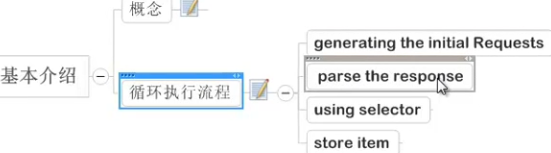
Scrapy框架的重要组件

# Spider组件

## 概念：



## 循环执行流程

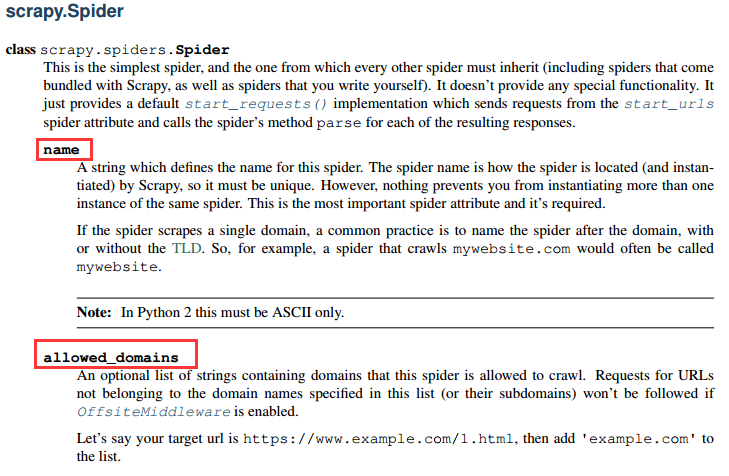


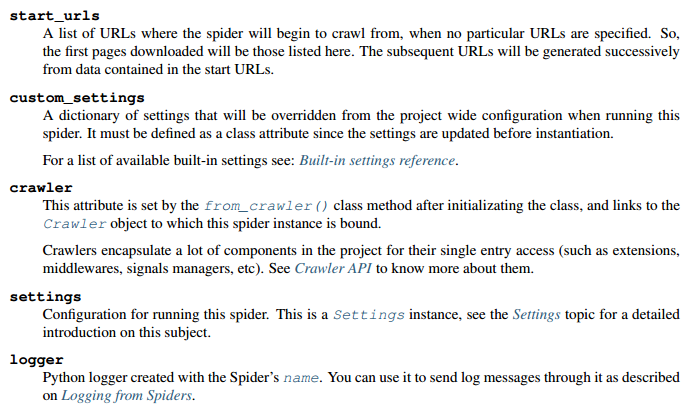
## 基类scrapy.Spider ：

### scrapy.Spider类的属性



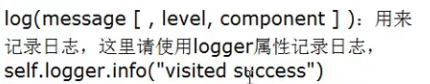
custom\_settings会覆盖settings.py 的全局设置信息。



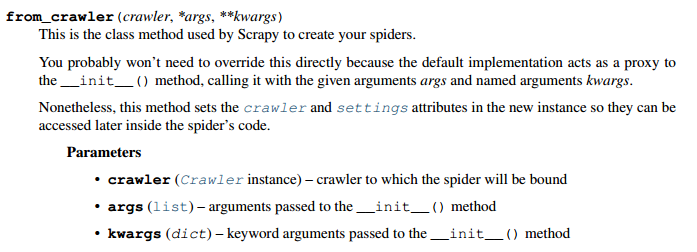


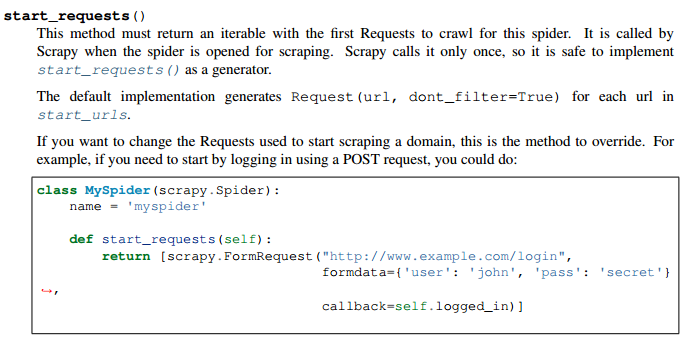
