크롤러 스크래퍼를 만들어보자 with Python

@wapj 박승규

이상한모임

오늘의 요리

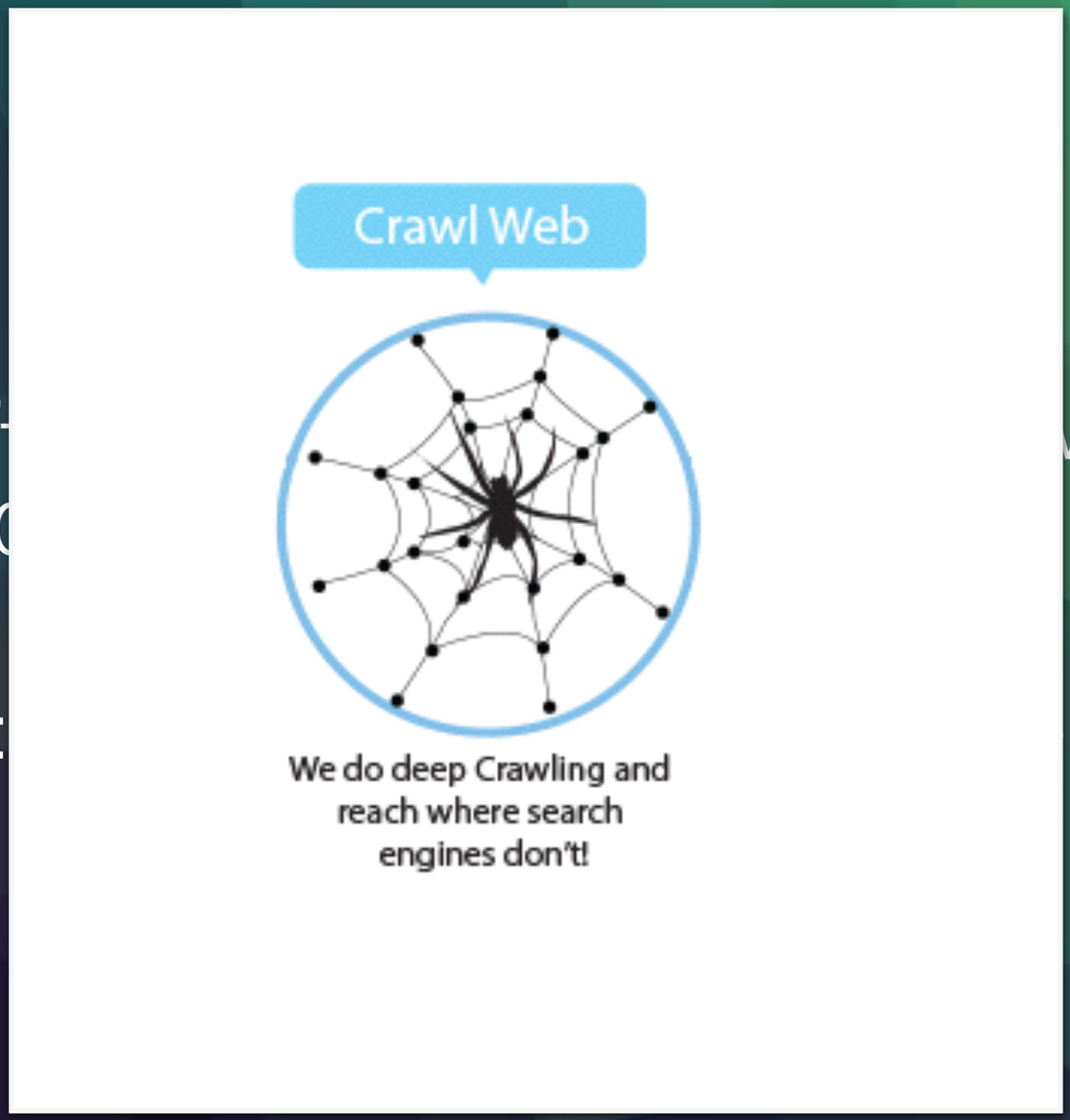
- 수동 스크래핑
- 스크래핑 1단계 : 1개 사이트 스크래핑
- 스크래핑 2단계 : 여러 사이트 스크래핑
- 스크래핑 3단계 : 2단계 + 스케줄러
- 크롤러도 만들어 볼까?

크롤링? 스크래핑?

