Scrapy : 크롤링 프레임워크 중급

여러 페이지 한번에 크롤링하는 spider 만들기

- 사전 작업
  - 기존 프로젝트 폴더에서, 다음과 같이 실행 및 확인

```
scrapy genspider multiple_webs davelee-fun.github.io
```

- 유사한 사이트에 대한 크롤링 요청시, Scrapy 는 해당 작업을 막는 것을 디폴트로 함
  - settings.py 에 다음 설정 추가로, 유사한 사이트의 크롤링 지원 가능
  - 무한 반복 크롤링등이 있을 수 있으므로, 주의깊게 코드 작성 필요

```
# Enable duplicated site crawling
DUPEFILTER_CLASS = 'scrapy.dupefilters.BaseDupeFilter'
```

여러 페이지 한번에 크롤링하는 spider 만들기

- 크롤링 사이트를 설정하는 방법

  (1) start_urls 에 크롤링할 사아트를 리스트 아이템으로 작성하기

  (2) start_requests 함수를 별도로 선언해서, 크롤링 사이트를 정의하기

```
def start_requests(self):
    yield scrapy.Request(크롤링사이트, 각 크롤링을 처리할 함수)
# 크롤링을 처리할 함수가 모든 페이지에 동일할 경우에는 self.parse 로 기재
```

- start_requests 함수를 사용하여 여러 페이지 크롤링 코드 작성 예

```
class MultipleWebsSpider(scrapy.Spider):
    name = 'multiple_webs'
    allowed_domains = ['davelee-fun.github.io']
    start_urls = ['http://davelee-fun.github.io/']

    def start_requests(self):
        yield scrapy.Request('http://davelee-fun.github.io/', self.parse)
        for i in range(2, 7):
            yield scrapy.Request('https://davelee-fun.github.io/page' + str(i), self.parse)

    def parse(self, response):
        print (response.url)
        pass
```

여러 페이지 한번에 크롤링하는 spider 만들기

- 테스트 (웹사이트 주소가 출력되는지 확인하기)

```
scrapy crawl multiple_webs
```

여러 페이지 한번에 크롤링하는 spider 만들기

- 사전 작업
  - 기존 프로젝트 폴더에서, 다음과 같이 실행 및 확인

```
scrapy genspider multiple_webs_item davelee-fun.github.io
```

- 기존 test_web.py 의 parse 함수와 multiple_webs.py 코드에서 start_requests 함수 복사
- 다음 쉘명령으로 여러 페이지 크롤링 결과 저장

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