

0822

≡ 텍스트

pymysql로 python-mysql 연결, streamlit, app.py 실습

python과 mysql 연결

mysql 디렉토리를 만들어서 jupyter notebook을 실행한다.

```
!pip install pymysql
```

pymysql 모듈을 설치한다.

데이터 베이스 연결

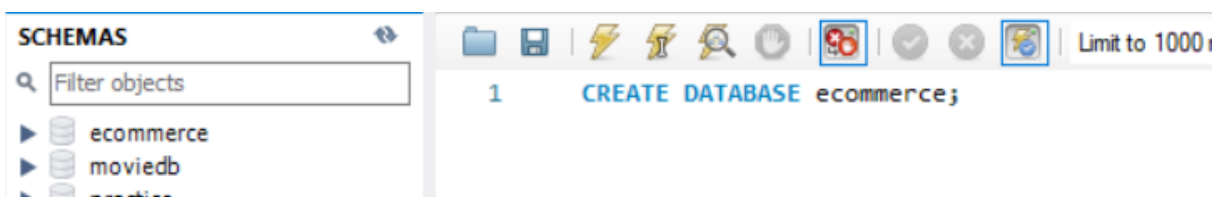
```
# 1. pymysql 모듈 import
import pymysql
```

```
# 2. pymysql.connect() 메소드를 사용하여 mysql에 연결
db = pymysql.connect(
    host = 'localhost',
    port = 3306,
    user='root',
    passwd = '1234',
    db = 'ecommerce',
    charset = 'utf8')
```

```
File ~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pymysql\er
148 if errorclass is None:
149     errorclass = InternalError if errno < 1000 else OperationalError
--> 150 raise errorclass(errno, errval)

OperationalError: (1049, "Unknown database 'ecommerce'")
```

위의 코드를 실행하면 ecommerce 라는 database가 없다는 오류문이 뜬다.
mysql에서 ecommerce 데이터 베이스를 만들어 주자.



```
db
```

```
<pymysql.connections.Connection at 0x2219a71b9b0>
```

ecommerce라는 데이터 베이스를 만든 후에 db를 출력하면 정상적으로 만들어진 것을 확인할 수 있다.

```
# 3. cursor() 메서드를 호출하여 cursor 객체 가져오기
ecommerce = db.cursor()
```

```
ecommerce
```

```
<pymysql.cursors.Cursor at 0x2219a8ec140>
```

4. Cursor 객체의 execute() 메서드를 사용하여 SQL 문장을 DB 서버에 전송
sql = """

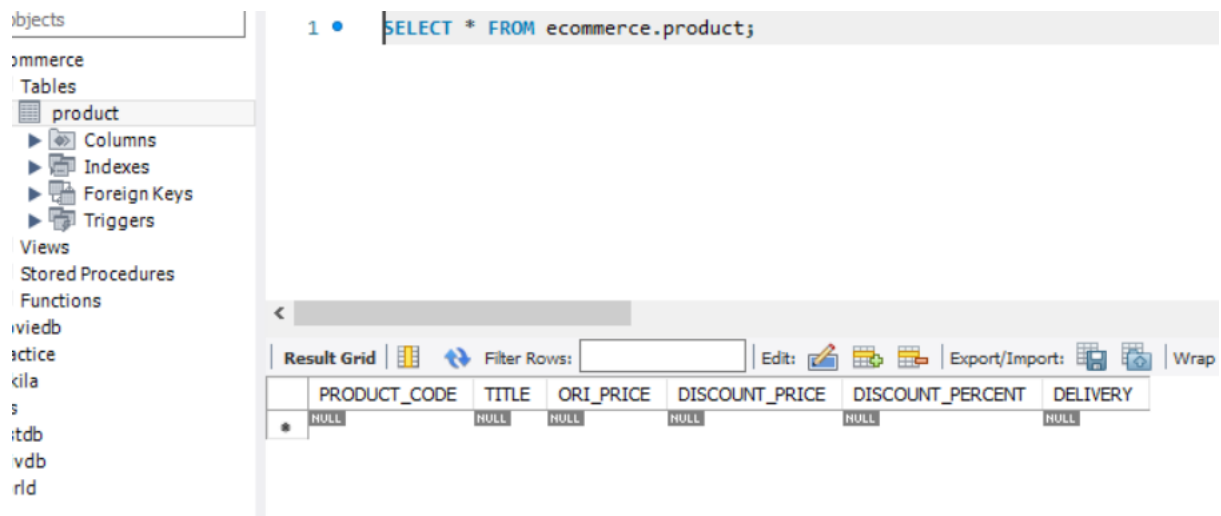
```
CREATE TABLE product(  
    PRODUCT_CODE VARCHAR(20) NOT NULL,  
    TITLE VARCHAR(200) NOT NULL,  
    ORI_PRICE INT,  
    DISCOUNT_PRICE INT,  
    DISCOUNT_PERCENT INT,  
    DELIVERY VARCHAR(2),  
    PRIMARY KEY(PRODUCT_CODE)  
);  
"""  
sql
```

```
'\n CREATE TABLE product(\n     PRODUCT_CODE VARCHAR(20) NOT NULL,\n     TITLE VARCHAR(200) NOT NULL,\n     ORI_PRICE INT,\n     DISCOUNT_PRICE INT,\n     DISCOUNT_PERCENT INT,\n     DELIVERY VARCHAR(2),\n     PRIMARY KEY(PRODUCT_CODE)\n );\n'
```

sql문을 저장한다. 이 sql문은 후에 실행하는 데 필요하다.

```
# 5. sql문 실행  
ecommerce.execute(sql)
```

0



mysql에서 product 테이블이 생성된 것을 확인할 수 있다.

```
# 6. SQL문 실행 내용을 데이터 베이스에 적용
db.commit()
```

```
# 7. 데이터 베이스를 모두 사용했으면 데이터베이스 close()
db.close()
```

여기까지 데이터 베이스에 접속해서 product 테이블을 만들고 연결을 닫는 것까지 한 내용이다.