### scrapy.Spider类的方法

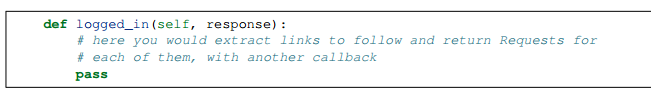


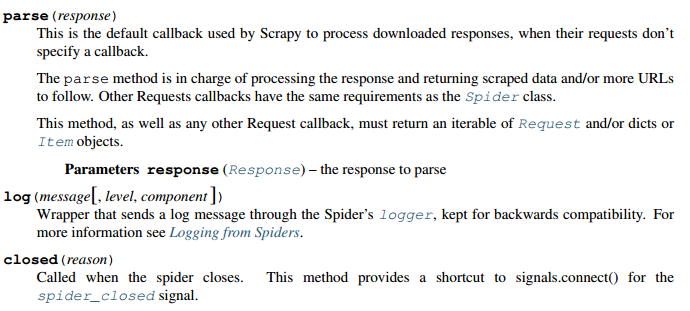


这个log()方法是为了与之前的版本兼容，所以以后尽量不要使用log方法，建议使用self.logger.info()等方法。

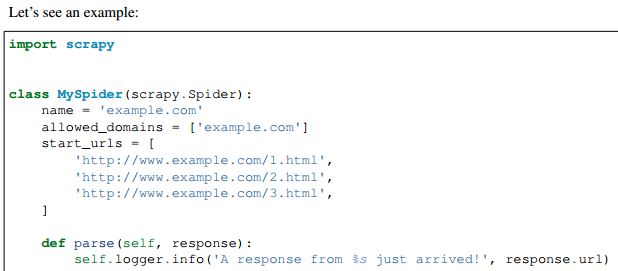


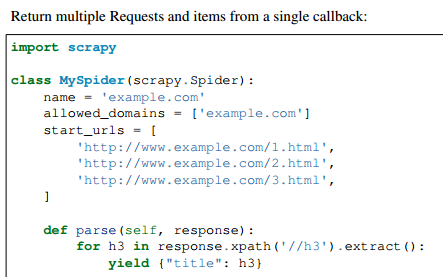






**示例：**

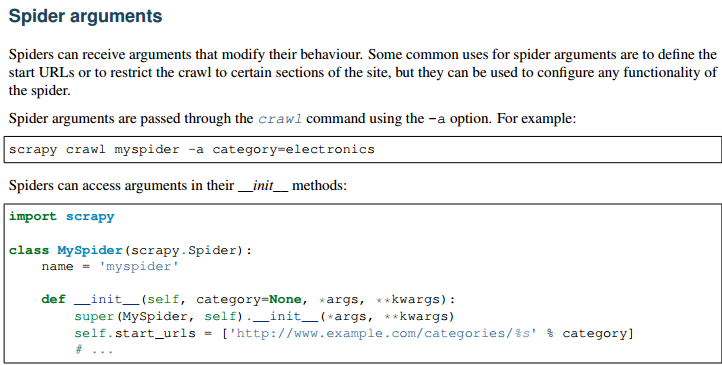


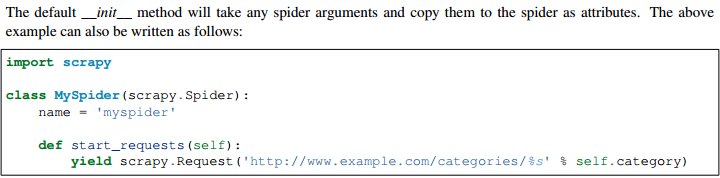


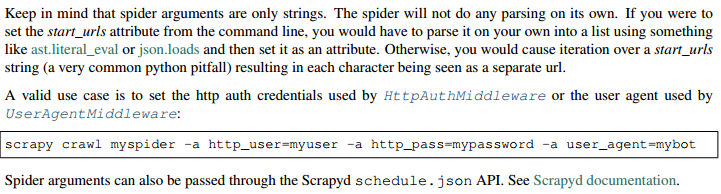




