**Scrapy at a glance**[**¶**](https://docs.scrapy.org/en/latest/intro/overview.html#scrapy-at-a-glance)

Scrapy is an application framework for crawling web sites and extracting structured data which can be used for a wide range of useful applications, like data mining, information processing or historical archival.

Parse:

passing the response object as an argument

Parsel : extract and remove data from [HTML](https://en.wikipedia.org/wiki/HTML) and [XML](https://en.wikipedia.org/wiki/XML) using [XPath](https://en.wikipedia.org/wiki/XPath) and [CSS](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) selectors, optionally combined with [regular expressions](https://docs.python.org/library/re.html).

# Virtual Environments

Applications will sometimes need a specific version of a library, because the application may require that a particular bug has been fixed or the application may be written using an obsolete version of the library’s interface.

* [**name**](https://docs.scrapy.org/en/latest/topics/spiders.html#scrapy.Spider.name): identifies the Spider. It must be unique within a project, that is, you can’t set the same name for different Spiders.
* [**start\_requests()**](https://docs.scrapy.org/en/latest/topics/spiders.html#scrapy.Spider.start_requests): must return an iterable of Requests (you can return a list of requests or write a generator function) which the Spider will begin to crawl from. Subsequent requests will be generated successively from these initial requests.
* [**parse()**](https://docs.scrapy.org/en/latest/topics/spiders.html#scrapy.Spider.parse): a method that will be called to handle the response downloaded for each of the requests made. The response parameter is an instance of **[TextResponse](https://docs.scrapy.org/en/latest/topics/request-response.html" \l "scrapy.http.TextResponse" \o "scrapy.http.TextResponse)** that holds the page content and has further helpful methods to handle it

The Yield keyword in Python is **similar to a return statement used for returning values or objects in Python**

# Items[¶](https://docs.scrapy.org/en/latest/topics/items.html#module-scrapy.item)

The main goal in scraping is to extract structured data from unstructured sources, typically, web pages. [Spiders](https://docs.scrapy.org/en/latest/topics/spiders.html#topics-spiders) may return the extracted data as items, Python objects that define key-value pairs. When you create an item, you may use whichever type of item you want. When you write code that receives an item

* --spider=SPIDER: bypass spider autodetection and force use of specific spider
* --a NAME=VALUE: set spider argument (may be repeated)
* --callback or -c: spider method to use as callback for parsing the response
* --meta or -m: additional request meta that will be passed to the callback request. This must be a valid json string. Example: –meta=’{“foo” : “bar”}’
* --cbkwargs: additional keyword arguments that will be passed to the callback. This must be a valid json string. Example: –cbkwargs=’{“foo” : “bar”}’
* --pipelines: process items through pipelines