INSERT

다음은 테이블에 내용을 넣는 코드를 실행해보자.

```
# 데이터 삽입(INSERT) 패턴 코드

# 1. 라이브러리 가져오기
import pymysql
```

```

# 2. 접속하기
db = pymysql.connect(
    host='localhost',
    port=3306,
    user='root',
    passwd='1234',
    db='ecommerce',
    charset='utf8')

# 3. 커서 가져오기
cursor = db.cursor()

# 4. sql문 작성 및 실행
for index in range(10):
    product_code = 215673140 + index + 1
    SQL = """INSERT INTO product VALUES('""" + str(product_code) +
        '스위트바니 여름신상 5900원 ~ 롱원피스 티셔츠/긴팔/반팔', 23000, 6)
    """
    print(SQL)
    cursor.execute(SQL)

# 5. 실행 결과 DB에 적용하기
db.commit()

# 6. db 연결닫기
db.close()

```

```

INSERT INTO product VALUES('215673141',
    '스위트바니 여름신상 5900원 ~ 롱원피스 티셔츠/긴팔/반팔', 23000, 6900, 70, 'F');

INSERT INTO product VALUES('215673142',
    '스위트바니 여름신상 5900원 ~ 롱원피스 티셔츠/긴팔/반팔', 23000, 6900, 70, 'F');

INSERT INTO product VALUES('215673143',
    '스위트바니 여름신상 5900원 ~ 롱원피스 티셔츠/긴팔/반팔', 23000, 6900, 70, 'F');

INSERT INTO product VALUES('215673144',
    '스위트바니 여름신상 5900원 ~ 롱원피스 티셔츠/긴팔/반팔', 23000, 6900, 70, 'F');

INSERT INTO product VALUES('215673145',
    '스위트바니 여름신상 5900원 ~ 롱원피스 티셔츠/긴팔/반팔', 23000, 6900, 70, 'F');

INSERT INTO product VALUES('215673146',
    '스위트바니 여름신상 5900원 ~ 롱원피스 티셔츠/긴팔/반팔', 23000, 6900, 70, 'F');

INSERT INTO product VALUES('215673147',
    '스위트바니 여름신상 5900원 ~ 롱원피스 티셔츠/긴팔/반팔', 23000, 6900, 70, 'F');

INSERT INTO product VALUES('215673148',
    '스위트바니 여름신상 5900원 ~ 롱원피스 티셔츠/긴팔/반팔', 23000, 6900, 70, 'F');

INSERT INTO product VALUES('215673149',
    '스위트바니 여름신상 5900원 ~ 롱원피스 티셔츠/긴팔/반팔', 23000, 6900, 70, 'F');

INSERT INTO product VALUES('215673150',
    '스위트바니 여름신상 5900원 ~ 롱원피스 티셔츠/긴팔/반팔', 23000, 6900, 70, 'F');

```

print(sql)로 SQL문이 제대로 적힌 것을 확인해보았다.

	PRODUCT_CODE	TITLE	ORI_PRICE	DISCOUNT_PRICE	DISCOUNT_PERCENT	DELIVERY
▶	215673141	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673142	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673143	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673144	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673145	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673146	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673147	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673148	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673149	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673150	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
★	NULL	NULL	NULL	NULL	NULL	NULL

mysql에서 내용이 insert된 것을 확인할 수 있다.

UPDATE

```
# 데이터 수정 (UPDATE) 패턴 코드
```

```
# 1. 라이브러리 가져오기
```

```
import pymysql
```

```

# 2. 접속하기
db = pymysql.connect(
    host='localhost',
    port=3306,
    user='root',
    passwd='1234',
    db='ecommerce',
    charset='utf8')

# 3. 커서 가져오기
cursor = db.cursor()

# 4. sql문 작성 및 실행
SQL = """
        UPDATE product SET
        TITLE = '달리샵 린넨 원피스 뷔스티에 썸머 가디건 코디전',
        ORI_PRICE=33000,
        DISCOUNT_PRICE=9900,
        DISCOUNT_PERCENT=70
        WHERE PRODUCT_CODE='215673141'
        """

cursor.execute(SQL)

# 5. 실행 결과 DB에 적용하기
db.commit()

# 6. db 연결닫기
db.close()

```

	PRODUCT_CODE	TITLE	ORI_PRICE	DISCOUNT_PRICE	DISCOUNT_PERCENT	DELIVERY
▶	215673141	달리샵 린넨 원피스 뷔스티에 썸머 가디건 코디전	33000	9900	70	F
	215673142	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673143	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673144	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673145	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673146	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673147	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673148	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673149	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673150	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
•	NULL	NULL	NULL	NULL	NULL	NULL

mysql에서 확인하면 PRODUCT_CODE=215673141인 행의 정보가 업데이트된 것을 확인할 수 있다.

DELETE

```
# 데이터 삭제(DELETE) 패턴 코드

# 1. 라이브러리 가져오기
import pymysql

# 2. 접속하기
db = pymysql.connect(
    host='localhost',
    port=3306,
    user='root',
    passwd='1234',
    db='ecommerce',
    charset='utf8')

# 3. 커서 가져오기
cursor = db.cursor()

# 4. sql문 작성 및 실행
SQL = """
    DELETE FROM product WHERE PRODUCT_CODE = '215673141'
    """
cursor.execute(SQL)

# 5. 실행 결과 DB에 적용하기
db.commit()

# 6. db 연결닫기
db.close()
```


	PRODUCT_CODE	TITLE	ORI_PRICE	DISCOUNT_PRICE	DISCOUNT_PERCENT	DELIVERY
▶	215673142	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673143	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673144	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673145	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673146	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673147	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673148	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673149	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
	215673150	스위트바니 여름신상 5900원 ~ 롱원피스 티셔...	23000	6900	70	F
★	NULL	NULL	NULL	NULL	NULL	NULL

mysql에서 PRODUCT_CODE=215673141인 행이 삭제된 것을 확인할 수 있다.

streamlit

db_test 디렉토리를 만들어서 다음의 두 sql 파일을 옮긴다.

sakila-schema.sql

mysql에서 위의 스키마 sql 파일을 실행한다.

sakila-data.sql

다음으로 위의 데이터 sql 파일을 실행한다.

```
python -m venv venv
```

cmd에서 db_test 디렉토리로 이동한 후, venv라는 가상환경을 만들어 준다.

```
venv\Scripts\activate.bat
```

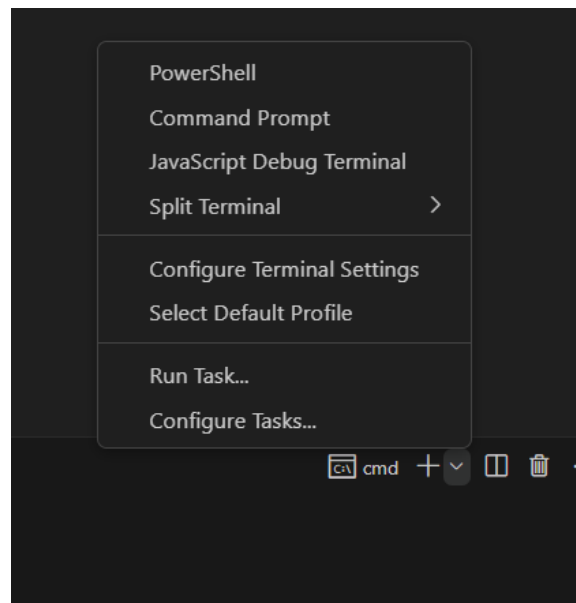
가상 환경에 접속한다.

```
(venv) C:\pmj\db_test>
```

