Scrapy : 크롤링 프레임워크 중급

pipelines.py 이해하기

> Scrapy 프로젝트 생성시 제공되는 pipelines.py 파일 역할 이해하기

- pipelines.py 역할: 아이템 데이터 후처리하기
    - 일부 아이템은 저장하지 않거나,
    - 중복되는 아이템을 저장하지 않거나,
    - 데이터베이스등에 저장하거나,
    - 특별한 포멧으로 아이템을 저장하고 싶거나
- pipelines.py 와 spider
    - 간단한 크롤링의 경우, 해당 spider 의 parse 함수에서 pipelines.py 역할을 처리해도 됨
        - 예 내가 원하는 데이터만 yield 를 호출하면 됨
    - 다만, 복잡하고 방대한 크롤링의 경우, 별도 파일에 작성할 수 있도록 하였음

테스트 프로젝트 사전 작업

- 기존 문제1/문제2 로 작성된 프로젝트를 기반으로 코드를 업데이트하며, pipelines.py 파일 역할 및 사용법 이해

pipelines.py 사용 설정

> pipelines.py 사용을 위해서는 다음 설정 필요

- settings.py 수정
  - 다음 코드를 찾아서, 주석을 풀어줌

```
ITEM_PIPELINES = {
   'mycrawler.pipelines.MycrawlerPipeline': 300,
}
```

- 일종의 우선순위 번호로, 0 ~ 1000 숫자중 임의로 숫자를 부여하면 됨
  - 여러 클래스가 있을 경우, 숫자가 낮을 수록 먼저 실행됨
- pipelines 설정 시, 스파이더 실행 후, 다음 예와 같이 crawling 실행시, Enabled item pipelines를 터미널에서 확인할 수 있음

```
2021-11-15 17:37:54 [scrapy.middleware] INFO: Enabled item pipelines:
['mycrawler.pipelines.MycrawlerPipeline']
```

pipelines.py 테스트로 이해하기

- 각 아이템 생성 시(yield 시), pipeline.py 에 있는 process_item 함수를 호출하게 되어 있음
- 필요한 아이템만 return 해주고, 필터링할 아이템은 DropItem 을 통해, 해당 item 은 처리하지 않음
  - 즉, 크롤링 후, 후처리를 통해, 원하는 item 만 저장할 수 있음

```
from scrapy.exceptions import DropItem


class MycrawlerPipeline(object):
    def process_item(self, item, spider):
        if item['product_type'] == '행거도어 관련 상품 추천':
            raise DropItem('drop item for hanger door')
        else:
            return item
```

- 다음과 같이 실행 후, '행거도어 관련 상품 추천' 상품은 저장되지 않음을 확인 (기존 파일 삭제 후 실행)

```
scrapy crawl test_web -o test_web.json -t json
```