이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

Ask Django

CSV/엑셀 파일 만들기

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

CSV [1]

- 각 줄은 개행문자로 구분하며, 각 컬럼은 콤마(,)로 구분
- 스프레드시트/데이터베이스 소프트웨어에서 지원
- TSV(Tab-Separated Values) 포맷도 CSV라 퉁쳐서 부름.
- Plain Text 파일이므로 다양한 인코딩으로 작성이 될 수 있습니다.
  - 한글 엑셀에서는 CSV파일을 생성하면 CP949 인코딩으로 생성. 그렇지만 UTF8 인코딩의 CSV파일을 읽어들일 순 있습니다.
  - 대개의 소프트웨어에서는 UTF8 인코딩으로 작성. 파이썬의 CSV모듈에서도 UTF8이 디폴트

---

[1] comma-separated values wikipedia

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

왜 CSV파일을 쓰는가?

- 표 형식의 데이터를 Plain Text로 쉽게 생성이 가능
  - 파일크기가 작다.
- 거의 대부분의 프로그래밍 언어나, 데이터분석 프로그램에서 CSV파일을 지원 (Pandas, 엑셀, Github 등)
- 서식 지정없이 데이터만 생성코자 할 때 유용
- 파이썬 내장 라이브러리만으로 읽기/쓰기가 가능

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

CSV SAMPLE / webtoon.xls

월요웹툰	화요웹툰	수요웹툰	목요웹툰	금요웹툰	토요웹툰	일요웹툰
신의 탑	마음의소리	고수	기기괴괴	덴마	호랑이형님	열럽전사
귀전구담	노블레스	퍼스트미션	하루 3컷	테러맨	부활남	다이스
히어로메이커	하이브	DEY 호러채널	마술사	오즈랜드	유미의세포들	조선왕조실톡

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

CSV SAMPLE / webtoon.csv

월요웹툰, 화요웹툰, 수요웹툰, 목요웹툰, 금요웹툰, 토요웹툰, 일요웹툰
신의 탑, 마음의소리, 고수, 기기괴괴, 덴마, 호랑이형님, 열럽전사
귀전구담, 노블레스, 퍼미스미션, 하루 3컷, 테러맨, 부활남, 다이스
히어로메이커, 하이브, DEY 호러채널, 마술사, 오즈랜드, 유미의세포들, 조선왕조실톡

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

CSV 파일 만들기

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

이렇게?

```
lines = '''월요웹툰,화요웹툰,수요웹툰,목요웹툰,금요웹툰,토요웹툰,일요웹툰
신의 탑,마음의소리,고수,기기괴괴,덴마,호랑이형님,열럽전사
귀전구담,노블레스,퍼미스미션,하루 3컷,테러맨,부활남,다이스
히어로메이커,하이브,DEY 호러채널,마술사,오즈랜드,유미의세포들,조선왕조실톡'''

with open('webtoon.csv', 'wt', encoding='utf8') as f:
    f.write(lines)
```

NO !!! 문자열 조합을 직접 해줘야 ;;;

가독성도 떨어지고, 데이터가 복잡/많아지면 번거롭습니다.

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

이렇게?

```
rows = [
    ['월요웹툰', '화요웹툰', '수요웹툰', '목요웹툰', '금요웹툰', '토요웹툰', '일요웹툰'],
    ['신의 탑', '마음의소리', '고수', '기기괴괴', '덴마', '호랑이형님', '열럽전사'],
    ['귀전구담', '노블레스', '퍼미스미션', '하루 3컷', '테러맨', '부활남', '다이스'],
    ['히어로메이커', '하이브', 'DEY 호러채널', '마술사', '오즈랜드', '유미의세포들', '조선왕조실톡'],
]

lines = '\r\n'.join(','.join(row) for row in rows)

with open('webtoon.csv', 'wt', encoding='utf8') as f:
    f.write(lines)
```

NO !!! 문자열 조합을 직접 해줘야 ;;;

가독성도 떨어지고, 데이터가 복잡/많아지면 번거롭습니다.

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

csv 라이브러리를 쓰세요.

```
import csv

rows = [
    ['월요웹툰', '화요웹툰', '수요웹툰', '목요웹툰', '금요웹툰', '토요웹툰', '일요웹툰'],
    ['신의 탑', '마음의소리', '고수', '기기괴괴', '덴마', '호랑이형님', '열럽전사'],
    ['귀전구담', '노블레스', '퍼미스미션', '하루 3컷', '테러맨', '부활남', '다이스'],
    ['히어로메이커', '하이브', 'DEY 호러채널', '마술사', '오즈랜드', '유미의세포들', '조선왕조실톡'],
]

with open('webtoon.csv', 'wt', encoding='utf8') as f:
    writer = csv.writer(f)
    writer.writerows(rows)
```

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

구분자로 변경할 수 있어요.

```
import csv

rows = [
    ['월요웹툰', '화요웹툰', '수요웹툰', '목요웹툰', '금요웹툰', '토요웹툰', '일요웹툰'],
    ['신의 탑', '마음의소리', '고수', '기기괴괴', '덴마', '호랑이형님', '열럽전사'],
    ['귀전구담', '노블레스', '퍼미스미션', '하루 3컷', '테러맨', '부활남', '다이스'],
    ['히어로메이커', '하이브', 'DEY 호러채널', '마술사', '오즈랜드', '유미의세포들', '조선왕조실톡'],
]

with open('webtoon.csv', 'wt', encoding='utf8') as f:
    writer = csv.writer(f, delimiter='|')
    writer.writerows(rows)
```

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

Tip: open 함수에 encoding을 지정해준 이유는?