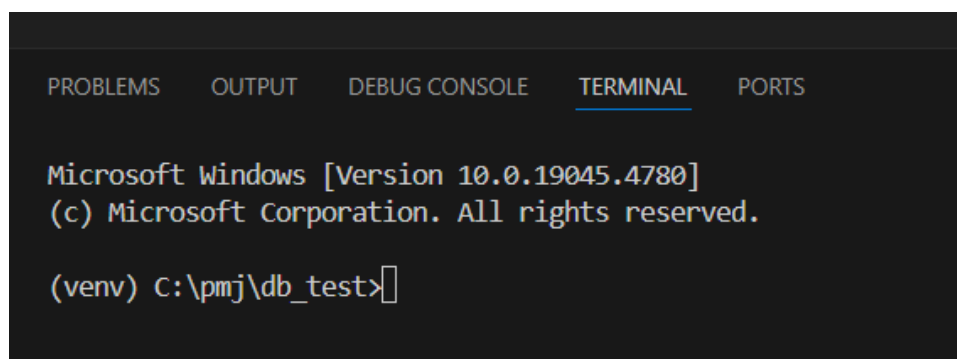
앞에 (venv)가 뜨면 가상 환경에 접속 완료한 것이다.

```
(venv) C:\pmj\db_test>code .
```

`code .` 를 실행하면 visual studio code가 열린다.



terminal 창에서 `Command Prompt` 를 클릭한다.



위와 같이 나오면 정상적으로 실행된 것이다. 앞으로 실행하는 코드는 모두 가상환경에서 실행된다.

```
pip list
```

```
(venv) C:\pmj\db_test>pip list
Package Version
-----
pip      24.0

[notice] A new release of pip is available: 24.0 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

(`python -m pip install --upgrade pip` 로 버전 업데이트를 해두자.)

```
pip install streamlit
```

streamlit을 설치한다.

```
pip list
```

```
(venv) C:\pmj\db_test>pip list
Package                               Version
-----
altair                                5.4.0
attrs                                 24.2.0
blinker                               1.8.2
cachetools                           5.5.0
certifi                              2024.7.4
charset-normalizer                    3.3.2
click                                 8.1.7
colorama                             0.4.6
gitdb                                 4.0.11
GitPython                             3.1.43
idna                                  3.7
Jinja2                                3.1.4
jsonschema                           4.23.0
jsonschema-specifications             2023.12.1
markdown-it-py                        3.0.0
MarkupSafe                           2.1.5
mdurl                                 0.1.2
```

다시 pip list를 하면 설치된 package들이 정렬된다.

가상환경에서 실행하기 때문에 여태까지 깔았었던 모듈을 다시 깔아야 한다.

```
pip install pymysql
pip install matplotlib
```

`app.py` 파일을 만들어준다.

```
import streamlit as st
import pymysql
import pandas as pd
import matplotlib.pyplot as plt

# Streamlit App
st.title("Sakila Movie Database Management")
```

모듈을 다시 깔고 실행시키면 오류 없이 실행시킬 수 있다.

terminal 창(현재 venv에서 실행 중)에서 다음을 실행시킨다.

```
streamlit run app.py
```

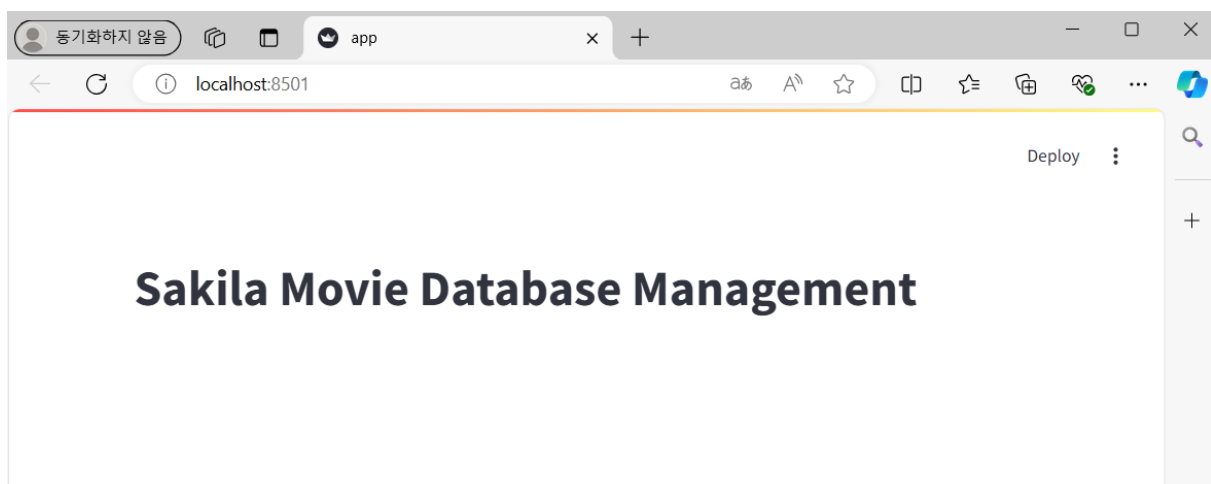
```
(venv) C:\pmj\db_test>streamlit run app.py

Welcome to Streamlit!

If you'd like to receive helpful onboarding emails, news, offers, promotions,
and the occasional swag, please enter your email address below. Otherwise,
leave this field blank.

Email: 
```

Email을 입력하라는 커서에서 Enter키를 누른다.



서버 창이 뜬다.

