聚焦爬虫：爬取页面中指定的页面内容

-编码流程：
  -指定url
  -发起请求
  -获取响应数据
  -数据解析
  -持久化存储

数据解析分类：
-正则
-bs4
-xpath（***最通用）

数据解析原理：
-解析的局部的文本内容都会在标签之间或者在标签对应的属性中进行存储
-1. 进行指定标签的定位
-2. 标签或标签对应的属性中存储的数据值进行提取（解析）

bs4进行数据解析：

-数据解析的原理：
-1.标签定位
-2.提取标签、标签属性中存储的数据值

-bs4数据解析的原理：
-1.实例化一个BeautifulSoup对象，并且将页面源码数据加载到该对象中
-2.通过调用BeautifulSoup对象中相关的属性或者方法进行标签定位和数据提取

-环境安装
-pip install bs4
-pip install lxml

-如何实例化BeautifulSoup对象：
-导包（from bs4 import BeautifulSoup）
-对象实例化：（两种可能）
-1.将本地html文档中的数据加载

到该对象

-2.将互联网上获取的页面源码加载到该对象中

soup = BeautifulSoup(文件名,"lxml") 参数2️⃣固定为lxml

-提供的用于数据解析的方法和属性

-soup.tagName返回的是html中第一次出现的tagName

-soup.find（）：两种用法

-soup.find_all（）：返回所有符合要求的标签（列表）

- select:

-select（'某种选择器'，返回的是一个列表

-层级选择器:

-soup.select('.tang > ul > li > a'): >表示的是一个层级

-soup.select('.tang > ul a'): a不为ul的直接下一层级，直接用空格间隔开即可

-获取标签之间的文本数据:

-soup.a.text/string/get_text()

-text/get_text():可以获取某一

个标签中所有的文本内容
-string：只可以获得该标签下面直系的文本内容
-获取标签中属性值
-soup.a['href']

xpath解析：最常用且便捷高效的一种解析方式。通用性。

-xpath解析原理：
1.实例化一个etree对象，且需要将被解析的页面源码数据加载到该对象中
2.调用etree对象中的xpath方法结合着xpath表达式实现标签的定位和内容的捕获

-xpath环境安装：
-pip install lxml

-如何实例化etree对象 |from lxml import etree|
1.将本地html加载入对象
etree.parse(filepath)

2.将互联网上获取的源码数据加载入对象

etree.HTML('page_txt')

3. xpath('xpath表达式')

-xpath表达式

- /:表示的是从根节点开始定位；表示的是一个层级
- //:表示的是多个层级；可以表示从任意位置开始定位
- 属性定位：//div[@class="song"]

tag[@attrName="attrValue"]

- 索引定位：//div[@class="song"]/p[3]（索引从1开始）
- 取文本：
  1. /text() 获取的标签中直系的文本内容
  2. //text() 获取的标签中非直系的文本内容
- 取属性：

  /@attrName ==> img/@src