encoding옵션을 지정하지않으면,
locale.getpreferredencoding(False) 값을 활용

```
>>> import locale
>>> locale.getpreferredencoding(False)
'UTF-8'
```

시스템에 따라 변경되지 않고, *UTF8* 인코딩으로 생성할 것임을 명시

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

CSV writer #doc

각 Row의 데이터가 list/tuple일 때

```
import csv

writer = csv.writer(파일객체)

# 1 Row를 쓸 때
writer.writerow(['컬럼1', '컬럼2', '컬럼3'])

# 다수 Row를 쓸 때
writer.writerows([
    ['1행1열', '1행2열','1행3열'],
    ['2행1열', '2행2열','2행3열', '2행4열'],
])
```

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

CSV writer #doc

각 Row의 데이터가 dict일 때

```
import csv

fieldnames = ['first_name', 'last_name']
writer = csv.DictWriter(파일객체, fieldnames=fieldnames)

writer.writeheader()

# 1 Row를 쓸 때
writer.writerow({'first_name': 'Baked', 'last_name': 'Beans'})

# 다수 Row를 쓸 때
writer.writerows([
    {'first_name': 'Lovely', 'last_name': 'Spam'},
    {'first_name': 'Wonderful', 'last_name': 'Spam'},
])
```

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

CSV 파일 읽기

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

이렇게?

```
with open('webtoon.csv', 'rt', encoding='utf8') as f:
    for line in f:
        row = line.split(',')
        print(row)  # list type
```

No !!!

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

csv 라이브러리를 쓰세요.

```
import csv

with open('webtoon.csv', 'rt', encoding='utf8') as f:
    reader = csv.reader(f)
    for row in reader:
        print(row)  # list type
```

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

dict으로 받을려면

```
import csv

with open('webtoon.csv', 'rt', encoding='utf8') as f:
    fieldnames = None  # 디폴트. None이면 첫번째 Row가 fieldnames으로 지정
    reader = csv.DictReader(f, fieldnames=fieldnames)
    for row in reader:
        print(row)  # dict type
```

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

Tip: 생성한 CSV파일을 윈도우 엑셀에서 열고자 할 때

인코딩 이슈가 발생할 수 있습니다. 한글이 깨진 것처럼 보여요.

상황 : 데이터 분석팀에서는 엑셀로 데이터분석을 합니다.

- CSV 파일은 Plain Text파일로서 인코딩을 명시할 수 없습니다.
- 한글 윈도우의 기본 인코딩은 CP949이며, 엑셀도 그러합니다.
    - 하지만 윈도우에서는 BOM[2]를 통해 인코딩을 명시할 수 있습니다.

---

[2] Byte Order Mark - wikipedia

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

CSV 인코딩별 처리

구분	CP949	UTF8 (추천)	UTF8 BOM [3]
엑셀	OK	옵션 지정	OK
G 스프레드시트	OK	OK	OK
Pandas	옵션 지정	OK	OK
Python	옵션 지정	OK	BOM 제거필요

[3] 0xEF 0xBB 0xBF

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

코드

```
# Pandas에서 CP949인코딩의 CSV읽기
df = pandas.read_csv('cp949.csv', encoding='cp949')  # DataFrame

# Python에서 CP949인코딩의 CSV읽기
with open('cp949.csv', encoding='cp949') as f:
    reader = csv.reader(f)

# Python에서 UTF8 BOM 읽기
import csv
import codecs

with open('웹툰_utf8_with_bom.csv', 'rb') as f:
    content = f.read()
    if content.startswith(codecs.BOM_UTF8):
        bom_size = len(codecs.BOM_UTF8)
        content = content[bom_size:]
    string = content.decode('utf8')

    reader = csv.reader(string.splitlines())
    for line in reader:
        print(line)
```

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

엑셀 파일 만들기

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

다양한 엑셀 지원 라이브러리

- **tablib** #doc : Python Module for Tabular Datasets in XLS, CSV, JSON, YAML, &c.
  - csv, json, yaml, xls(x) 포맷에 대한 import/export 지원
- openpyxl #doc : A Python library to read/write Excel 2010 xlsx/xlsm files
- pyexcel #github : Python Wrapper that provides one API for reading, manipulating and writing data in csv, ods, xls, xlsx and xlsm files
- xlwt #github, xlrd #github

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

tablib

```
rows = [
    ['월요웹툰', '화요웹툰', '수요웹툰', '목요웹툰', '금요웹툰', '토요웹툰', '일요웹툰'],
    ['신의 탑', '마음의소리', '고수', '기기괴괴', '덴마', '호랑이형님', '열럽전사'],
    ['귀전구담', '노블레스', '퍼미스미션', '하루 3컷', '테러맨', '부활남', '다이스'],
    ['히어로메이커', '하이브', 'DEY 호러채널', '마술사', '오즈랜드', '유미의세포들', '조선왕조실톡'],
]

import tablib
data = tablib.Dataset()
data.headers = rows[0]
for row in rows[1:]:
    data.append(row)

data.json  # 문자열 반환
data.yaml  # 문자열 반환
data.xlsx  # xlsx 바이너리 반환

with open('webtoon.xlsx', 'wb') as f:
    f.write(data.xlsx)
```

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)

연습문제

<네이버 웹툰의 요일별 전체웹툰> 목록을 CSV파일로 생성해보세요.

```
import requests
from bs4 import BeautifulSoup

html = requests.get('http://comic.naver.com/webtoon/weekday.nhn').text
soup = BeautifulSoup(html, 'html.parser')

# 나머지를 구현해보세요.
# 크롤링부분은 크롤링 VOD를 참고하세요.
```

이 문서는 user124 님께 사용이 허가된 문서입니다. (2020-01-01 21:50:24, Asia/Seoul)