```
### app.py ###

import streamlit as st
import pymysql
import pandas as pd
import matplotlib.pyplot as plt

# Streamlit App
st.title("Sakila Movie Database Management")
```


```
# Main menu
menu = ["View Movies"]

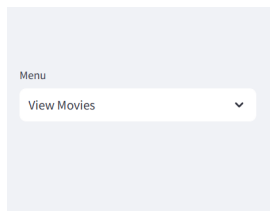
choice = st.sidebar.selectbox("Menu", menu)
```



수정된 py 파일을 브라우저 창에 적용하는 법

py 파일을 저장한 후 브라우저 창에서 Rerun 버튼을 누르면 된다.
(py 파일을 실행하지 않는 것에 주의하자.)

 Source file changed. Rerun Always rerun Deploy 



Deploy 

Sakila Movie Database Management

```
import streamlit as st
import pymysql
import pandas as pd
import matplotlib.pyplot as plt

# 데이터베이스 연결 함수
def connect_to_db():
    try:
        conn = pymysql.connect(
```

```

        host='localhost',
        port=3306,
        user='root',
        passwd='1234',
        db='sakila',
        charset='utf8'
    )
    return conn

# 예외 처리
except pymysql.connect.Error as err:
    st.error(f"Error: {err}")
    return None

def fetch_movies():
    conn = connect_to_db()

    # 데이터베이스와 연결이 잘 되었다면 if문 실행
    if conn:
        cursor = conn.cursor()
        cursor.execute("SELECT film_id, title, release_year FROM movies")
        movies = cursor.fetchall()
        cursor.close()
        conn.close()
        return movies
    return []

# Streamlit App
st.title("Sakila Movie Database Management")

# Main menu
menu = ["View Movies"]

choice = st.sidebar.selectbox("Menu", menu)

# View Movies를 누른다면
if choice == "View Movies":
    st.subheader("View Movie List")

```

```
movies = fetch_movies()
if movies:
    for movie in movies:
        st.write(movie)
else:
    st.write("No movies found.")
```

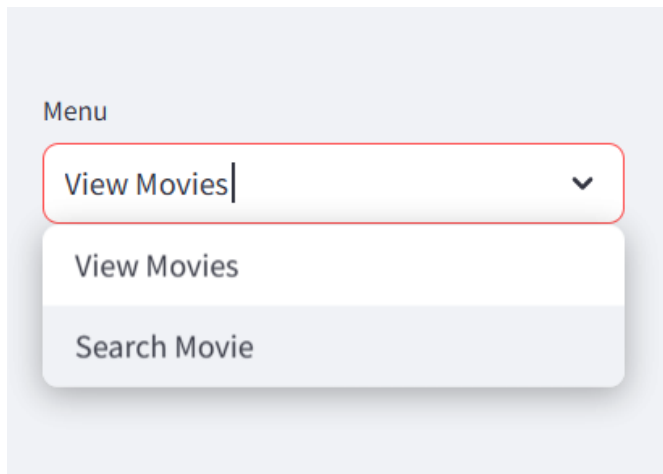
Sakila Movie Database Management

View Movie List

```
(1, 'ACADEMY DINOSAUR', 2006)
(2, 'ACE GOLDFINGER', 2006)
(3, 'ADAPTATION HOLES', 2006)
(4, 'AFFAIR PREJUDICE', 2006)
(5, 'AFRICAN EGG', 2006)
(6, 'AGENT TRUMAN', 2006)
(7, 'AIRPLANE SIERRA', 2006)
(8, 'AIRPORT POLLOCK', 2006)
(9, 'ALABAMA DEVIL', 2006)
(10, 'ALADDIN CALENDAR', 2006)
(11, 'ALAMO VIDEOTAPE', 2006)
```

db에 있는 movies가 잘 넘어온 것을 확인할 수 있다.

```
menu = ["View Movies", "Search Movie"]
```

`app.py` 코드에서 menu 요소를 추가하면 menu가 늘어난다.

```
# Search Movie를 누른다면
elif choice == "Search Movie":
    st.subheader("Search for a Movie")
```

Search Movie를 누르는 경우의 if문을 추가한다.



```
import streamlit as st
import pymysql
import pandas as pd
import matplotlib.pyplot as plt

# 데이터베이스 연결 함수
def connect_to_db():
    try:
        conn = pymysql.connect(
            host='localhost',
```

```

        port=3306,
        user='root',
        passwd='1234',
        db='sakila',
        charset='utf8'
    )
    return conn

# 예외 처리
except pymysql.connect.Error as err:
    st.error(f"Error: {err}")
    return None

def fetch_movies():
    conn = connect_to_db()

    # 데이터베이스와 연결이 잘 되었다면 if문 실행
    if conn:
        cursor = conn.cursor()
        cursor.execute("SELECT film_id, title, release_year FROM film")
        movies = cursor.fetchall()
        cursor.close()
        conn.close()
        return movies
    return []

def fetch_movie(title):
    conn = connect_to_db()

    if conn:
        cursor = conn.cursor()
        SQL = """SELECT film_id, title, release_year FROM film"""
        cursor.execute(SQL)
        rows = cursor.fetchall()
        return rows
    return None

```

```

# Streamlit App
st.title("Sakila Movie Database Management")

# Main menu
menu = ["View Movies", "Search Movie", "View Movie Details"]

choice = st.sidebar.selectbox("Menu", menu)

# View Movies를 누른다면
if choice == "View Movies":
    st.subheader("View Movie List")
    movies = fetch_movies()
    if movies:
        for movie in movies:
            st.write(f"ID: {movie[0]}, Title: {movie[1]}, Yea
    else:
        st.write("No movies found.")

# Search Movie를 누른다면
elif choice == "Search Movie":
    st.subheader("Search for a Movie")
    title = st.text_input("Enter movie title")

    if st.button("Search"):
        movies = fetch_movie(title)
        if movies:
            for movie in movies:
                st.write(f"ID: {movie[0]}, Title: {movie[1]},
        else:
            st.write("No movies found.")

```

Search for a Movie

Enter movie title

ACADEMY DINOSAUR

Search

(1, 'ACADEMY DINOSAUR', 2006)

View Movies에서 나오는 영화 제목을 서치 박스에 넣고 Search 버튼을 누르면 해당 영화의 정보를 보여준다.

```
if st.button("Search"):
    movies = fetch_movie(title)
    if movies:
        for movie in movies:
            st.write(f"ID: {movie[0]}, Title: {movie[1]},
    else:
        st.write("No movies found.")
```

Search for a Movie

Enter movie title

ADAPTATION HOLES

Search

ID: 3, Title: ADAPTATION HOLES, Year: 2006

`st.write(f"ID: {movie[0]}, Title: {movie[1]}, Year: {movie[2]}")` 문장을 통해 출력하는 텍스트를 수정할 수도 있다.

