Pandas 기본 및 Seaborn 기본

In [15]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm

# 맑은 고딕 폰트 설정
font_name = fm.FontProperties(fname="c:/Windows/Fonts/malgun.ttf").get_name()
plt.rc('font', family=font_name)

DATE_COLUMN = 'date/time'
DATA_URL = ('https://s3-us-west-2.amazonaws.com/streamlit-demo-data/uber-raw-dat
```

데이터 불러오기

`pd.read_csv(DATA_URL, nrows=nrows)`

- `DATA_URL` 은 데이터 파일의 URL 주소입니다.
- `nrows=nrows` 는 읽어들일 행의 수를 지정합니다. 이 경우 `nrows` 매개변수로 전달된 값만큼의 행을 읽어들입니다.
- 이 코드는 지정된 URL에서 CSV 파일을 읽어들여 Pandas DataFrame 형태로 데이터를 가져옵니다.

`lowercase = lambda x: str(x).lower()`

- 이 부분은 열 이름을 모두 소문자로 변환하기 위한 람다 함수입니다.
- 데이터 프레임의 열 이름을 모두 소문자로 변환하면 데이터 처리 및 분석 작업이 편리해집니다.

`data.rename(lowercase, axis='columns', inplace=True)`

- 이 코드는 데이터 프레임의 열 이름을 소문자로 변환합니다.
- `lowercase` 함수를 사용하여 각 열 이름을 소문자로 변환합니다.
- `axis='columns'` 는 열 이름을 변경한다는 것을 의미합니다.
- `inplace=True` 는 원본 데이터 프레임을 직접 수정한다는 것을 의미합니다.

`data[DATE_COLUMN] = pd.to_datetime(data[DATE_COLUMN])`

- `DATE_COLUMN` 은 데이터 프레임의 날짜/시간 열 이름입니다.
- `pd.to_datetime(data[DATE_COLUMN])` 은 해당 열의 데이터를 datetime 형식으로 변환합니다.
- 이 작업을 통해 날짜/시간 데이터를 Pandas에서 효과적으로 처리할 수 있습니다.

`return data`

- 이 함수는 전처리된 데이터 프레임을 반환합니다.

In [19]:
```
# 데이터 불러오기
def load_data(nrows):
    data = pd.read_csv(DATA_URL, nrows=nrows)
    lowercase = lambda x: str(x).lower()
    data.rename(lowercase, axis='columns', inplace=True)
    data[DATE_COLUMN] = pd.to_datetime(data[DATE_COLUMN])
    return data

# 10000개의 행의 데이터를 로드한다.
data = load_data(10000)
print( data.shape )    # 데이터의 정보
print( data.columns )  # 데이터의 컬럼 정보
data.head(5)
```

```
(10000, 4)
Index(['date/time', 'lat', 'lon', 'base'], dtype='object')
```

Out[19]:

	date/time	lat	lon	base
0	2014-09-01 00:01:00	40.2201	-74.0021	B02512
1	2014-09-01 00:01:00	40.7500	-74.0027	B02512
2	2014-09-01 00:03:00	40.7559	-73.9864	B02512
3	2014-09-01 00:06:00	40.7450	-73.9889	B02512
4	2014-09-01 00:11:00	40.8145	-73.9444	B02512

In [17]:
```
# 원본 데이터 출력
print('원본 데이터:')
print(data)

# seaborn을 사용하여 히스토그램 그리기
print('시간대별 픽업 횟수:')
```

```
원본 데이터:
                date/time      lat      lon    base
0     2014-09-01 00:01:00  40.2201 -74.0021  B02512
1     2014-09-01 00:01:00  40.7500 -74.0027  B02512
2     2014-09-01 00:03:00  40.7559 -73.9864  B02512
3     2014-09-01 00:06:00  40.7450 -73.9889  B02512
4     2014-09-01 00:11:00  40.8145 -73.9444  B02512
...                   ...      ...      ...     ...
9995  2014-09-08 18:15:00  40.7194 -74.0000  B02512
9996  2014-09-08 18:15:00  40.7426 -74.0079  B02512
9997  2014-09-08 18:16:00  40.7358 -73.9758  B02512
9998  2014-09-08 18:16:00  40.7385 -73.9952  B02512
9999  2014-09-08 18:16:00  40.7279 -73.9961  B02512

[10000 rows x 4 columns]
시간대별 픽업 횟수:
```

In [18]:
```
# Seaborn을 사용하여 히스토그램 그리기
plt.figure(figsize=(10, 6))
sns.histplot(data=data, x=data[DATE_COLUMN].dt.hour, bins=24, kde=False)
```

```
plt.xlabel('시간')
plt.ylabel('픽업 횟수')
plt.title('시간대별 픽업 횟수')
plt.show()
```