

See also: [Scrapy homepage](#), [Official documentation](#), [Scrapy snippets on Snipplr](#)

## Getting started

If you're new to Scrapy, start by reading [Scrapy at a glance](#).

## Course (Online)

Check <https://scrapy.org/resources/>.

## Google Summer of Code

- [GSoC 2016](#)
- [GSoC 2015](#)
- [[GSoC 2014]]

## Scrapy help docs and resources

Check <https://scrapy.org/resources/>

- Articles & blog posts
- Books
- Videos
- Slides

## Projects, tools and libraries using Scrapy

- [Django Dynamic Scraper](#) - a web application (written in django) for running and controlling Scrapy spiders
- [Slybot](#) - A supervised learning crawler based on [Scrapely](#).
- [scrapy-sentry](#) - Logs Scrapy exceptions into Sentry
- [ScrapyGraphite](#) - Output scrapy statistics to carbon/graphite
- [scrapy-mongo](#) - A pipeline to store scrapy items in a MongoDB database
- [scrapy-boilerplate](#) - small set of utilities to simplify writing low-complexity spiders
- [scrapy-inline-requests](#) - provides a decorator to write spider callbacks which performs multiple requests without the need to write multiple callbacks for each request
- [scrapy-redis](#) - provides Redis-backed components for Scrapy
- [scrapy](#) - Create simple spiders easily.
- [Scrapy-related libraries on PyPI](#)
- [Scrapy\\_cn](#) - provided a demo to solve encoding problems(utf-8).
- [elite-proxies-scrapy-middleware](#) - get new proxies from your EliteProxies account
- [scrapydo](#) - Crochet-based blocking API for Scrapy.

## Companies using Scrapy

See <http://scrapy.org/companies/>

## Release Notes

- see [Release notes](#) in the official documentation

## Developer documentation

- [[Scrapy Release Procedure]]

- [[Asyncio use cases]]

## **Scrapy Enhancement Proposals**

- SEPs are available in [scrapy/sep](https://scrapy.org/en/latest/topics/sep.html).