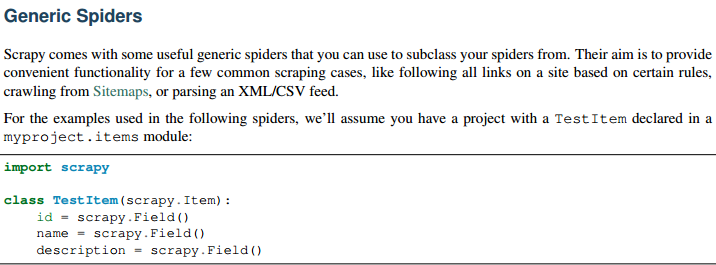
### Spider 参数



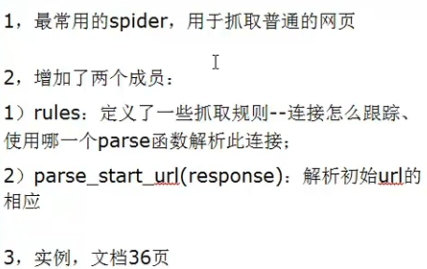


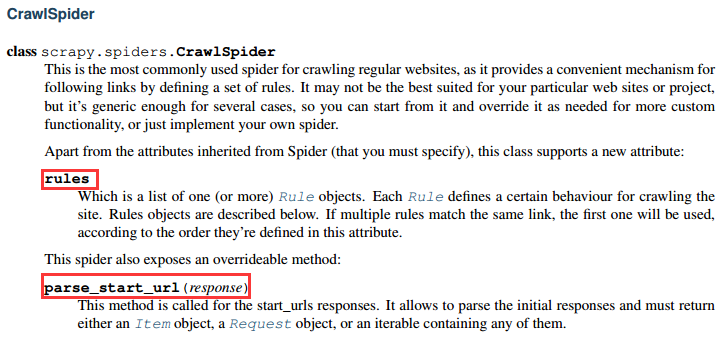


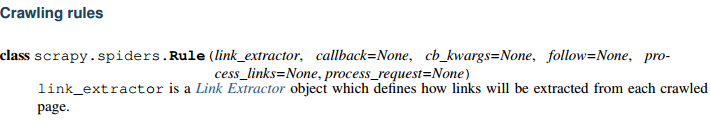
## 子类介绍

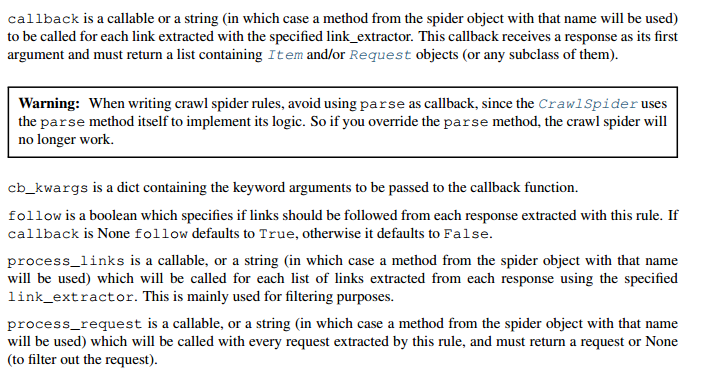


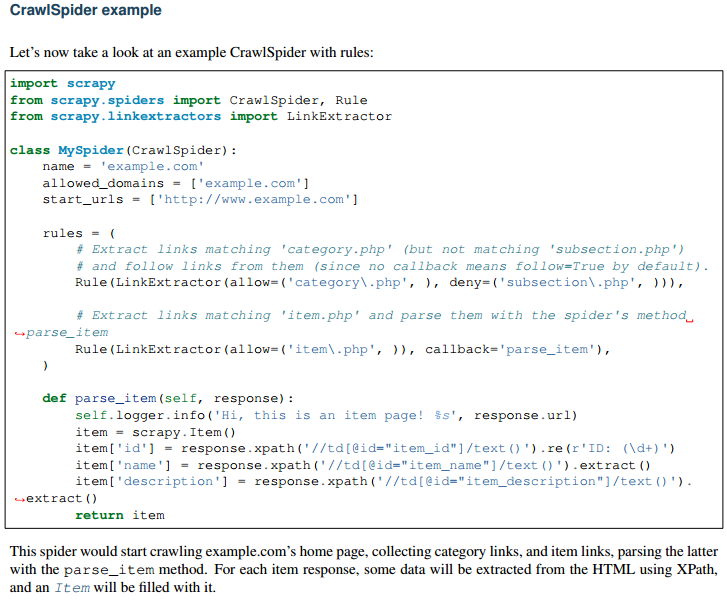
### CrawlSpider

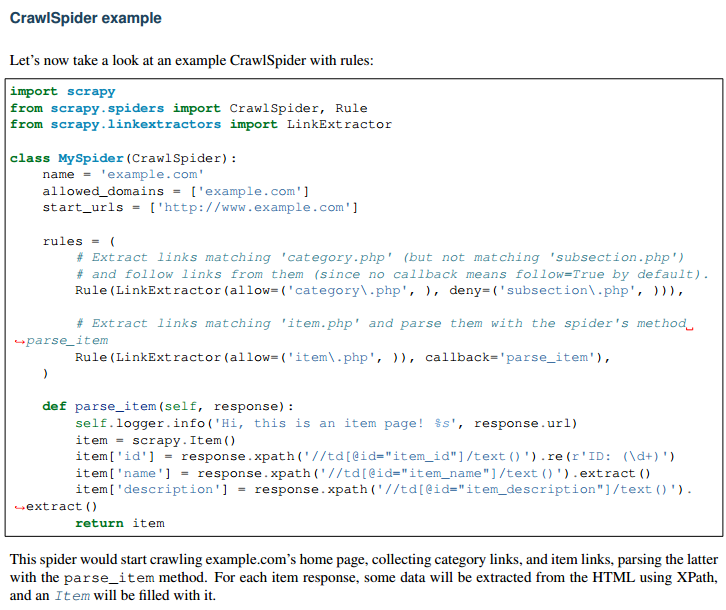




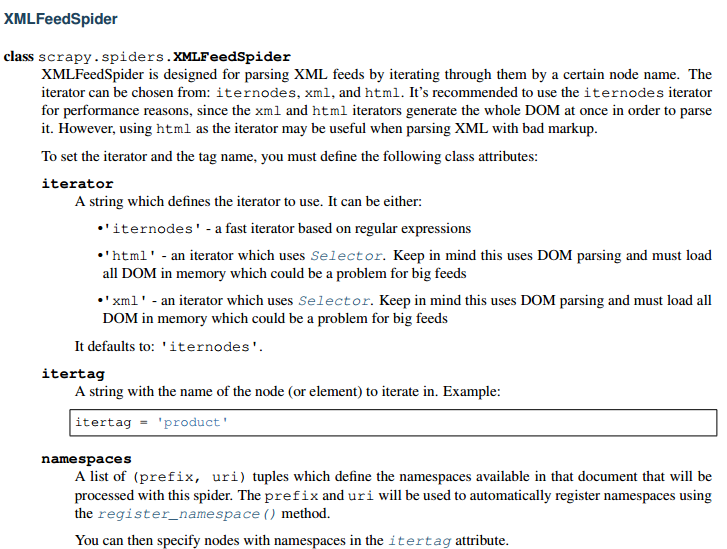


