Scrapy框架的命令行的常用命令

# Scrapy命令行工具命令分类

（对应Scrapy文档的第三章第一部分3.1 **Command line tool**）

Scrapy提供了两种类型的命令。一种必须在Scrapy项目中运行(针对项目(Project-specific)的命令)，另外一种则不需要(全局命令)。**全局命令在项目中运行时的表现可能会与在非项目中运行有些许差别(因为可能会使用项目的设定)。**

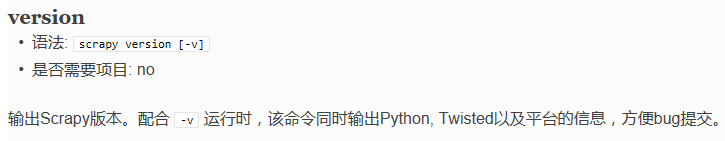
# scrapy命令行工具命令分类

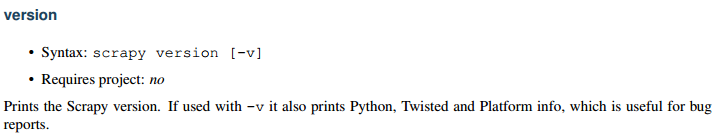
## 全局命令

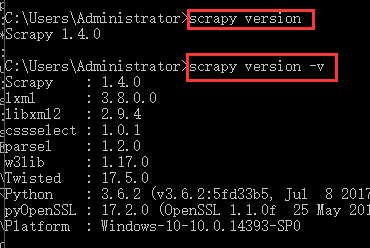
**version、startproject、genspider、view、shell、settings、runspider、fetch**

### scrapy version 及 scrapy version –v

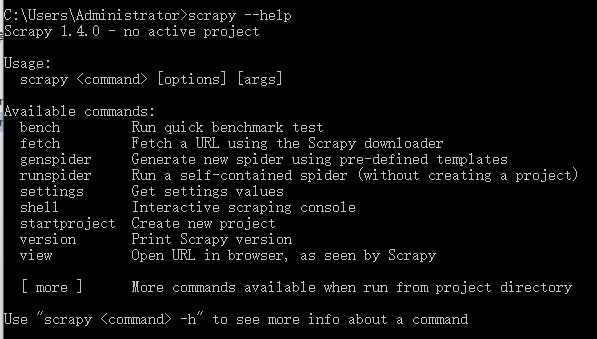
查看scrapy版本信息，添加-v 可以查看scrapy的各个组件的版本信息。







### scrapy –help：用于查看帮助信息。



### scrapy startproject projectName：创建一个工程。

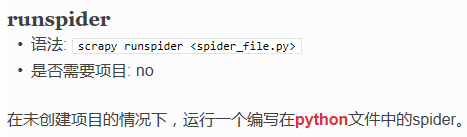
见下面详细介绍。

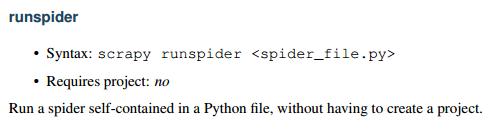
### scrapy genspider spiderName

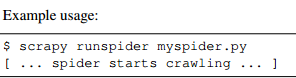
不需要进入项目目录。见下面介绍。就是用来生成一个特定的spide.py文件，完全可以自己创建这个文件，并书写相应的代码。

