信线路和通信设备连接起来,在网络管理软件 及网络通信协议下,实现资源共享和信息传递的 虚拟平台	简单来说:网络就是一种辅助双方或者多方能够 连接在一起的工具			
	01-网络通信		学习网络的目的就是为了联通多方然后进行通信用的,即把数据从一方传递给另外一方 IP地址: IP地址是分配给网络设备上网使用的数字标签,它能够标识网络中唯一的一台设备,好比现实中每个人都有一个手机号。	IPv4:是目前使用的ip地址,呈现形式是十进制的			
		P地址	ip地址分类 ip地址的作用:通过IP地址找到网络中唯——台设备,然后可以跟这个设备通信	IPv6:是未来使用的ip地址 ipconfig:查看ip地址			
			windows常用几个cmd命令 端口:端口是传输数据的通道,是数据传输的必经之路 端口号:操作系统为了统一管理这么多端口,就	ping 域名 : 检查是否能够上网 ping 127.0.0.1: 检查本地网卡是否正常			
		端口	对端口进行了编号,这就是端口号,端口号其实就是一个数字,好比我们现实生活中的门牌号。端口号有65536个 最终通信流程:通过ip地址找到对应的设备,通过端口号找到对应的端口,然后通过端口把数据给应用程序				
				知名端口号: 0到1023	知名端口号是指众所周知的端口号,这些端口号一般固定分配给一些服务,比如21端口分配给FTP(文件传输协议)服务,25端口分配给SMTP(简单邮件传输协议)服务,80端口分配给HTTP服务 这些端口不像知名端口那样为众所周知的服务保留,但用于一些特定的应用程序。它们可以被公		
			端口号的分类	注册端口号: 1024到49151 动态端口号: 49152到65535	司或应用程序开发者申请注册,用于某些特定的服务。虽然不是强制的,但在软件开发中通常遵循这一约定来避免端口冲突 这些端口不像知名端口那样为众所周知的服务保留,但用于一些特定的应用程序。它们可以被公司或应用程序开发者申请注册,用于某些特定的服务。虽然不是强制的,但在软件开发中通常遵循这一约定来避免端口冲突		
		socket	Socket (套接字) 是计算机网络编程中的一个抽象概念,它提供了一种用于不同计算机之间进行通信的机制。Socket允许不同计算机上的程序能够通过网络互相发送数据,以实现信息交流和数据传输。Socket是网络编程中的核心工具,用于建立和管理网络连接	socket可以看做是编程的接口,两个之间建立了虚拟通道			
		UDP通信	UDP: UDP (User Datagram Protocol) 是一种无连接的网络协议, UDP是一种快速、简单的数据传输协议, 但它不保证数据的可靠传输(不保证数据一定到达, 也不保证按顺序到达)。这使得UDP非常适合那些对速度要求高但可以容忍一定丢包的应用				
			TCP: TCP (Transmission Control Protocol,传输控制协议)是一种面向连接的、可靠的、基于字节流的传输层协议,它是互联网协议套件中的一部分,用于在计算机网络上可靠地传输数据 TCP通信步骤	1. 创建连接 2. 传输数据			
			TCP特点	3. 关闭连接 1. 面向连接 2. 可靠连接 编码是将字符串转换为二进制数据	字符串内容.encode("utf-8")		
		TCP通信	Service Control of the Control of th	解码则是将二进制数据还原为字符串 三次握手建立连接: 在TCP通讯中,建立连接需要进行三次握手	二进制数据.decode("utf-8") 第一次握手:客户端向服务器发送连接请求,并等待服务器确认 第二次握手:服务器收到请求后,确认连接,并向客户端发送确认		
			三次握手,四次挥手	四次挥手关闭连接: 在数据传输完成后, 需要进行四次挥手来关闭连接	第三次握手:客户端收到确认后,发送最终确认,连接建立完成 第一次挥手:客户端一方发送连接关闭请求。 第二次挥手:服务器确认请求,并停止发送数据。 第三次挥手:服务器发送连接关闭请求		
			TOD流传、体现连手 L.油体和	具体流程,请参考链接: https://blog.csdn.net/m0_49330686/article/details/129400464	第四次挥手:客户端确认请求,连接关闭完成。		
		爬虫概念: 网络爬虫 (又被称为网页蜘蛛, 网络机器人) 就是模拟客户端(主要指浏览器)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	TCP通信:代码请看上课代码 **原则上,只要是客户端(浏览器)能做的事情,爬虫都能够做** **爬虫也只能获取客户端(浏览器)所展示出来的数据**(可见可爬)				
		一 爬虫的分类	通用爬虫聚焦爬虫	特点: 爬取网站数量没有上限示例: 搜索引擎特点: 爬取网站数量有上限, 有明确的目标	功能性爬虫	特点:不获取数据,只为实现某一功能	
				分类	数据增量爬虫	示例: 投票/抢票/短信轰炸等 特点: 获取数据用于后续分析 分类	url与数据同时变化 url不变,数据变化