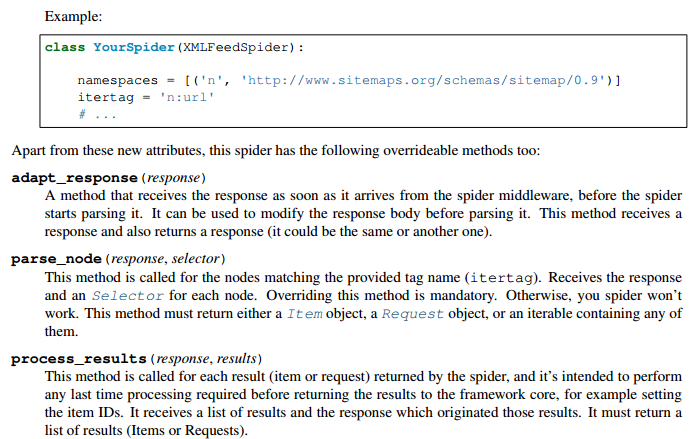


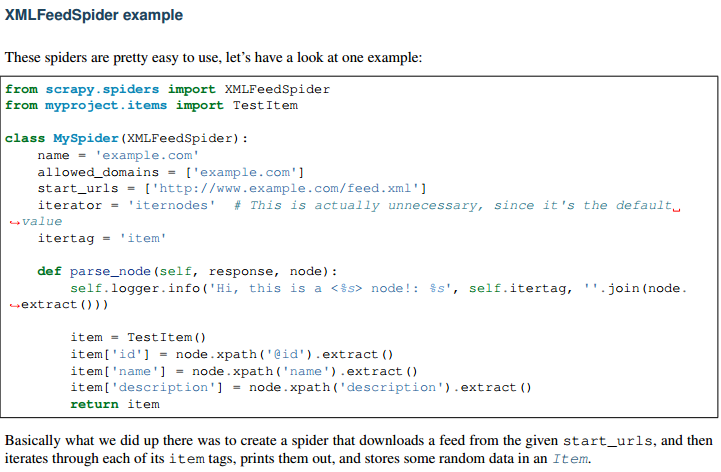




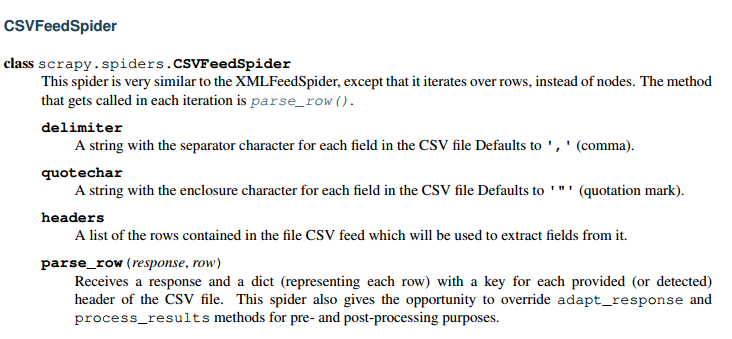
### XMLFeedSpider

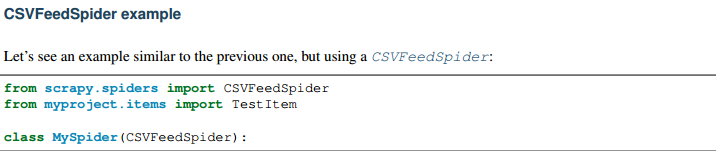


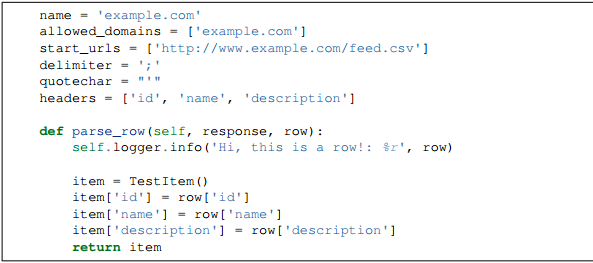




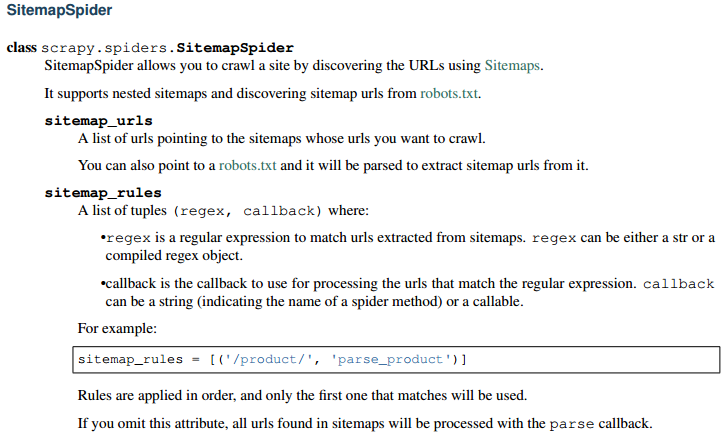
### CSVFeedSpider

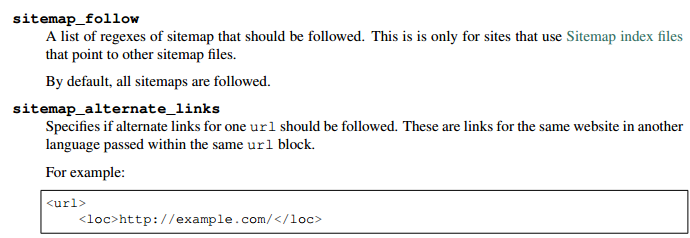


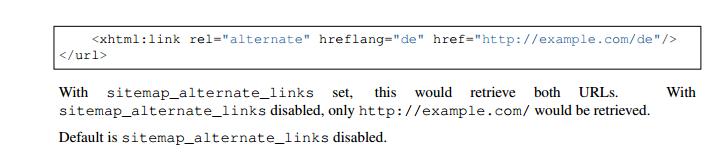




