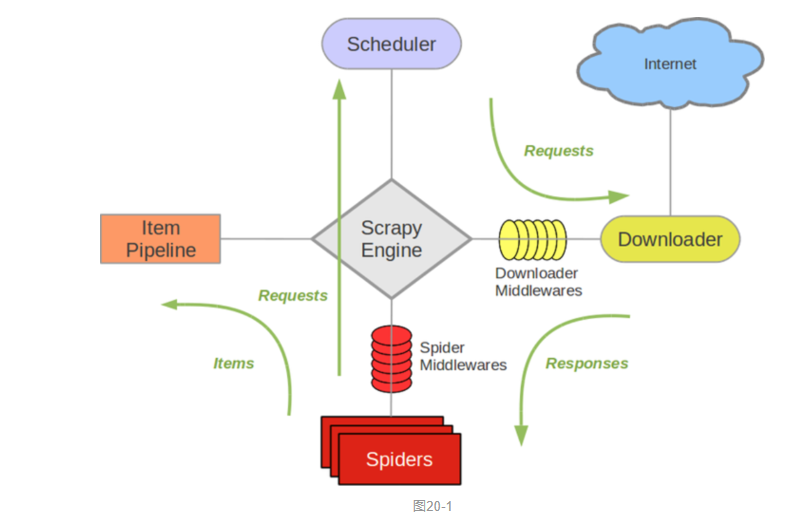
本项目使用scrapy框架，从途牛网爬去景点相关数据并将信息写入MySQL数据库。由于使用了框架，这一过程变得较为简单快捷。

在Scrapy框架中，Item Pipeline是处理数据的组件，如下图20-1所示，当Item在Spider中被收集之后，将会被传递到Item Pipeline，每一个组件会按照一定的顺序执行对Item的处理。每个Item Pipeline是实现了简单方法的Python类。他们接收到Item并通过它执行一些行为，同时也决定此Item是否继续通过pipeline，或是被丢弃而不再进行处理。



以下是实验步骤：

 在CMD中，进入你想要存储代码的目录下执行：scrapy startproject myspiders，其中quotes可以是你想要创建的目录名字。

 Scrapy会自动创建一个名为myspiders的目录，并在它里面初始化一些内容。

 进入myspiders/spiders目录，新建一个名为quotestoscrape.py的文件，并添加如下代码

保存后，切回CMD，执行scrapy crawl quotestoscrape，先简单解释一下这部分的代码：

* 首先经过我的测试start\_requests(self)这个方法并不是必须的，至少它也可以是一个名为start\_urls[]的列表。不过我觉得还是遵循某种标准写法比较好。如果有的话，按照文档的说法，必须返回一个Requests的迭代器（它可以是一系列请求也可以是一个生成迭代器的方法），它代表了这个爬虫要从哪个或哪些地址开始爬取。同时也会同来进一步生成之后的请求。
* 每条请求都会从服务器下载下来一些内容，parse()方法是用来处理这些内容的。参数response包含了整个页面的内容，之后你可以使用其他函数方法来进一步处理它。
* yield关键字代表了Python另一个特性：生成器。
* 指令执行后，都会输出一大堆的log，大多数不难理解，我这里只截取其中我们想看的一部分，其中前半部分是爬取到的结果，后面一部分是一个统计：

进一步处理response

初次接触爬虫，可能会对上述代码中的response.css(), quote.css(), quote.xpath()和extract\_first()感到陌生，这些就是所谓的进一步处理response的方法。

这部分内容需要用到一些HTML/CSS的知识，你需要知道通过怎样的表达式才能从返回内容中获取到你需要的内容。因为网页的代码都是树形结构，理论上通过合理的表达式，我们可以获取任何我们想要获得的内容。通常情况下，我们有两种方法可以计算出我们的表达式：

* 第一种是用浏览器的审查模式。
* 第二种是利用scrapy提供的命令行模式。

CSS选择器

上述代码中，response.css('div.quote')和quote.css('span.text::text')都是CSS选择器。如果我们打开该网页的元素审查页面，会有如下结果：

依我之见，流程大概如下：利用屏幕底下几个标签可以先定位到一个大概的位置，比如说quote = response.css('div.quote')定位到图中蓝框的位置，之后我们要进行进一步的筛选，我没有找到文档说明应如何进行筛选，这里是我的一点经验之谈：如果是html标签用空格分割，如果标签带class标识，则用.连接，最后再加上::text 用来剔除首尾的<>标识。

在整个过程中，我们都可以用scrapy的命令行来测试，在你的CMD下输入：scrapy shell "http://quotes.toscrape.com/"。之后出现一大推日志和一些可用的指令：

我们主要用到的是response对象，之后我们就可以进行调试，如下：

如果你之前自己写过网站的CSS，这些其实还是很好理解的，因为内在的逻辑是一样的，伴随这个命令行指令自己琢磨琢磨很容就就能掌握。如果你仔细看，会发现这个函数返回的其实是个列表，这点可以方便我们写代码。

XPath选择器

另一种方法是使用XPath选择器，如上文中的代码：quote.xpath('span/small/text()')。根据文档的描述，XPath才是Scrapy的基础，事实上，即使是CSS选择器最终也会在底层被转化为XPath。XPath比CSS选择强大的地方在于它还可以对筛选出的网页的内容本身就行操作，比如说它可以进行诸如选择那个内容为（下一页）的链接的操作。官方提供了三个关于XPath的文档：[using XPath with Scrapy Selectors](https://doc.scrapy.org/en/latest/topics/selectors.html" \l "topics-selectors" \t "_blank)，[learn XPath through examples](http://zvon.org/comp/r/tut-XPath_1.html" \t "_blank)和[how to think in XPath](http://plasmasturm.org/log/xpath101/" \t "_blank)。

保存数据

这个只是一行命令的事，比如说我要将上文爬虫的内容写入一个json文件，我只需要在cmd中执行：

-o应该就是output，这个linux命令很像，不难理解。当然也可以是其他格式的文件，官方推荐一个叫[JSON Lines](http://jsonlines.org/" \t "_blank)的格式，虽然我目前还不知道这是什么格式。

所有指出的到处数据类型为：'json', 'jsonlines', 'jl', 'csv', 'xml', 'marshal', 'pickle'。