	02-爬虫基础	一爬虫的流程	确定目标网站(确定网址。爬取哪一个网站) 程序模拟浏览器向服务器发请求,(要数据) 服务器给你响应(响应对象) 数据的精确提取 数据保存				
			 	http: HTTP协议的全称是超文本传输协议。超文本是指在文本数据的基础上还包括非文本数据,非文本数据有图片、音乐、视频等,而这些非文本数据会使用链接的方式进行加载显示,通俗来说超文本就是带有链接的文本数据也就是我们常说的网页数据			
			url构成:	https: http+ssl (ssl: 具有服务器身份验证和数据输出的加密功能) 输入网址: 我们在浏览器的地址栏中输入网址,也就是统一资源定位符(URL),它指定了我们想要访问的网站的域名和路径 域名解析: 浏览器需要将域名转换为服务器的IP地址,也就是一串数字,表示服务器在互联网上			
				的位置。浏览器会先检查本地缓存或者操作系统的hosts文件,看是否有对应的IP地址。如果没有,浏览器会向域名系统(DNS)服务器发送查询请求,获取IP地址 建立连接:浏览器根据IP地址和端口号,通过传输控制协议(TCP)与服务器建立一个连接(三次握手)			
			浏览器访问web服务器过程	发送请求:浏览器通过超文本传输协议 (HTTP)或者其安全版本(HTTPS)向服务器 发送一个请求报文,包含了请求方法、请求头和 请求体等信息。请求方法指定了我们想要对服务 器资源进行什么操作,如GET、POST等 根据请求从对应的数据库里面获取对应的数据			
		网址的概念:网址又称为URL,URL的英文全拼是(Uniform Resoure Locator),表达的意思是统一资源定位符,通俗理解就是网络资源地址。		服务器收到请求后,会根据请求内容进行处理,并返回一个响应报文,包含了响应状态码、响应头和响应体等信息将对应响应的数据返回给浏览器 get请求报文:所有的参数都会在url地址上显示	获取Web服务器数据	请求行	
					组成 向Web服务器提交数据(注册,姓名,电话号码,提交服务器,保存数据库)	空行	
			HTTP请求报文	post请求报文:参数不会在url地址上显示	组成	请求行 请求头 空行 请求体	
爬虫基础				HTTP响应报文的组成 — 200: 代表请求正常	中应头 空行 响应体		
			HTTP状态码	404:请求资源在服务器不存在 403:程序反爬 500:服务器内部源代码出现错误 其他状态码: https://www.runoob.com/http/http-status-codes.html			
				启动开发者	鼠标右击——检查 快捷键: f12 mac: 快捷键alt+command+i 元素(Elements): 用于查看或修改HTMl标签		
			谷歌浏览器开发者工具使用	标签选项说明	控制台(Console): 执行JS代码 源代码(Sources): 查看静态资源文件,断点调试js 代码		1. 点击Network标签选项 2. 在浏览器地址栏输入百度的网址,就可以看到
					网络(Network): 查看http协议通信过程	使用教程	百度首页的HTTP的通信过程 3. 会刷新出很多的数据包,每一个数据包记录都是请求+响应的一次过程 ***********************************
						界面认识,点击网址查看	日本の
		urllib模块,保存方法——urlretrieve(链接,'文件 路径')					PROTOCKERS TUDESINDER CONCENTRATE SEMBNISH TUDESINDER CONCENTRATE SEMBNISH TUDESINDER CONCENTRATE SEMBNISH TUDESINDER CONCENTRATE SEMBNISH SEMBNISH TUDESIN TUDESI
	03-爬虫请求库	requests模块	第三方库:需要借助pip进行安装: pip install requests/ pip install requests -i 换源地址	import requests	get请求: response = requests.get(url, parmas=data, headers=head)	1. 携带字典数据类型的参数	
				发请求,获取响应结果	post请求:: response = requests.post(url, data=data, headers=head)	代码示例: from=zh&to=en&query=%E7%99%BE%E5%BA %A6&transtype=realtime&simple_means_flag=3 &sign=515480.212649&token=c60f88ffbdf82cb 78c1296527b3b8ef0&domain=common&ts=1706 529029923 格式: data=参数 2. 携带json格式(字符串)的参数 代码示例:	