### scrapy runspider *spiderName.py*

**作用：可以运行任意位置的spider文件。**

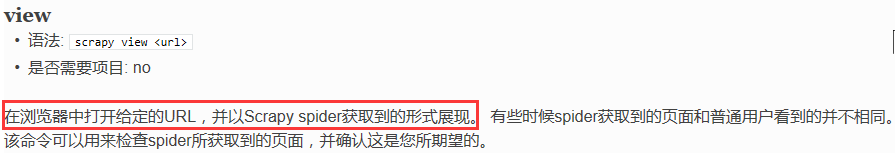


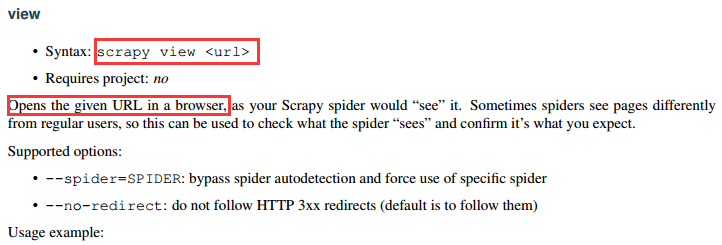




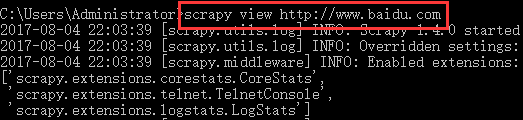
### scrapy view <http://www.baidu.com>

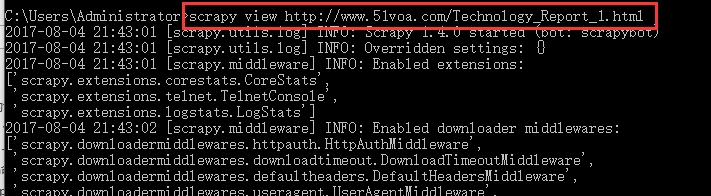
**作用： 用于查看页面源码在浏览器中显示的样子，用于验证获取的信息是否与源码一致。自动在默认浏览器打开网页。**











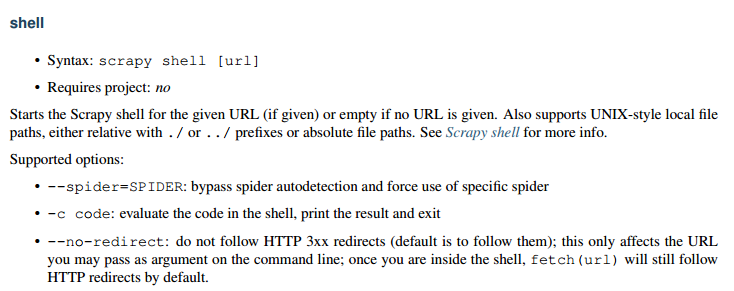
### scrapy shell <http://www.baidu.com>

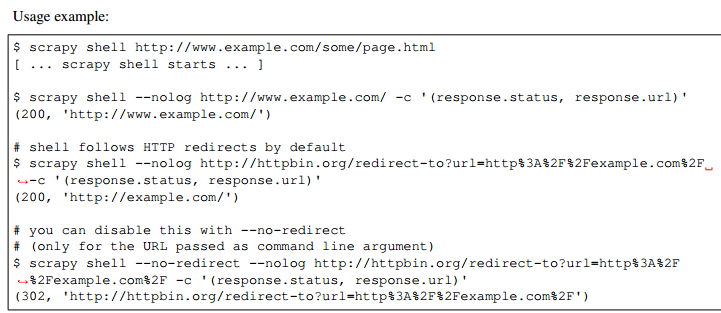
**一个非常有用的命令**。

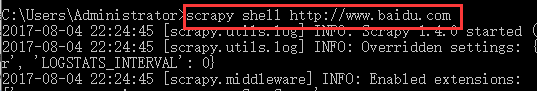
**作用**： **可用于获取对应网址信息，返回很多对象，接下来可以利用这些对象进行操作。主要是用于调试数据、检测xpath、查看页面源码等**。

