Scrapy

# Install Iphython

For easier terminal reading

* Pip install iphython
* Goto scrapy.cfg
* Shell = iphython (under [settings])

# Starting Project

Scrapy startproject <project\_name>

# Spider

* Goto spider folder
* Scrapy genspider <spider\_name> <url of website to scrape>

# Launching Spider

Scrapy crawl <spider\_name>

Should be in the master folder

# Syntax

## Fetch

Fetch(‘<url>’)

Items = Response.css(‘<tag>.<class\_name>’)

Response.css(‘<tag>.<class\_name>’).get() [To get the 1st item]

# To get text

Say book is a container of the 1st item of the list then

Book.css(‘<tag> <tag>::text’).get()

Book.css(‘.class\_name .class\_name::text’).get()

# To get Attribute

Book.css(‘<tag> .class\_name’).attrib[‘<arttrib\_name>’]

Response.css(‘<tag>.class\_name <tag>::attr(href)’).get()

# Going to the next page

Yield response.follow(next\_page\_url, callback = self.parse)

# Xpath

Response.xpath(“//<tag>[@class=’class\_name’]/other\_tags\_classes/preceding-sibling::li[no\_of\_back/forward\_you\_want\_to\_go]/ <tag\_and\_class\_to\_reach\_the\_text>/text()’).get()

Response.xpath(“//<tag>[@class=’class\_name’]/other\_tags\_classes/**following-sibling**::li[no\_of\_back/forward\_you\_want\_to\_go]/ <tag\_and\_class\_to\_reach\_the\_text>/text()’).get()

# Tables

* Table\_rows = response.css(“table tr”)
* Table\_rows[index].css(th/td :: text’).get()

# To get the names of class/id

Response.css(‘Path to the required tag’).attrib[‘class’]

Response.css(‘Path to the required tag’).attrib[‘id’]

# Items.py

Define class for items . for eg

class BookItem(scrapy.Item):

* url = scrapy.Field()
* title= scrapy.Field()
* product\_type = scrapy.Field()
* Total\_price = scrapy.Field()
* tax =scrapy.Field()

**To Import:** from Bookscraper.items import BookItem

**Usage:** book\_item = BookItem()

* book\_item['url'] = response.url
* book\_item['title'] = response.css('.product\_main h1::text').get()
* book\_item['product\_type'] =table\_rows[1].css('td::text').get()
* book\_item['Total\_price'] = table\_rows[3].css('td::text').get()

yield BookItem

## Pipelines.py

Clean your data

from itemadapter import ItemAdapter

class BookscraperPipeline:

    def process\_item(self, item, spider):

        adapter = ItemAdapter(item)

<Data cleaning happens here>

       return item

# Creating new output file

Append Data: Scrapy crawl <spidername>-o <filename>.csv

Create a new file: Scrapy crawl <spidername>-O <filename>.csv

### IN Settings.py

FEED = {

‘<filename.extension>’: {‘format’: ‘<format>’}

}

### IN spider

Custom\_settings = {

‘FEEDS” : {

‘<filename.extension>’: {‘format’: ‘<format>’, ‘overwrite’ : True}

}

}

# Saving into Database