```

#### app.py ####

import streamlit as st
import pymysql
import pandas as pd
import matplotlib.pyplot as plt

# 데이터베이스 연결 함수
def connect_to_db():
    try:
        conn = pymysql.connect(
            host='localhost',
            port=3306,
            user='root',
            passwd='1234',
            db='sakila',
            charset='utf8'
        )
        return conn

    # 예외 처리
    except pymysql.connect.Error as err:
        st.error(f"Error: {err}")
        return None

def fetch_movies():
    conn = connect_to_db()

    # 데이터베이스와 연결이 잘 되었다면 if문 실행
    if conn:
        cursor = conn.cursor()
        cursor.execute("SELECT film_id, title, release_year FROM film")
        movies = cursor.fetchall()
        cursor.close()
        conn.close()
        return movies

```

```

        return []

def fetch_movie(title):
    conn = connect_to_db()

    if conn:
        cursor = conn.cursor()
        SQL = """SELECT film_id, title, release_year FROM film
        cursor.execute(SQL)
        rows = cursor.fetchall()
        return rows
    return None

def fetch_movie_details(film_id):
    conn = connect_to_db()

    if conn:
        cursor = conn.cursor()
        cursor.execute("""
                        SELECT title, description, release_year,
                        FROM film WHERE film_id = %s;
                        """, film_id)
        movie_detail = cursor.fetchone()
        cursor.close()
        conn.close()
        return movie_detail

    return None

# Streamlit App
st.title("Sakila Movie Database Management")

# Main menu
menu = ["View Movies", "Search Movie", "View Movie Details"]

choice = st.sidebar.selectbox("Menu", menu)

# View Movies를 누른다면

```

```

if choice == "View Movies":
    st.subheader("View Movie List")
    movies = fetch_movies()
    if movies:
        for movie in movies:
            st.write(f"ID: {movie[0]}, Title: {movie[1]}, Yea
    else:
        st.write("No movies found.")

# Search Movie를 누른다면
elif choice == "Search Movie":
    st.subheader("Search for a Movie")
    title = st.text_input("Enter movie title")

    if st.button("Search"):
        movies = fetch_movie(title)
        if movies:
            for movie in movies:
                st.write(f"ID: {movie[0]}, Title: {movie[1]},
        else:
            st.write("No movies found.")

# View Movie Details를 누른다면
elif choice == "View Movie Details":
    st.subheader("View Movie Details")
    film_id = st.text_input("Enter movie ID")
    if st.button("Get Details"):
        details = fetch_movie_details(film_id)
        if details:
            st.write(f"Title: {details[0]}")
            st.write(f"Description: {details[1]}")
            st.write(f"Release Year: {details[2]}")
            st.write(f"Length: {details[3]} minutes")
            st.write(f"Rating: {details[4]}")
        else:
            st.write("Movie not found")

```

View Movie Details

Enter movie ID

3

Get Details

Title: ADAPTATION HOLES

Description: A Astounding Reflection of a Lumberjack And a Car who must Sink a Lumberjack in A Baloon Factory

Release Year: 2006

Length: 50 minutes

Rating: NC-17

```
### app.py ###

import streamlit as st
import pymysql
import pandas as pd
import matplotlib.pyplot as plt

# 데이터베이스 연결 함수
def connect_to_db():
    try:
        conn = pymysql.connect(
            host='localhost',
            port=3306,
            user='root',
            passwd='1234',
            db='sakila',
            charset='utf8'
        )
        return conn

    # 예외 처리
```



```

except pymysql.connect.Error as err:
    st.error(f"Error: {err}")
    return None

def fetch_movies():
    conn = connect_to_db()

    # 데이터베이스와 연결이 잘 되었다면 if문 실행
    if conn:
        cursor = conn.cursor()
        cursor.execute("SELECT film_id, title, release_year FROM film")
        movies = cursor.fetchall()
        cursor.close()
        conn.close()
        return movies
    return []

def fetch_movie(title):
    conn = connect_to_db()

    if conn:
        cursor = conn.cursor()
        SQL = """SELECT film_id, title, release_year FROM film WHERE title = %s;"""
        cursor.execute(SQL)
        rows = cursor.fetchall()
        return rows
    return None

def fetch_movie_details(film_id):
    conn = connect_to_db()

    if conn:
        cursor = conn.cursor()
        cursor.execute("""
            SELECT title, description, release_year,
            FROM film WHERE film_id = %s;
            """, film_id)
        movie_detail = cursor.fetchone()

```

```

        cursor.close()
        conn.close()
        return movie_detail

    return None

def update_movie(film_id, title, description):
    conn = connect_to_db()

    if conn:
        cursor = conn.cursor()
        cursor.execute("""
            UPDATE film
            SET title = %s, description = %s
            WHERE film_id = %s;
            """, (title, description, film_id))
        conn.commit()
        cursor.close()
        conn.close()
        return True
    return None

# Streamlit App
st.title("Sakila Movie Database Management")

# Main menu
menu = ["View Movies", "Search Movie", "View Movie Details",

choice = st.sidebar.selectbox("Menu", menu)

# View Movies를 누른다면
if choice == "View Movies":
    st.subheader("View Movie List")
    movies = fetch_movies()
    if movies:
        for movie in movies:
            st.write(f"ID: {movie[0]}, Title: {movie[1]}, Yea
    else:

```

```

        st.write("No movies found.")

# Search Movie를 누른다면
elif choice == "Search Movie":
    st.subheader("Search for a Movie")
    title = st.text_input("Enter movie title")

    if st.button("Search"):
        movies = fetch_movie(title)
        if movies:
            for movie in movies:
                st.write(f"ID: {movie[0]}, Title: {movie[1]},
            else:
                st.write("No movies found.")

# View Movie Details를 누른다면
elif choice == "View Movie Details":
    st.subheader("View Movie Details")
    film_id = st.text_input("Enter movie ID")
    if st.button("Get Details"):
        details = fetch_movie_details(film_id)
        if details:
            st.write(f"Title: {details[0]}")
            st.write(f"Description: {details[1]}")
            st.write(f"Release Year: {details[2]}")
            st.write(f"Length: {details[3]} minutes")
            st.write(f"Rating: {details[4]}")
        else:
            st.write("Movie not found")

# Update Movie를 누른다면
elif choice == "Update Movie":
    st.subheader("Update Movie")
    film_id = st.text_input("Enter movie ID")
    title = st.text_input("Enter new title")
    description = st.text_area("Enter new description")
    if st.button("Update Movie"):
        if update_movie(film_id, title, description):

```

```
        st.success("Movie update successfully")
    else:
        st.error("Failed to update movie.")
```

Update Movie

Enter movie ID

3

Enter new title

new title

Enter new description

new description

Update Movie

Movie update successfully

movie ID를 입력한 후, 새로운 title과 description을 적고 Update Movie를 누르면 성공적으로 업데이트됐다는 메시지가 뜬다.

View Movie Details

Enter movie ID

3

Get Details

Title: new title

Description: new description

Release Year: 2006

Length: 50 minutes

Rating: NC-17

View Movie Details에서 확인하면 title과 description이 변경된 것을 확인할 수 있다.

app.py

새로운 `app1.py` 을 만들고 Command Prompt에서 `streamlit run app1.py` 을 실행한다.