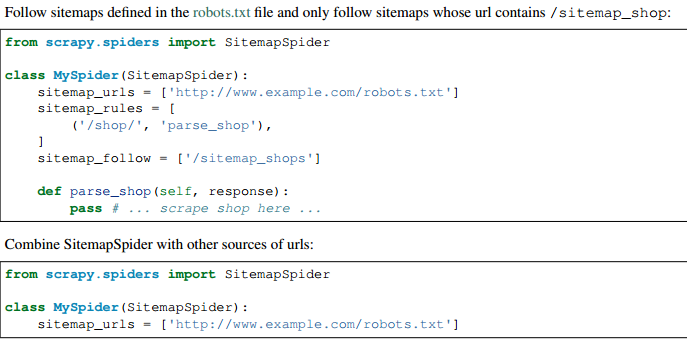
### SitemapSpider

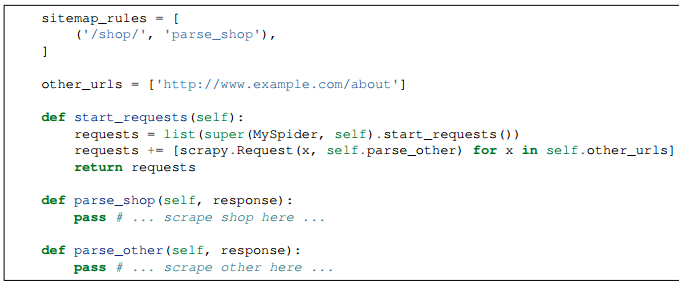








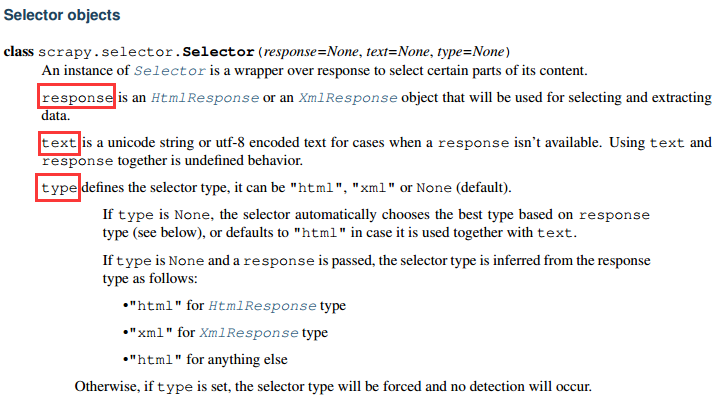




# Selector

## Selector Objects

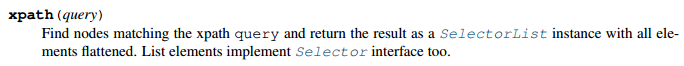
属性：**response、text、type**

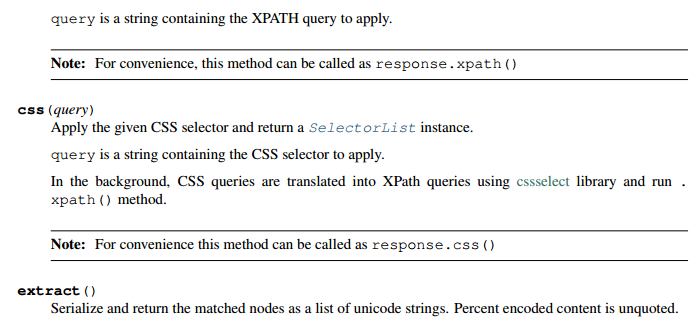


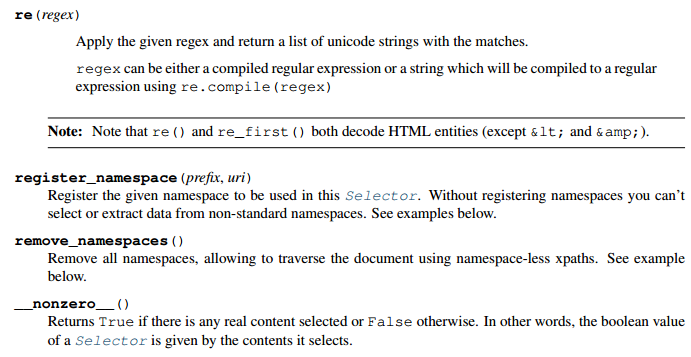
方法：xpath()、css()、extract()、re()