- 크롤링 : 거미가 거미줄을 기어다니듯(crawl), link들을 다니면서 데이터를 수집하는 것

- 스크래핑 : 웹페이지에서 필요한 데이터를 갈무리(scrap)하는 것

크롤링? 스크래핑?

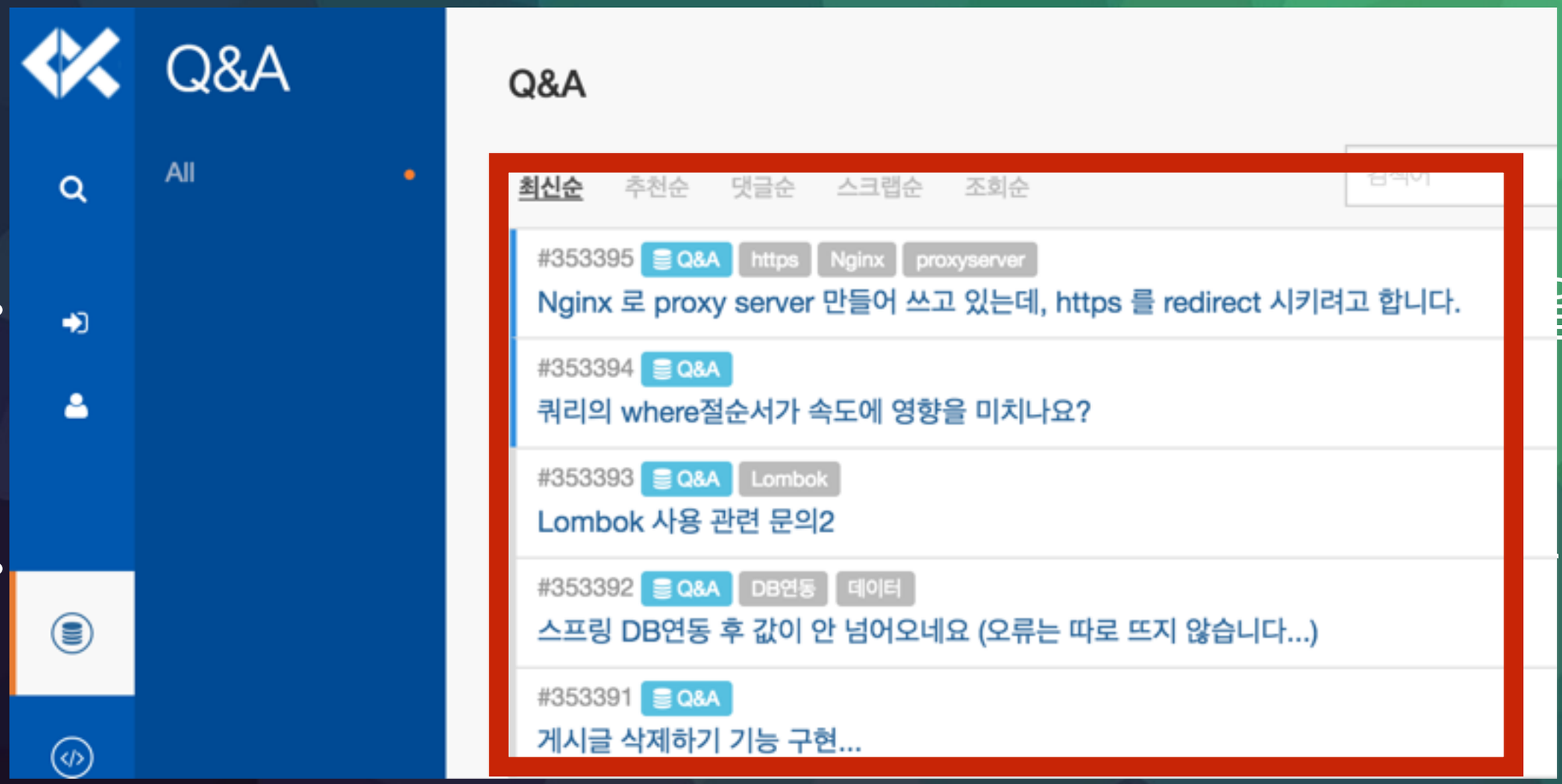


- 크롤링 : 거 wl), link들을 다 니면서 데(

- 스크래핑 : 갈무리(scrap) 하는 것

이미지 출처 https://www.promptcloud.com/web-crawler-tool-online-tools-for-crawling/

크롤링? 스크래핑?