```
#### app1.py ####

import streamlit as st
import time
import numpy as np

st.write("streamlit version = {}".format(st.__version__))

progress_bar = st.sidebar.progress(0)
status_text = st.sidebar.empty()
last_rows = np.random.randn(1,1)
chart = st.line_chart(last_rows)

for i in range(1, 101):
    new_rows = last_rows[-1, :] + np.random.randn(5,1).cumsum(1)
    status_text.text("%i%% Complete" %i)
```

```

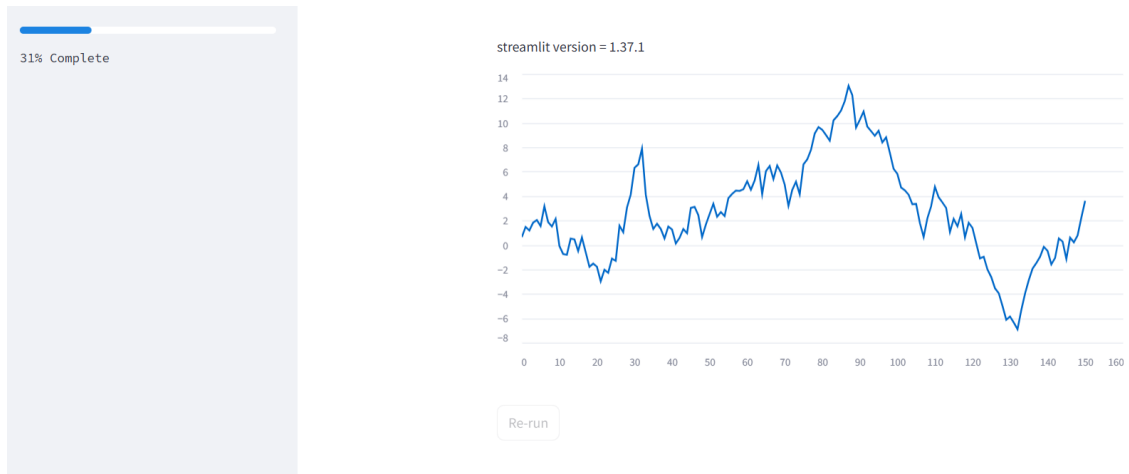
progress_bar.progress(i)
chart.add_rows(new_rows)
last_rows = new_rows
time.sleep(0.1)

```

```

progress_bar.empty()
st.button("Re-run")

```



진행 상황을 사이드바로 보여준다.



그래프 출력이 완료되면 Re-run 버튼을 눌러 다시 실행시킬 수 있다.

새로운 파일 `app2.py` 를 만들고 Command Prompt에서 `streamlit run app2.py` 를 실행한다.

```
#### app2.py ####

import streamlit as st
import pandas as pd
import numpy as np

# st.write

st.write('Welcom to streamlit App APIS')
st.write(1234)

df = pd.DataFrame({
    'first': [1,2,3,4,5],
    'second': [6,7,8,9,10]
})

st.write(df)
```

Welcom to streamlit App APIS

1234

	first	second
0	1	6
1	2	7
2	3	8
3	4	9
4	5	10

streamlit.write 함수는 body 부분에 출력한다.
문자열, 숫자, 데이터프레임 모두 출력 가능하다.

```
# display numpy array
st.write(np.array([1,2,3,4,5]))

st.write("Magic commands")
df1 = pd.DataFrame({'col':[1,2,3,4,5]})
df1
```

value
1
2
3
4
5

Magic commands

	col
0	1
1	2
2	3
3	4
4	5

```
# title
st.title('This is Title')
st.caption("Using st.title()")

# header
```



```
st.header("This is header")
st.caption("The text inside st.header()")
```

This is Title

Using st.title()

This is header

The text inside st.header()

```
st.markdown("----")
st.subheader("Generate Random Numbers")
body = """
import numpy as np

def generate_random(size):
    rand = np.random.random(size=size)
    return rand
number = generate_random(10)
"""
st.code(body, language='python')
```

Generate Random Numbers

```
import numpy as np

def generate_random(size):
    rand = np.random.random(size=size)
    return rand
number = generate_random(10)
```

```
# Latex
st.subheader('Latex')
formula = """
a + ar + ar^2 + ar^3 + \cdots + ar^{n-1} = \sum_{k=0}^{nb-1} a
"""
st.latex(formula)
```

Latex

$$a + ar + ar^2 + ar^3 + \cdots + ar^{n-1} = \sum_{k=0}^{nb-1} ar^k$$

새로운 파일인 `app3.py` 을 만들어 Command Prompt에서 `streamlit run app3.py` 을 실행하자.

디렉토리에 데이터프레임 파일인 `tips.csv` 을 추가한다.

`tips.csv`

```
### app3.py ###

import streamlit as st
import pandas as pd
import numpy as np

df = pd.read_csv('tips.csv')

st.header('st.dataframe')
st.dataframe(data=df, width=1000, height=1000)
```

st.dataframe

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.5	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
5	25.29	4.71	Male	No	Sun	Dinner	4
6	8.77	2	Male	No	Sun	Dinner	2
7	26.88	3.12	Male	No	Sun	Dinner	4
8	15.04	1.96	Male	No	Sun	Dinner	2
9	14.78	3.23	Male	No	Sun	Dinner	2
10	10.27	1.71	Male	No	Sun	Dinner	2
11	35.26	5	Female	No	Sun	Dinner	4
12	15.42	1.57	Male	No	Sun	Dinner	2
13	18.43	3	Male	No	Sun	Dinner	4

dataframe을 불러올 수 있다.