其中，xpath()与css()方法返回的都是**SelectorList类型**；

re()与extract()返回的是**字符串的List类型**。







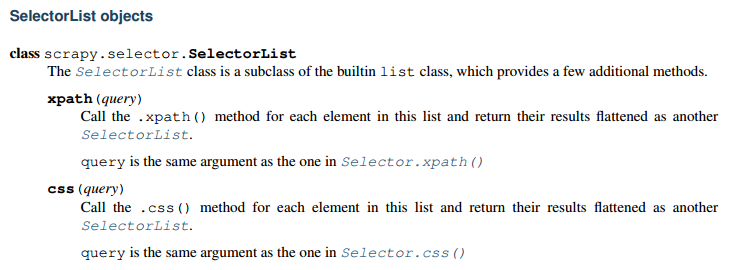
## SelectorList Objects

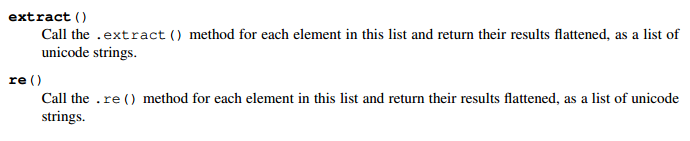
SelectorList是list的一个子类，外加的方法有：

方法： **xpath()、css()、extract()、re()**

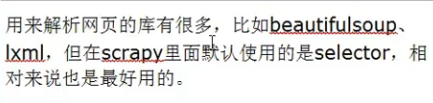
**xpath()与css()方法返回的是SelectorList类型；**

**extract()与re()方法返回的是字符串的list类型。**

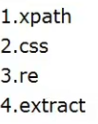


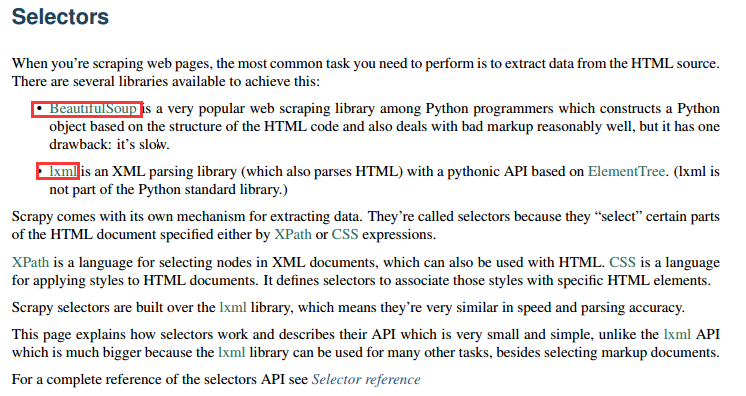


## Selector介绍

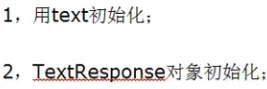


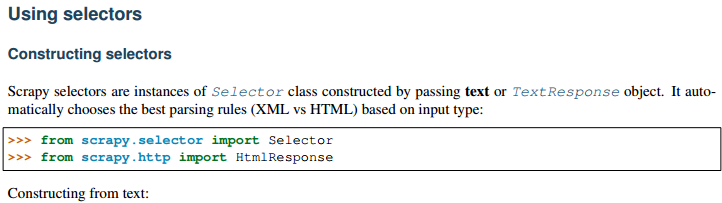
抽取时的常用方法：

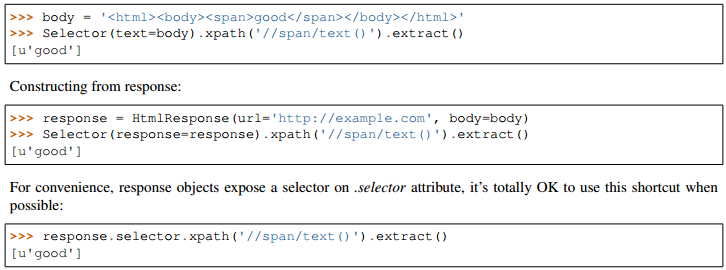




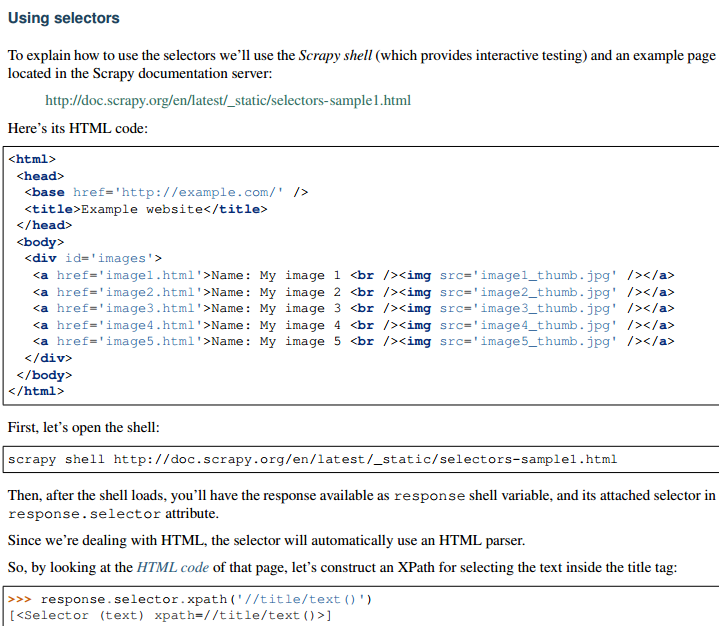
### 创建Selector对象的方法







### 使用Selector



### 嵌套使用Selector

由于css()与xpath()方法都是返回的SelectorList，所以可以直接嵌套抽取所需要的数据。

## 为了方便，可以直接利用Response调用xpath()与css()方法

Selector(response=response).xpath(‘\*\*\*\*\*\*’) 🡸🡺 response.xpath()

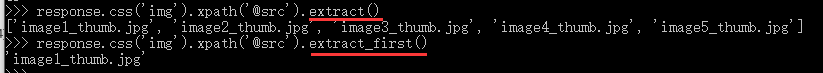
Selector(response=response).css(‘\*\*\*\*\*\*’) 🡸🡺 response.css()

功能完全一样，返回的都是SelectorList对象，就是简单了。

## extract()与extract\_first()方法

如果是一个SelectorList，调用extract则返回一个string的list；

如果只想获取该list的第一个string，则利用extract\_first()即可。



如果SelectorList是空的，这时候调用extract()方法，返回的是一个空的list，但是extract\_first()则不返回任何值，但是想让其返回一定提示信息时，可以添加参数**default=’not found’**

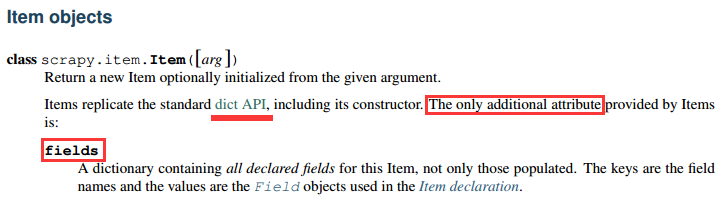
如：response.css('imgs').xpath('@src').extract\_first(default='not found')



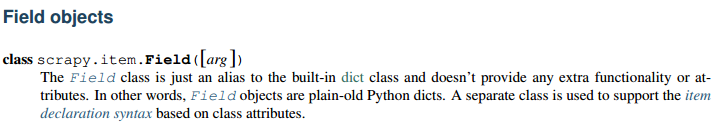
# Items

## Item Objects

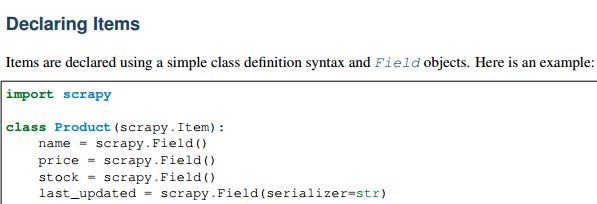
**Item是个字典dict**，外加的唯一的一个属性，就是**fields**。



