该爬虫采用Scrapy框架，并结合其他模块编写而成。

遇到的问题：

1. 多次请求之间的传值问题：

Scrapy 用scrapy.Request发起请求可以带上 meta={'item': item} 把之前已收集到的信息传递到新请求里，在新请求里用 item = response.meta('item') 接受过来，在 item 就可以继续添加新的收集的信息了。多少级的请求的数据都可以收集。

传值：yield scrapy.Request(item['url'], meta={'item': item},callback=self.detail\_parse)

取值：# 接收上级已爬取的数据

item = response.meta['item']

1. 指定了项目目录，命令才能生效。sys.path.append(os.path.dirname(os.path.abspath(\_\_file\_\_)))
2. 在scrapy中可以用xpath和css来提取数据，但感觉每种法并不是万能，根据网页的结构，选取合适的方法很重要。也可以结合第三方库，如BeautifulSoup和re, 但相较于xpath和css, BeautifulSoup的速度更慢。
3. Return和yield的区别：

yield是用于生成器。什么是生成器，你可以通俗的认为，在一个函数中，使用了yield来代替return的位置的函数，就是生成器。它不同于函数的使用方法是：函数使用return来进行返回值，每调用一次，返回一个新加工好的数据返回给你；yield不同，它会在调用生成器的时候，把数据生成object，然后当你需要用的时候，要用next()方法来取，同时不可逆。