- 을 다
- ap)

재료부터 준비해봅시다

재료준비

- python3.x
- requests : http client 라이브러리
- beautifulsoup4 : xml이나 html 파싱을 위한 라이브러리
- apscheduler : 스케줄링 작업을 위한 라이브러리
- scrapy : 크롤러 프레임워크

수동 스크래핑

수동 스크래핑

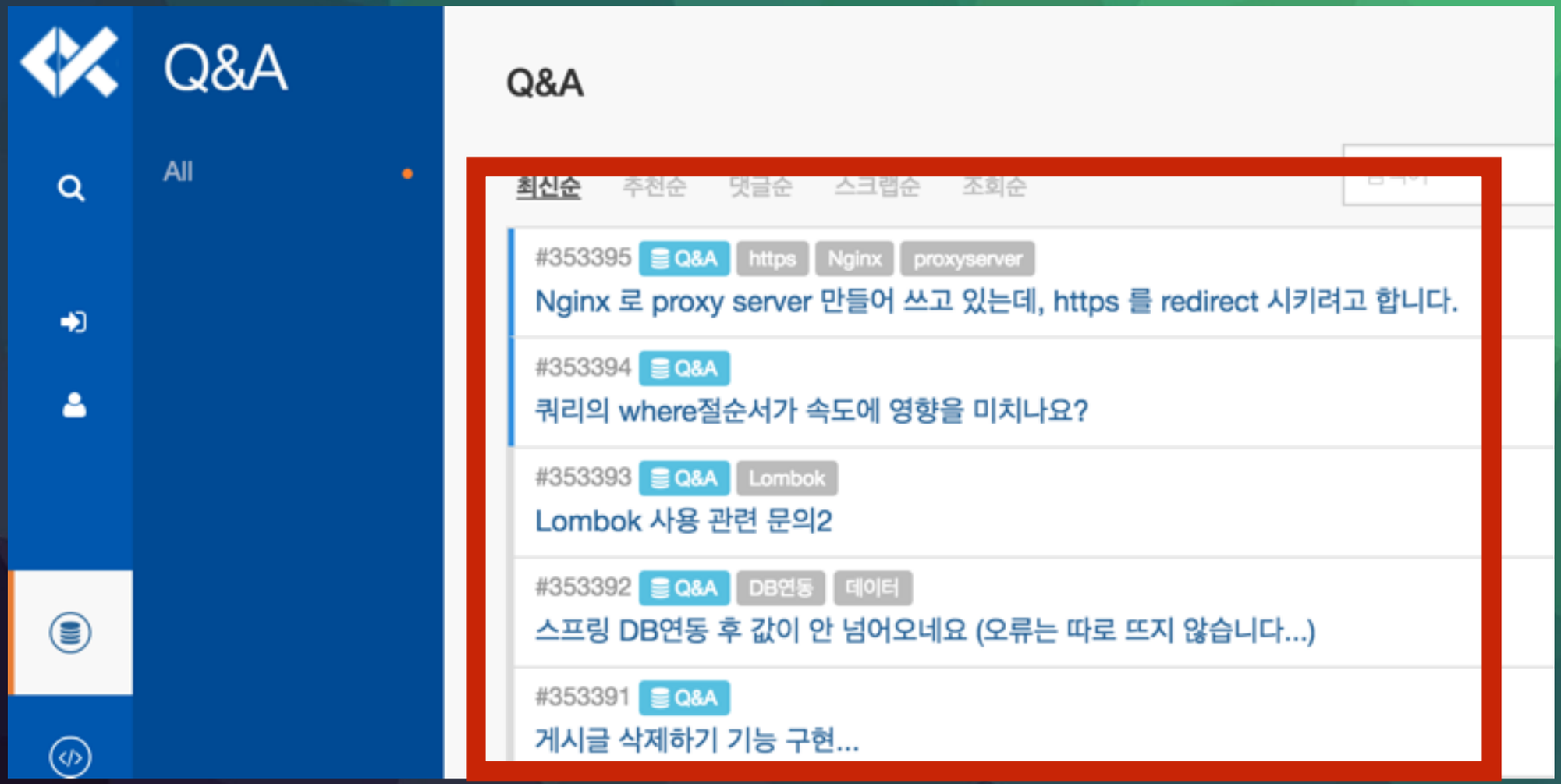
- 사이트에 접속한다
- html을 까본다
- 필요한 내용을 긁는다(scraping)
- 저장한다