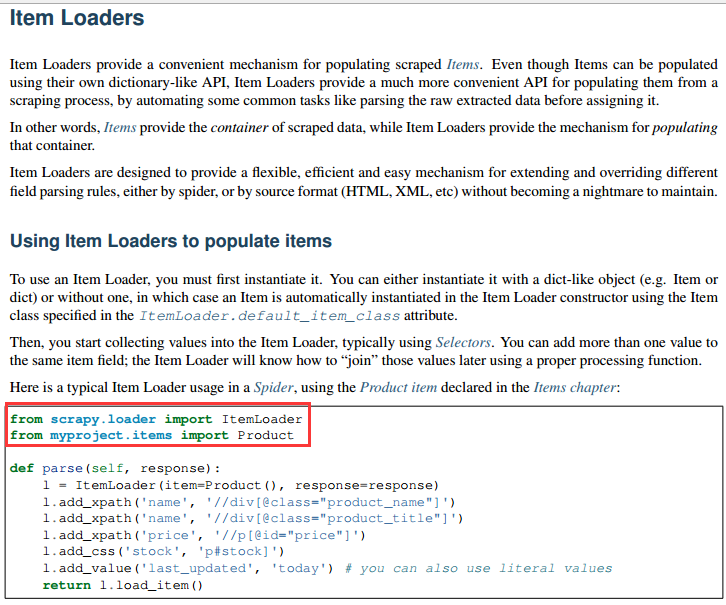
## Field Objects



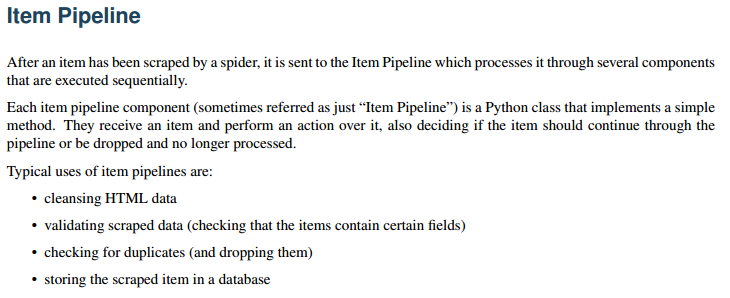
## Declaring Items 声明Items

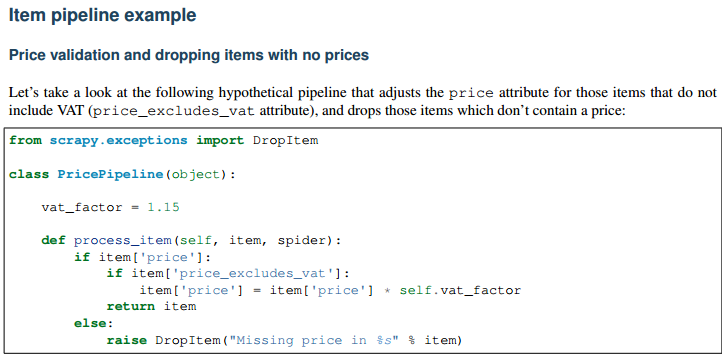


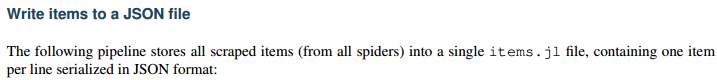
## Items Loaders

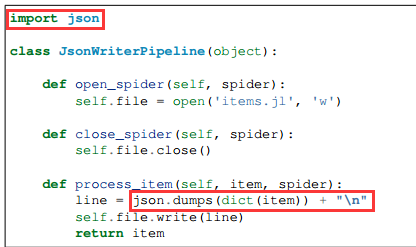


# Item Pipeline

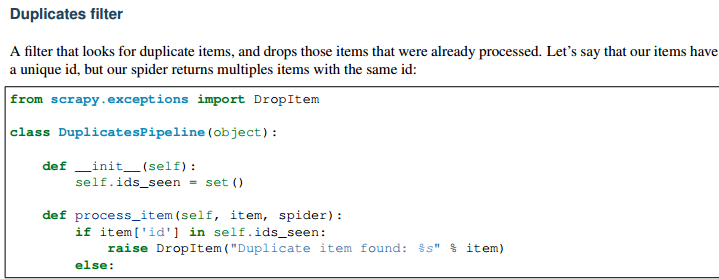




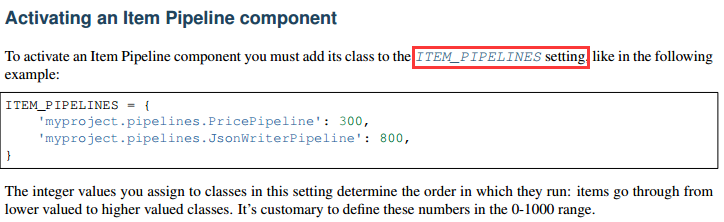






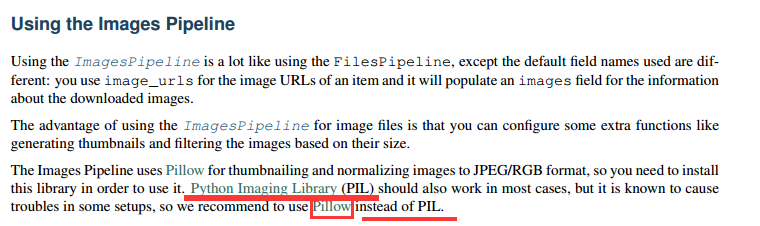






内置的Pipeline：

## ImagesPipeline的介绍及使用方法：



使用**ImagesPipeline**之前，必须安装Pillow。

pip install pillow

其余的需要在settings.py 文件中设置：











总结：

**在settings.py 中至少三个参数：**

**ITEM\_PIPLINES={‘’}**

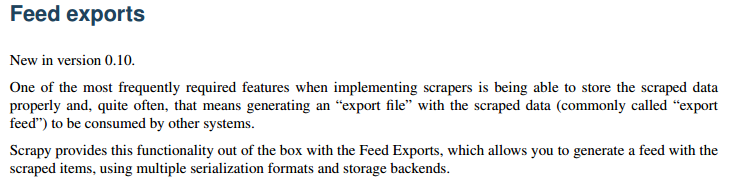
