

- MySQL设置远程连接

【1】修改配置文件,使MySQL允许远程连接

1.1) `sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf`  
    # bind-address = 127.0.0.1 此行注释掉

1.2) `sudo /etc/init.d/mysql restart`

【2】添加授权用户,使用授权用户进行远程连接

2.1) `mysql -uroot -p123456`

2.2) `mysql>grant all privileges on *.* to 'root'@'%' identified by '123456' with grant option;`

2.3) `mysql>flush privileges;`

【3】连接测试

`mysql -h192.168.100.128 -uroot -p123456`

- Windows安装Redis并添加到系统服务中

【1】从网上下载redis的windows的安装包

【2】直接解压就可以使用

【3】将解压的路径添加到系统环境变量中

右键 - 此电脑 - 属性 - 高级系统设置 - 环境变量 - Path - 编辑 - 添加redis的本地路径

【4】添加到系统服务

4.1) 把E:\王伟超\Redis\redis.windows.conf 重命名为: redis.conf

4.2) 在cmd中进入到redis的路径 : E:\王伟超\Redis

4.3) `redis-server --service-install redis.conf --loglevel verbose`

4.4) 启动redis: 右键-此电脑-管理-服务和应用程序-服务-找到Redis-双击-点击启动

- Redis设置远程连接

【1】修改配置文件,允许redis进行远程链接

1.1) `sudo vi /etc/redis/redis.conf`  
    # bind 127.0.0.1 ::1 此行注释  
    protected-mode no     默认为yes,需要改为 no

1.2) `sudo /etc/init.d/redis-server restart`

- 请求模块

【1】requests

`html = requests.get(url=url,headers=headers).text`

`html = requests.post(url=url,data=data,headers=headers).text`

参数: url headers proxies timeout verify cookies

【2】urllib

2.1) 请求模块: `urllib.request`

2.2) 编码模块: `urllib.parse`

【3】urllib.request模块使用

3.1) `res = urllib.request.urlopen(url=url)`

`html = res.read().decode() # read() -> bytes`

3.2) `urlopen()`方法不支持包装 headers

urllib.request正常使用流程

- 1) 创建请求对象(包装User-Agent) - Request()
- 2) 发请求获取响应对象 - urlopen()
- 3) 获取响应内容 - read().decode()

## • 三种xpath

### 【1】lxml + xpath

```
from lxml import etree
p = etree.HTML(res.text)
r_list = p.xpath('xpath表达式')
1.1) 节点对象列表: //div/a //ul/li
1.2) 字符串列表: //div/a/text() //ul/li/a/@href
```

### 【2】selenium - 得到的永远是节点对象

```
from selenium import webdriver
driver = webdriver.Chrome(executable_path='/xx/xx')
2.1) 节点对象列表: driver.find_elements_by_xpath('')
2.2) 节点对象: driver.find_element_by_xpath('')
特点: //div/a //ul/li /div//p
2.3) 如何获取文本内容字符串??? node.text 属性
2.4) 如何获取节点的属性值??? node.get_attribute('href')
```

### 【3】scrapy中xpath

爬虫文件: baidu.py

```
import scrapy
class BaiduSpider(scrapy.Spider):
    name = 'baidu'
    allowed_domains = ['www.baidu.com']
    start_urls = ['http://www.baidu.com/']
    def parse(self, response):
        response.xpath('xpath表达式')
3.1) 结果一定为列表
3.2) 情况1: 列表中存放大量节点对象
    '//ul/li' [<li1>,<li2>,...,<li>]
3.3) 情况2: 列表中存放大量选择器对象
    '//ul/li/text()' '//ul/li/a/@href'
    [
        <selector xpath='xx' data="百度一下">,
        <selector xpath='xx' data="你就知道">,
    ]
extract() : ['百度一下','你就知道']
extract_first() : '百度一下'
get() : '百度一下'
```