```
# st.static
st.header("st.table")
st.table(data=df.head(5))
```

st.table

	total_bill	tip	sex	smoker	day	time	size
0	16.9900	1.0100	Female	No	Sun	Dinner	2
1	10.3400	1.6600	Male	No	Sun	Dinner	3
2	21.0100	3.5000	Male	No	Sun	Dinner	3
3	23.6800	3.3100	Male	No	Sun	Dinner	2
4	24.5900	3.6100	Female	No	Sun	Dinner	4

```
# st.json
st.header("st.json")
json_values = df.head(3).to_dict()
st.json(json_values)
```

st.json

```
{
  "total_bill": {
    "0": 16.99
    "1": 10.34
    "2": 21.01
  },
  "tip": {
    "0": 1.01
    "1": 1.66
    "2": 3.5
  },
  "sex": {
    "0": "Female"
    "1": "Male"
    "2": "Male"
  },
  "smoker": {
    "0": "No"
    "1": "No"
    "2": "No"
  }
}
```

새로운 파일인 `app4.py` 을 만들어서 Command Prompt에서 `streamlit run app4.py` 을 실행한다.

```
### app4.py ###

import streamlit as st
import time

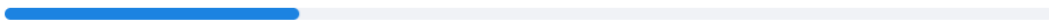
# progress
st.header("st.progress")
```

```
st.caption("Display a progress bar")

my_bar = st.progress(0)
for pct_complete in range(1,101):
    time.sleep(0.5)
    my_bar.progress(pct_complete)
```


st.progress

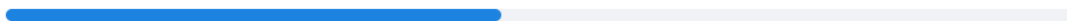
Display a progress bar



```
def progress_bar():
    for pct_complete in range(1,101):
        time.sleep(0.1)
        pct_complete = min(pct_complete, 100)
        my_bar.progress(pct_complete)

# spinner
with st.spinner("Something is processing..."):
    my_bar = st.progress(0)
    progress_bar()
```

 Something is processing...



```
st.subheader("st.info")
st.info("This is Information message")

st.subheader("st.success")
```

```
st.success("This is Success message")

st.subheader("st.warning")
st.warning("This is Warning message")

st.subheader("st.error")
st.error("This is Error message")
```

st.info

This is Information message

st.success

This is Success message

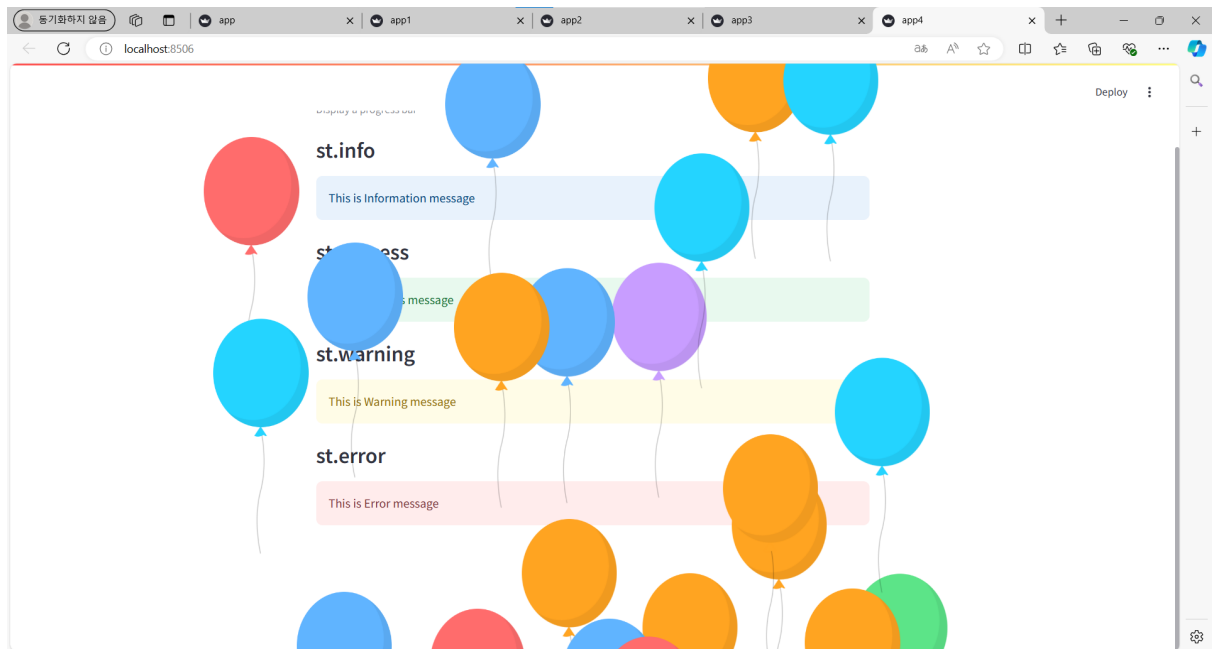
st.warning

This is Warning message

st.error

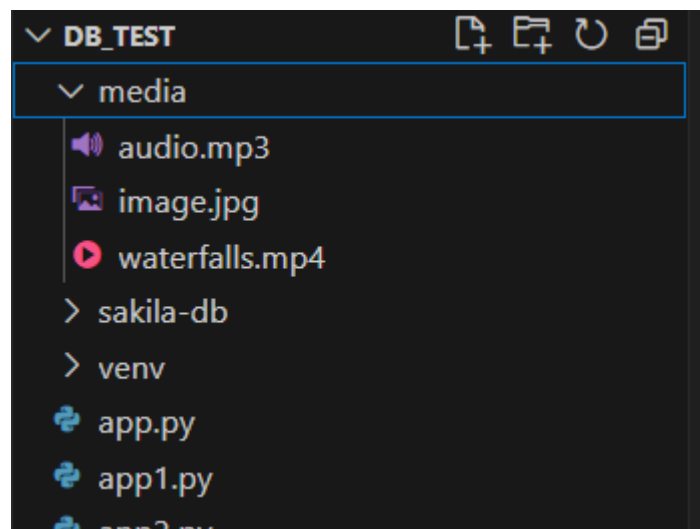
This is Error message

```
time.sleep(1)
st.balloons()
```



1초 기다렸다가 풍선이 창에 뜬다.

media 압축파일을 db_test 디렉토리에 압축풀기 한다.



새로운 파일 `app5.py` 을 추가한다.