**IMAGES\_URLS\_FIELD=’’**

**IMAGES\_STORE=’’**

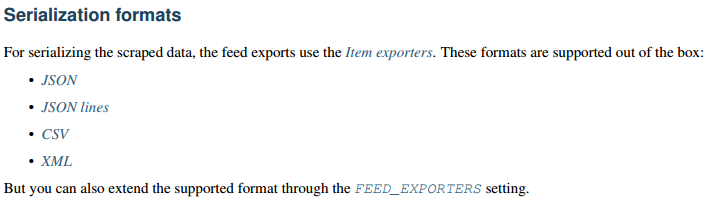
## FilesPipeline

使用方法与**ImagesPipeline**类似。

# Feed Exports



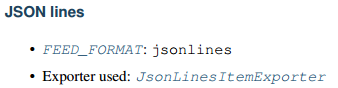
## Serialization formats



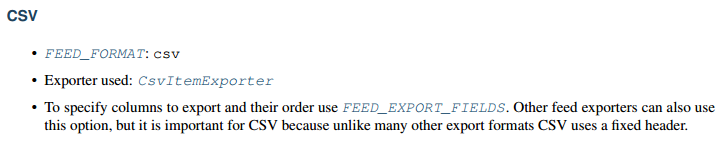
### JSON



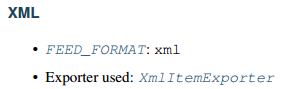
### JSON lines



### CSV

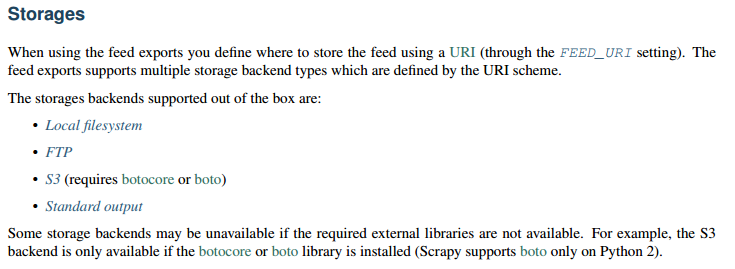


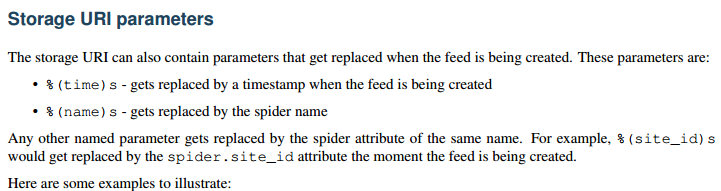
### XML

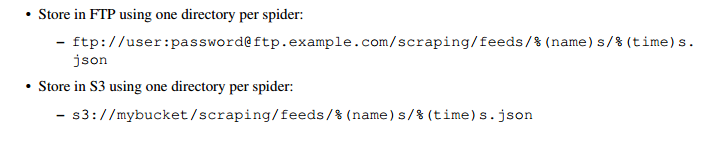




## Storages

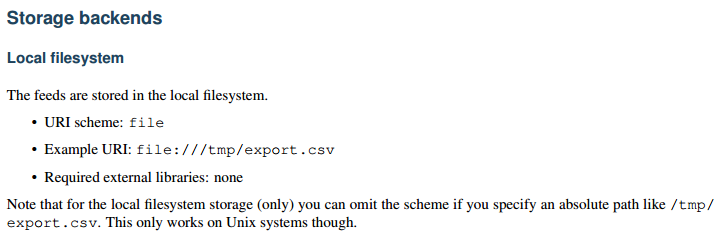




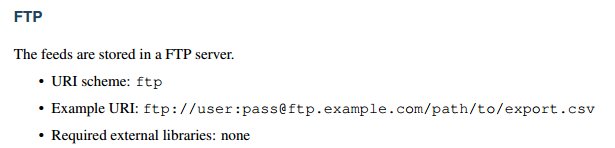


## Storage backends

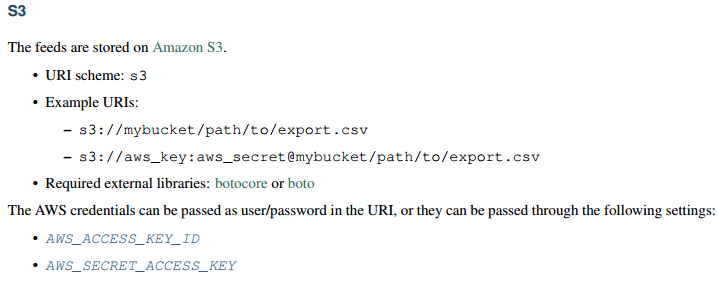
### Local filesystem



### FTP



### S3



### Standard output



## Settings