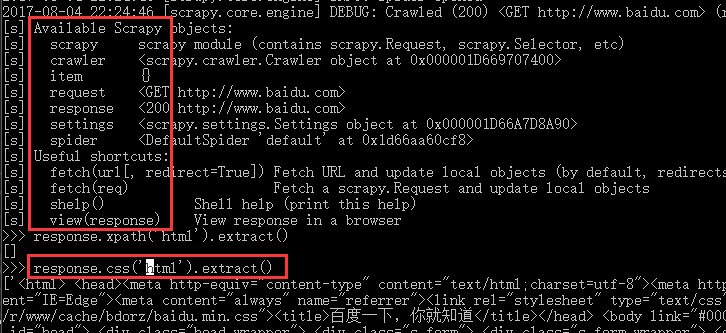
其中，通常使用最多的就是Response对象，用来检验提取信息是否正确。

利用exit()函数退出当前shell。

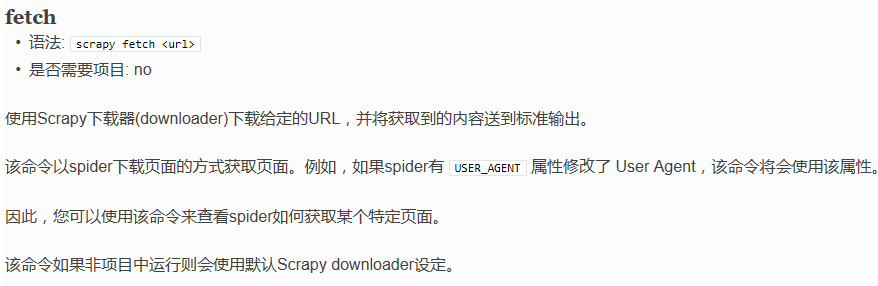


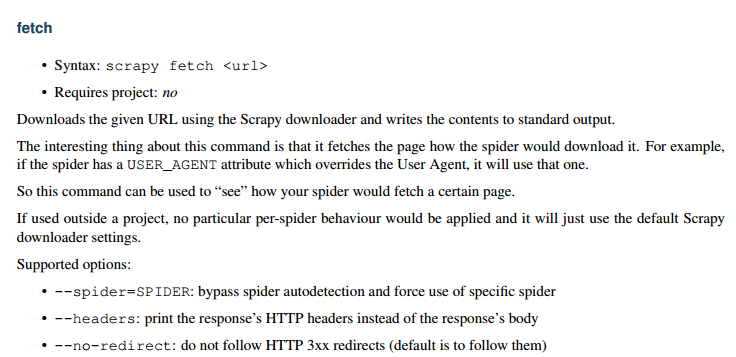


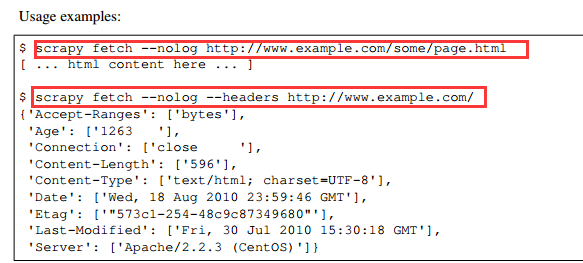




### scrapy fetch <http://www.baidu.com>



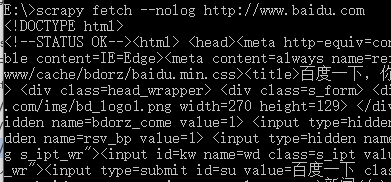




**先打印日志，最后把结果打印出来。**

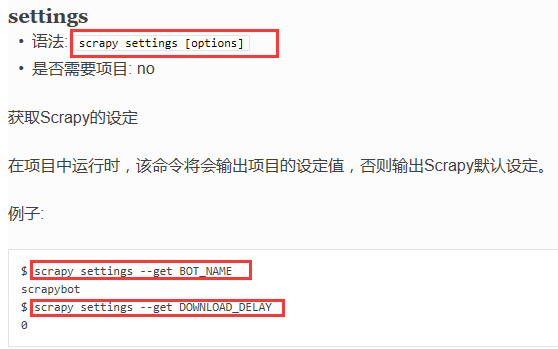
**scrapy fetch --nolog** [**http://www.baidu.com**](http://www.baidu.com) **不打印日志，只打印结果。**