```
### app5.py ###

import streamlit as st

st.header("Display image using st.image")
st.image("./media/image.jpg", caption = "Beautiful City", wid

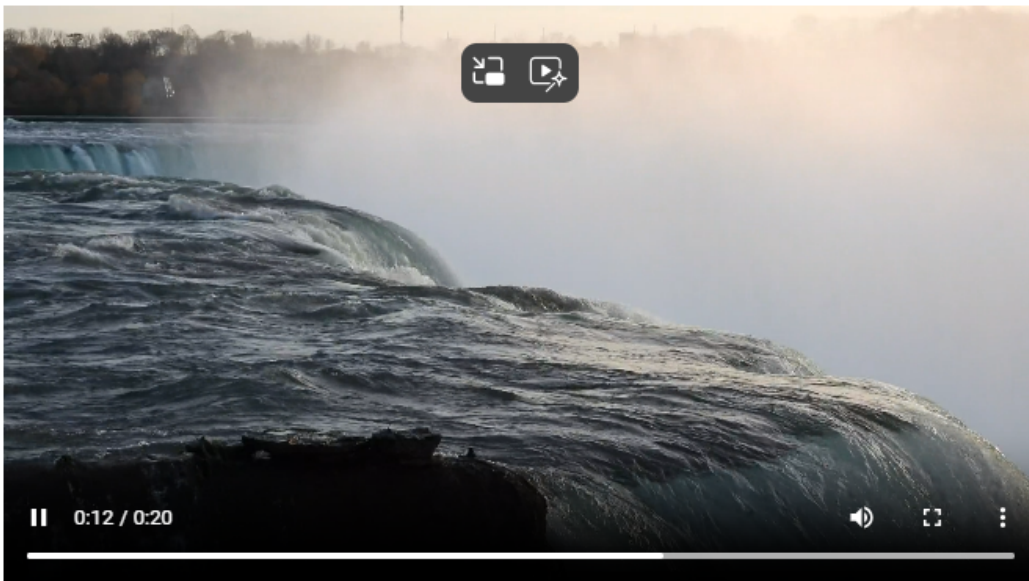
st.header("Display video")
video_file = open("./media/waterfalls.mp4", "rb")
video_bytes = video_file.read()
st.video(video_bytes)
```


Display image using st.image



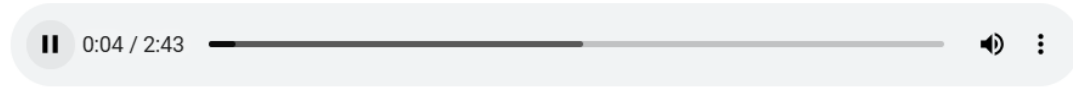
Beautiful City

Display video



```
# display audio
st.header("Display audio")
audio_file = open("./media/audio.mp3", "rb")
audio_bytes = audio_file.read()
st.audio(audio_bytes, format="audio/ogg")
```

Display audio



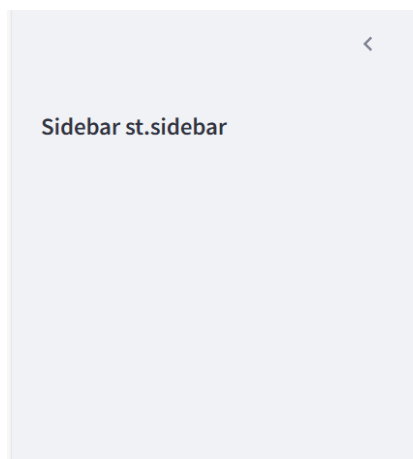
새로운 파일 `app6.py` 을 추가한다.

```
#### app6.py ##3

import streamlit as st
import pandas as pd
import time

side_bar = st.sidebar

side_bar.header("Sidebar st.sidebar")
```



사이드바가 추가된다.

```
df = pd.read_csv("tips.csv")
columns = tuple(df.columns)
```

```

select_column = side_bar.selectbox(
    "Select the column you want to display",
    columns
)

side_bar.write("You selected the column_name = {}".format(select_column))

st.dataframe(df[[select_column]])

```

Sidebar st.sidebar

Select the column you want to display

tip

total_bill

tip

sex

smoker

day

time

size

Sidebar st.sidebar

Select the column you want to display

size

You selected the column_name = size

	size
0	2
1	3
2	3
3	2
4	4
5	4
6	2
7	4
8	2
9	2

```

# Layout Columns
st.header("Columns")
col1, col2, col3 = st.columns(3)

with col1:
    st.subheader("column-1")
    st.image("./media/image.jpg")

with col2:
    st.subheader("column-2")
    st.dataframe(df)

with col3:
    st.subheader("column-3")
    st.dataframe(df[[select_column]])

```

Columns

column-1



column-2

	total_bill	tip	sex
0	16.99	1.01	Female
1	10.34	1.66	Male
2	21.01	3.5	Male
3	23.68	3.31	Male
4	24.59	3.61	Female
5	25.29	4.71	Male
6	8.77	2	Male
7	26.88	3.12	Male
8	15.04	1.96	Male
9	14.78	3.23	Male

column-3

	size
0	2
1	3
2	3
3	2
4	4
5	4
6	2
7	4
8	2
9	2

```
# expander
st.header("Expander: st.expander")
with st.expander("Some explanation"):
    st.write("""
        Insert a multi-element

        Insert a container into your app
        """)
```

Expander: st.expander

Some explanation

Expander: st.expander

Some explanation

Insert a multi-element

Insert a container into your app

expander를 누르면 밑의 항목들이 나온다.

새로운 파일 `app7.py` 을 추가한다.

Command Prompt에서 `pip install seaborn` 을 실행한 후, `streamlit run app7.py` 을 실행한다.

```
#### app7.py ####

from re import L
import streamlit as st
import pandas as pd
import numpy as np

# static
import matplotlib.pyplot as plt
import seaborn as sns

st.header("matplotlib and seaborn")
df = pd.read_csv("./tips.csv")
st.dataframe(df.head())
```

matplotlib and seaborn

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.5	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

```

st.markdown("---")

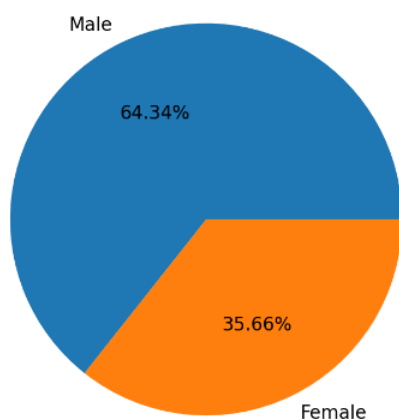
with st.container():
    st.write("1. 남성과 여성의 분포 수를 찾기")
    value_counts = df['sex'].value_counts()
    col1, col2 = st.columns(2)
    with col1:
        st.subheader("Pie Chart")
        fig, ax = plt.subplots()
        ax.pie(value_counts, autopct="%0.2f%%", labels=["Male", "Female"])
        st.pyplot(fig)

    with col2:
        st.subheader('Bar Chart')
        fig, ax = plt.subplots()
        ax.bar(value_counts.index, value_counts)
        st.pyplot(fig)

```

1. 남성과 여성의 분포 수를 찾기

Pie Chart



Bar Chart