					参数快速转字典:	{"pageNo":1,"pageSize":15,"categoryCode":"GovernmentProcurement","_t":1706528507000} 格式: json=参数 选中要转为字典的内容 按ctrl+r快捷键	
			requests使用		response.text - 类型: str - 解码类型: requests模块自动根据HTTP 头部 对响应的编码作出有根据的推测,推测的文本编码	● THE THE TOTAL	
				响应结果属性	- 乱码解决:看网页源码的charset=", response.encoding=" response.content:获取二进制数据 `response.url`响应的url;有时候响应的url和请求的url并不一致		
					`response.status_code` 响应状态码 `response.request.headers` 响应对应的请求头 `response.json()`自动将json字符串类型的响应内容转换为python对象(dict or list) 代理ip格式:字典	proxies = { "http": "http://12.34.56.79:9527", "https://12.34.56.79:9527".	
				设置代理IP requests处理不被信任的证书: verify=False	携带代理: response = requests.get(url, verify=False) response = requests.get(url=start_url, headers=headers, allow_redirects=False)	"https": "https://12.34.56.79:9527", } response = requests.get(url, proxies=proxies)	
			内置模块:import csv	禁止重定向	301: 永久重定向,浏览器会记住这个状态码,之后访问这个url都会跳转到新的url 302:临时重定向,不会记得改状态码		
		csv读写模式	一 写入数据	普通的序列写入	数据要有对应的表头 office excel:encoding='utf-8-sig' 1. 创建write对象:csv.write(f) 写入数据	写入一条: writerow(data) 写入多条: writerows(data)	
			读取数据	字典方式写入 创建读取对象 普通序列读取:csv.reader(f)	数据是以{key:value}呈现 csv.DictWriter(f, head)	写入多条: writerows(data)	
		json模块	JSON是一种使用广泛的轻量数据格式 json.loads()> json转为字典(适用于语句) json.dumps()> 字典转为json(适用于语句)	字典方式读取:csv.DictReader(f)			
			json.load()> json转为字典(适用于文件) json.dump()> 字典转为json(适用于文件) json中的数据必须使用双引号包裹		print(jsonpath.jsonpath(data, '\$name'))		
	04-数据提取	— jsonPath模块	XPanh - ASCAPAN ME	的key取对应的内容即可 通配符*,获取老师这个节点对象下所有姓名 ————————————————————————————————————	print(jsonpath.jsonpath(data, '\$teachers[*].name')) 老师年龄大于18 &&并且, 年龄>18并且是广东深圳 : 或者 print(jsonpath.jsonpath(data, '\$teachers[? (@.age>18 or @.addr=="四川成都")]'))		
			语法 使用xpath语法:pip install lxml from lxml import etree—etree.HTML(网页源码) 从根节点获取数据:/	数据组成一组,借助enumerate函数	HATTPARP JJ JJ		
		xpath	从根节点获取数据:/ 不需要考虑位置获取节点:// 选取属性: @ 选择当前元素的父节点				
			/text(): 获取元素的文本内容 pip install bs4 解析器:lxml, html5lib(都需要安装) from bs4 import BeautifulSoup				
		bs4	soup = BeautifulSoup(网页源码,解析器) 根据需求调用方法	find():返回搜索到的第一条数据 find_all (),以列表的形式返回所有搜索到的数据 string: 获取标签内的文本内容			
			遍历文档树	strings:返回多个标签的内容 stripped_strings: 可以将多余的空去掉			
			select()	class="info"——.info id="info"——#info			