컴퓨터에게 시켜보자

스크래핑 1단계

- 사이트에 접속한다
- html을 까본다 : requests
- 필요한 내용을 긁는다(scraping) : beautifulsoup4
- 저장한다 : 일단 파일로

사이트에 접속한다


Q&A
All
Q&A
최신순
추천순
댓글순
스크랩순
조회순
#353395 Q&A https Nginx proxyserver
Nginx 로 proxy server 만들어 쓰고 있는데, https 를 redirect 시키려고 합니다.
#353394 Q&A
쿼리의 where절순서가 속도에 영향을 미치나요?
#353393 Q&A Lombok
Lombok 사용 관련 문의2
#353392 Q&A DB연동 데이터
스프링 DB연동 후 값이 안 넘어오네요 (오류는 따로 뜨지 않습니다...)
#353391 Q&A
게시글 삭제하기 기능 구현...

html을 까본다

li.list-group-item

```
<li class="list-group-item list-group-item-question list-group-has-note clearfix">

    <div class="list-title-wrapper clearfix">
        <div class="list-tag clearfix">
            <span class="list-group-item-text article-id">#353395</span>
            <a href="/articles/questions" class="list-group-item-text item-tag lab l
label-info"><i clas "fa fa-database"></i> Q&A</a>
            <a href="/articles/tagged/https" class="list-group-item-text item-tag
label label-gray">ht ps</a> <a href="/articles/tagged/Nginx" class="list-group-item-text item-tag label
label-gray">Nginx</ <a href="/articles/tagged/proxyserver" class="list-group-item-text item-tag label
ver</a>
        </div>

        <h5 class="list-group-item-heading list-group-item-evaluate">
            <a href="/article/353395">

                Nginx 로 proxy server 만들어 쓰고 있는데, https 를 redirect 시키려고 합니다.
            </a>
        </h5>
    </div>

    <div class="list-summary-wrapper clearfix">

        <div class="item-evaluate-wrapper pull-right clearfix">
            <div class="item-evaluate">
                <div class="item-evaluate-icon">
                    <i class="item-icon fa fa-thumbs-o-up"></i>
                </div>
                <div class="item-evaluate-count">
                    <span>0
                </div>
            </div>
            <div class="item-evaluate item-evaluate-has-note">
                <div class="item-evaluate-icon">

                    <i class="item-icon fa fa-exclamation-circle"></i>

                </div>
                <div class="item-evaluate-count">
                    1
                </div>
            </div>
        </div>
```

필요한 내용을 긁는다

```
<li class="list-group-item list-group-item-question list-group-has-note clearfix">

    <div class="
        <div cl
            <sp                                    id">#353395</span>
            <a                                     t-group-item-text item-tag label
label-info"><i class="fa fa-database"></i>
            <a href="/articles/tagged/https" class="list-group-item-text item-tag
label label-gray">https</a> <a href="/articles/tagged/Nginx" class="list-group-item-text item-tag label
label-gray">Nginx</a> <a href="/articles/tagged/proxyserver" class="list-group-item-text item-tag label
label-gray">proxyserver</a>
        </div>

        <h5 class="list-group-item-heading list-group-item-evaluate">
            <a href="/article/353395">
                Nginx 로 proxy server 만들어 쓰고 있는데, https 를 redirect 시키려고 합니다.
            </a>
        </h5>
    </div>

    <div c                                         ll-right clearfix">
            <div class="item-evaluate-icon">
                <i class="item-icon fa fa-thumbs-o-up"></i>
            </div>
            <div class="item-evaluate-count">
                <span>0
            </div>
        </div>
        <div class="item-evaluate item-evaluate-has-note">
            <div class="item-evaluate-icon">

                <i class="item-icon fa fa-exclamation-circle"></i>

            </div>
            <div class="item-evaluate-count">
                1
            </div>
        </div>
    </div>

</div>
```

li.find('span').text[1:]

li.find('h5').find('a')['href']

li.find('h5').find('a').text

코드로 볼까요?

```
import requests
from bs4 import BeautifulSoup

def parse_url(url):
    #html을 요청
    response = requests.get(url)

    soup = BeautifulSoup(response.text, 'html.parser')

    #li 들의 리스트를 가져옴
    list = soup.select('li.list-group-item')

    # 필요한 내용을 파싱
    for li in list:
        a = li.find('h5').find('a').text
        id = li.find('span').text[1:] # 게시글 번호
        link = a['href']              # 링크
        title = a.text.strip()        # 타이틀
        print(id, link, title)

url= 'http://okky.kr/articles/questions'
parse_url(url)
```