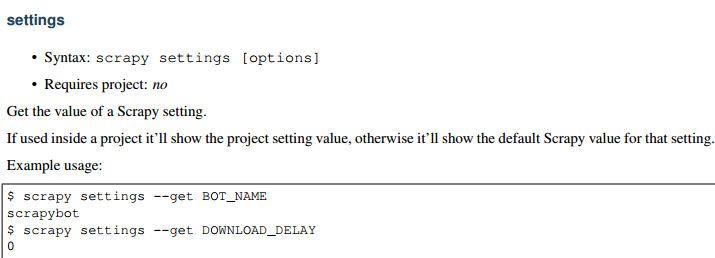
**scrapy fetch --nolog** [**http://www.baidu.com**](http://www.baidu.com) **不打印日志，只打印headers信息。**





### scrapy setting [options]



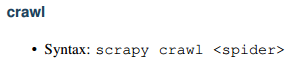


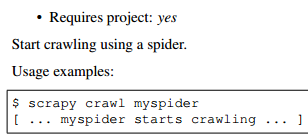
## 项目工程命令：必须进入项目目录下才能使用。

**crawl、check、list、parse、edit、deploy、bench**

### scrapy crawl *spiderName*

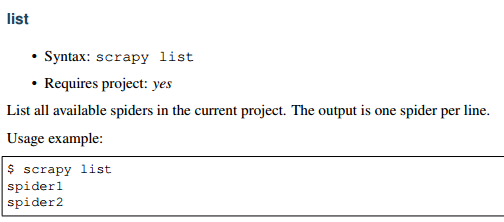
**使用spider进行爬虫。需要在项目目录下执行才有效。**

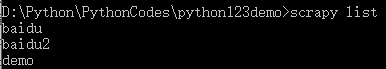


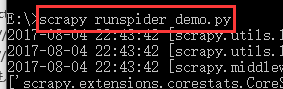


### scrapy list：列出当前项目目录下的所有的spider。

当忘记了具体的spider名字时，可以利用list列出所有的spider。



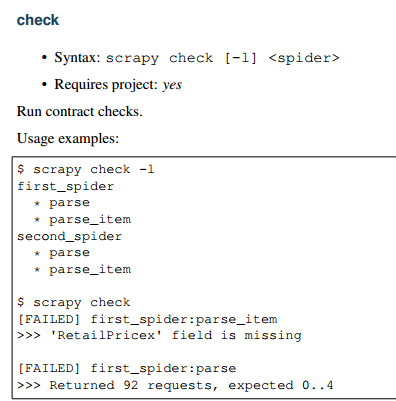




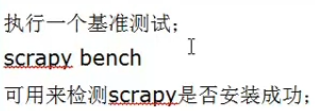
### scrapy parse ...

