

# Ask Django

여러분의 파이썬/장고 페이스메이커가 되겠습니다.  
**EP 8. 크롤링 실습 - 네이버 웹툰 목록 크롤링 및 이미지 합치기**

# 네이버 웹툰 목록 크롤링

```
from collections import OrderedDict
from itertools import count
from urllib.parse import urljoin
import requests
from bs4 import BeautifulSoup

def get_list(title_id):
    list_url = 'http://comic.naver.com/webtoon/list.nhn'
    ep_dict = OrderedDict()

    for page in count(1):
        params = {'titleId': title_id, 'page': page}
        print('try {}'.format(params))

        list_html = requests.get(list_url, params=params).text
        soup = BeautifulSoup(list_html, 'html.parser')

        for tag in soup.select('.viewList tr td.title'):
            tag_a = tag.find('a')
            is_up = bool(tag.find('img'))
            link = urljoin(list_url, tag_a['href'])
            title = tag_a.text
            print(title, is_up, link, img_url)

            if link in ep_dict:
                return ep_dict

        ep = {
            'title': title,
            'is_up': is_up,
            'link': link,
            'img_url': img_url,
        }
        ep_dict[link] = ep
```

```
>>> get_list(650305)
```

```
try {'titleId': 650305, 'page': 1}
```

```
2부30화 함정1 True http://comic.naver.com/webtoon/detail.nhn?titleId=650305&no=113&weekday=sat http://static.naver.net/comic/images/2012/ico_toonup.png
```

```
2부29화 먼 길을 나서다 False http://comic.naver.com/webtoon/detail.nhn?titleId=650305&no=112&weekday=sat http://static.naver.net/comic/images/2012/ico_toonup.png
```

```
2부28화 도모지의 계략2 False http://comic.naver.com/webtoon/detail.nhn?titleId=650305&no=111&weekday=sat http://static.naver.net/comic/images/2012/ico_toonup.png
```

```
2부27화 도모지의 계략1 False http://comic.naver.com/webtoon/detail.nhn?titleId=650305&no=110&weekday=sat http://static.naver.net/comic/images/2012/ico_toonup.png
```

```
2부26화 응징2 False http://comic.naver.com/webtoon/detail.nhn?titleId=650305&no=109&weekday=sat http://static.naver.net/comic/images/2012/ico_toonup.png
```

```
2부25화 응징1 False http://comic.naver.com/webtoon/detail.nhn?titleId=650305&no=108&weekday=sat http://static.naver.net/comic/images/2012/ico_toonup.png
```

```
2부24화 비녀단의 습격5 False http://comic.naver.com/webtoon/detail.nhn?titleId=650305&no=107&weekday=sat http://static.naver.net/comic/images/2012/ico_toonup.png
```

```
2부23화 비녀단의 습격4 False http://comic.naver.com/webtoon/detail.nhn?titleId=650305&no=106&weekday=sat http://static.naver.net/comic/images/2012/ico_toonup.png
```

```
# 종락
```

```
4화 추이와 황요 False http://comic.naver.com/webtoon/detail.nhn?titleId=650305&no=4&weekday=sat http://static.naver.net/comic/images/2012/ico_toonup.png
```

```
try {'titleId': 650305, 'page': 12}
```

```
3화 무커의 분노 False http://comic.naver.com/webtoon/detail.nhn?titleId=650305&no=3&weekday=sat http://static.naver.net/comic/images/2012/ico_toonup.png
```

```
2화 창귀호(虎)의 습격 False http://comic.naver.com/webtoon/detail.nhn?titleId=650305&no=2&weekday=sat http://static.naver.net/comic/images/2012/ico_toonup.png
```

```
1화 산군(山君). 산의 왕 False http://comic.naver.com/webtoon/detail.nhn?titleId=650305&no=1&weekday=sat http://static.naver.net/comic/images/2012/ico_toonup.png
```

```
try {'titleId': 650305, 'page': 13}
```

```
3화 무커의 분노 False http://comic.naver.com/webtoon/detail.nhn?titleId=650305&no=3&weekday=sat http://static.naver.net/comic/images/2012/ico_toonup.png
```

# 특정 에피소드의 이미지 다운받기

```
def ep_download(ep_url):
    html = requests.get(ep_url).text
    soup = BeautifulSoup(html, 'html.parser')

    comic_title = soup.select('.comicinfo h2')[0].text
    ep_title = soup.select('.tit_area h3')[0].text

    comic_title = ' '.join(comic_title.split())
    ep_title = ' '.join(ep_title.split())

    img_path_list = []

    for tag in soup.select('.wt_viewer img'):
        img_url = tag['src']
        headers = {'Referer': ep_url, 'User-Agent': 'Mozilla/5.0 (Windows; U; MSIE 9.0; Windows NT 9.0; en-US);'}

        img_name = os.path.basename(img_url)
        img_path = os.path.join(comic_title, ep_title, img_name)

        dir_path = os.path.dirname(img_path)
        if not os.path.exists(dir_path):
            os.makedirs(dir_path)

        print(img_url, end=' : ')
```

```
# print(img_url, end=' : ') 코드에 이어

img_path_list.append(img_path)

if os.path.exists(img_path):
    #     img_headers = requests.head(img_url, headers=headers).headers
    #     if int(img_headers['Content-Length']) == os.stat(img_path).st_size:
        print('skip')
        continue

img_data = requests.get(img_url, headers=headers).content

with open(img_path, 'wb') as f:
    f.write(img_data)

print('download')

break

print('ENDED')

return img_path_list

>>> ep_download('http://comic.naver.com/webtoon/detail.nhn?titleId=650305&no=103&weekday=sat')
```