파일로 저장

```
import requests
from bs4 import BeautifulSoup

result = ''

def parse_url(url):
    #html을 요청
    response = requests.get(url)
    #…중략

    # result
    global result
    # 필요한 내용을 파싱
    for li in list:
        a = li.find('h5').find('a').text
        id = li.find('span').text[1:] # 게시글 번호
        link = a['href']              # 링크
        title = a.text.strip()        # 타이틀
        print(id, link, title)
        result +=  'id : %s, link : %s, title : %s\n' % (id, link, title)

f = open("okky.kr-scrap.txt", 'w', encoding='utf-8')
f.write(result)
f.close()
```

스크래핑 2단계

- 사이트들의 URL을 준비한다
- 사이트별로 필요한 파싱 메서드를 만든다
- html 요청시 파싱 메서드를 동적으로 바꾼다(스트래티지 패턴)
- 필요한 내용을 긁어온다
- 저장한다

스크래핑 2단계
사이트들의 URL을 준비한다

```
### 총 3개의 사이트의 7개의 URL을 준비했습니다
http://okky.kr/articles/questions
http://okky.kr/articles/tech
http://okky.kr/articles/community
http://okky.kr/articles/columns
http://www.todayhumor.co.kr/board/list.php?table=programmer
http://www.todayhumor.co.kr/board/list.php?table=it
https://qna.iamprogrammer.io/
```

스크래핑 2단계
사이트별로 파싱메서드를 만든다

```
def todayhumor_parser(_url):
    response = requests.get(_url)

    soup = BeautifulSoup(response.text, 'html.parser')
    lists = soup.select('tr.view')

    for li in lists:
        a = li.select_one('td.no a')
        list_id = a.text
        link = a['href']
        title = li.select_one('td.subject a').text
        print(list_id, link, title)
```

스크래핑 2단계
사이트별로 파싱메서드를 동적으로 바꾼다

```
def iamprogrammer_parse(_url):
    print('아직 안만들었어~ 일해라~~')
    pass

parser_select_dict = {
    'okky.kr': okky_parser,
    'www.todayhumor.co.kr': todayhumor_parser,
    'qna.iamprogrammer.io': iamprogrammer_parse,
    'www.ppomppu.co.kr': ppomppu_parser
}
```

```
for url in urls:
    parsed_url = urlparse(url)

    # 사이트별로 사용할 파서를 선택해서 넣어줌
    print("==========================================================")
    print("[selected parser] %s" % parsed_url[1])
    print("==========================================================")

    try:
        func = parser_select_dict[parsed_url[1]]
        parser = PokoParser(func)
        parser.parse_url(url)
    except KeyError as e:
        print('이 사이트의 파서를 만들어야 해요~ 개발자님 ', e)
```

스크래핑 3단계

실행시키는 것도
컴퓨터에게 시켜보자

스크래핑 3단계

APScheduler를 사용해
1분에 한번씩 스크랩하자

스크래핑 3단계

- APScheduler의 BlockingScheduler() 를 사용
- 기존 스크래핑 코드를 함수로 묶어줌
- APScheduler의 인터벌을 사용하여 1분에 한번씩 스크래핑하는 코드가 실행되도록 한다.

스크래핑 3단계
기존의 스크래핑하는 코드를 함수로 묶어줌

```
def scraping():
    for url in urls:
        parsed_url = urlparse(url)
        # 사이트별로 사용할 파서를 선택해서 넣어줌

print("==========================================================")
        print("[selected parser] %s" % parsed_url[1])

print("==========================================================")
        try:
            func = parser_select_dict[parsed_url[1]]
            parser = PokoParser(func)
            parser.parse_url(url)
        except KeyError as e:
            print('이 사이트의 파서를 만들어야 해요~ 개발자님 ', e)
```

스크래핑 3단계
APScheduler로 1분에 한번씩
scraping()메서드 실행

```
import os
from apscheduler.schedulers.blocking import BlockingScheduler

if __name__ == '__main__':
    scheduler = BlockingScheduler()
    print("START!")
    scheduler.add_job(scraping, 'interval', seconds=30)
    print('Press Ctrl+{0} to exit'.format('Break' if os.name == 'nt' else 'C'))

    try:
        scheduler.start()
    except (KeyboardInterrupt, SystemExit):
        pass
```

크롤러도 만들어볼까요?

크롤러도 만들어볼까요?

크롤러는 다음에…ㅠㅠ

Scrapy가 좋다고 합니다..