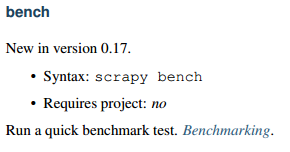
见后面详解。

### scrapy check



### scrapy bench

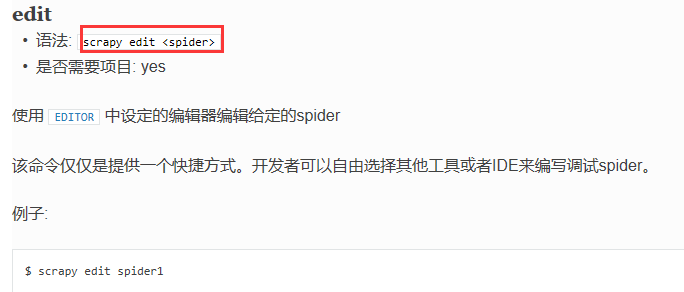


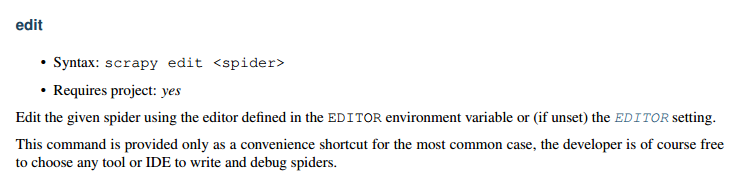




既可以在**任意目录下**执行，也可以在项目下进行，作用效果不一样。

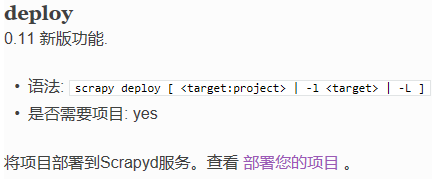
### scrapy edit <spider>







### scrapy deploy

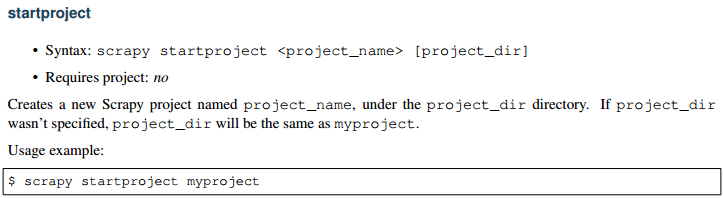


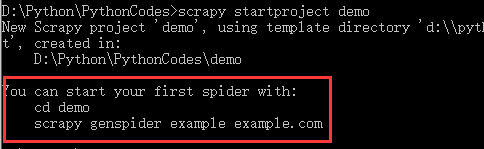


# startproject 及genspider：创建一个工程及spider

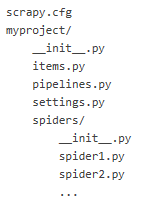
## scrapy startproject

**scrapy startproject demo 创建了一个名称为demo的工程，并会产生一个完整的scrapy工程目录结构。**





**默认的项目目录结构：**

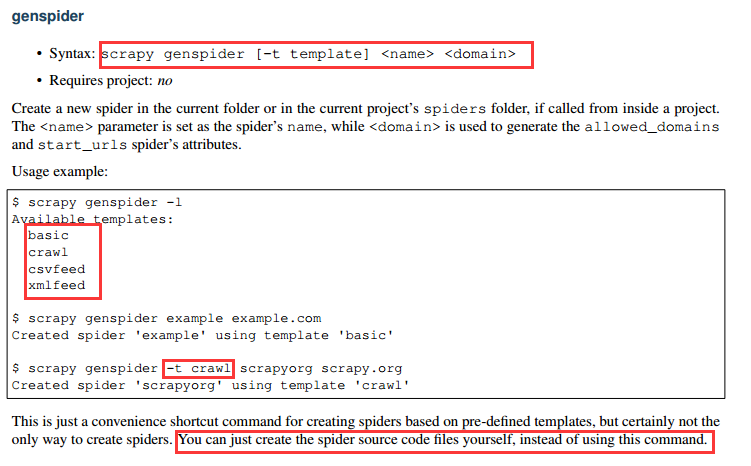


## scrapy genspider

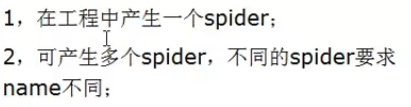
**gen** : 是单词generate的缩写，意思是产生，形成。

**genspider** 就是生成一个spider（蜘蛛）。

在当前项目中创建spider。这**仅仅是创建spider的一种快捷方法**。该方法可以使用**提前定义好的模板**来生成spider。您也可以自己创建spider的源码文件。

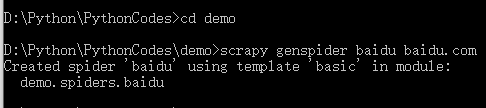


**紧接着，进入工程目录，然后创建一个spider。**



**cd demo**

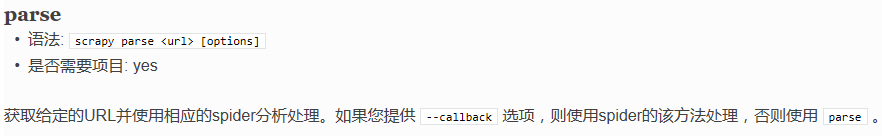
**scrapy genspider spidername domainName**