# 이미지를 세로로 합치기 (주요코드)

```
from PIL import Image

im_list = []
for img_path in img_path_list:
    im = Image.open(img_path)
    im_list.append(im)

canvas_size = (
    max(im.width for im in im_list),
    min(65500, sum(im.height for im in im_list))
)

print(canvas_size)

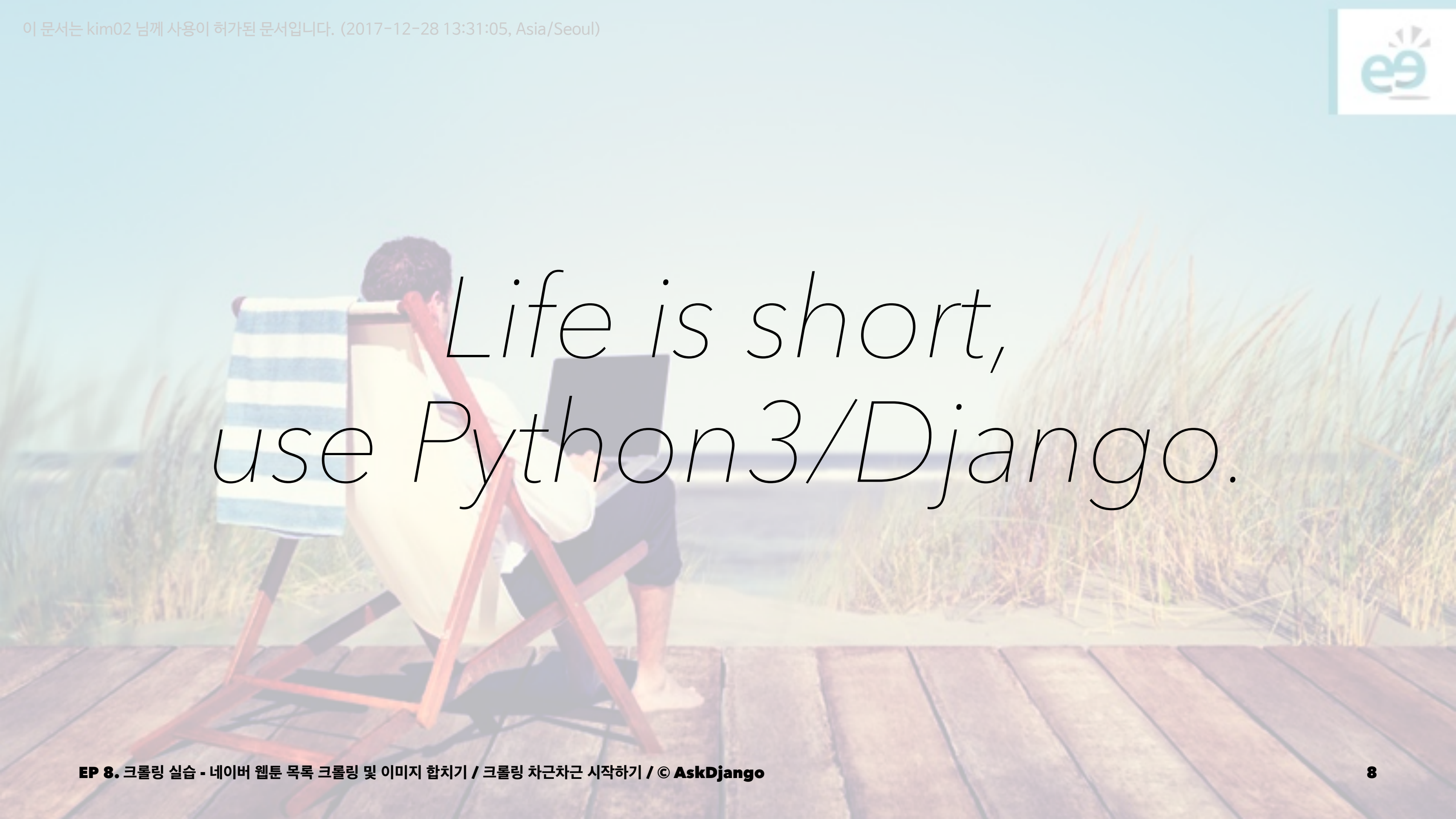
canvas = Image.new('RGB', canvas_size)

top = 0
for im in im_list:
    canvas.paste(im, (0, top))
    top += im.height

canvas.save('canvas.jpg')
```

## Tip : 포맷별 최대 지원 크기

- jpg는 최대  $2^{16}-1$  (65,535) 픽셀
- png는 최대  $2^{31}-1$  (2,147,483,647) 픽셀 (signed) #spec



*Life is short,  
use Python3/Django.*