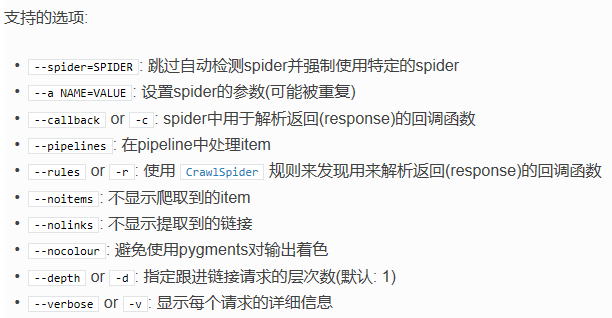


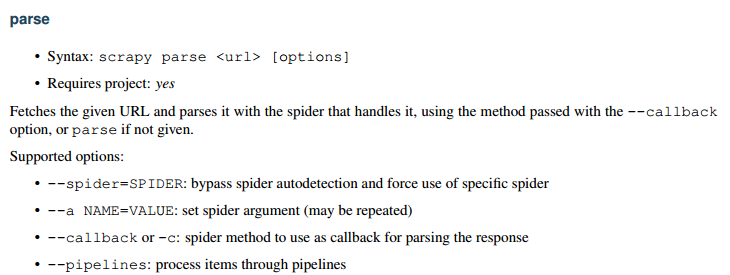
# scrapy parse：用于验证我们编写的parse函数逻辑是否正确，功能是否完全。

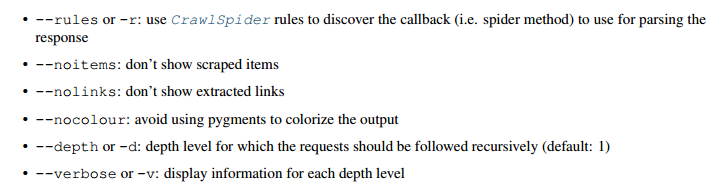
注意：必须在**工程目录**下执行。

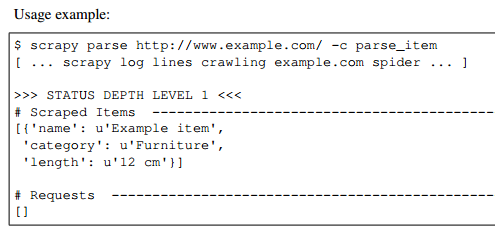




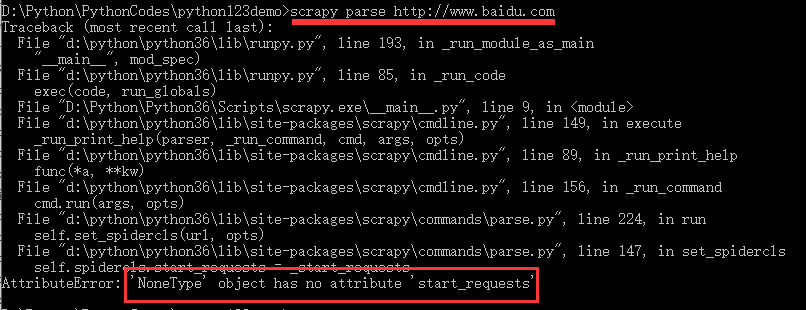




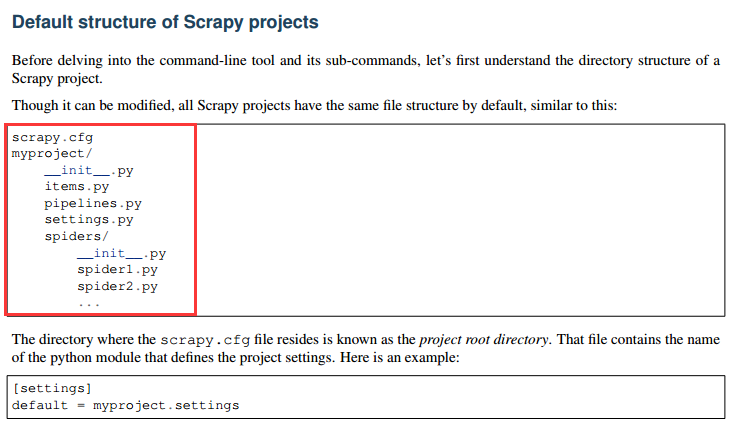




测试出错。



# 默认的工程目录结构



# 自定义项目命令：Custom project commands



