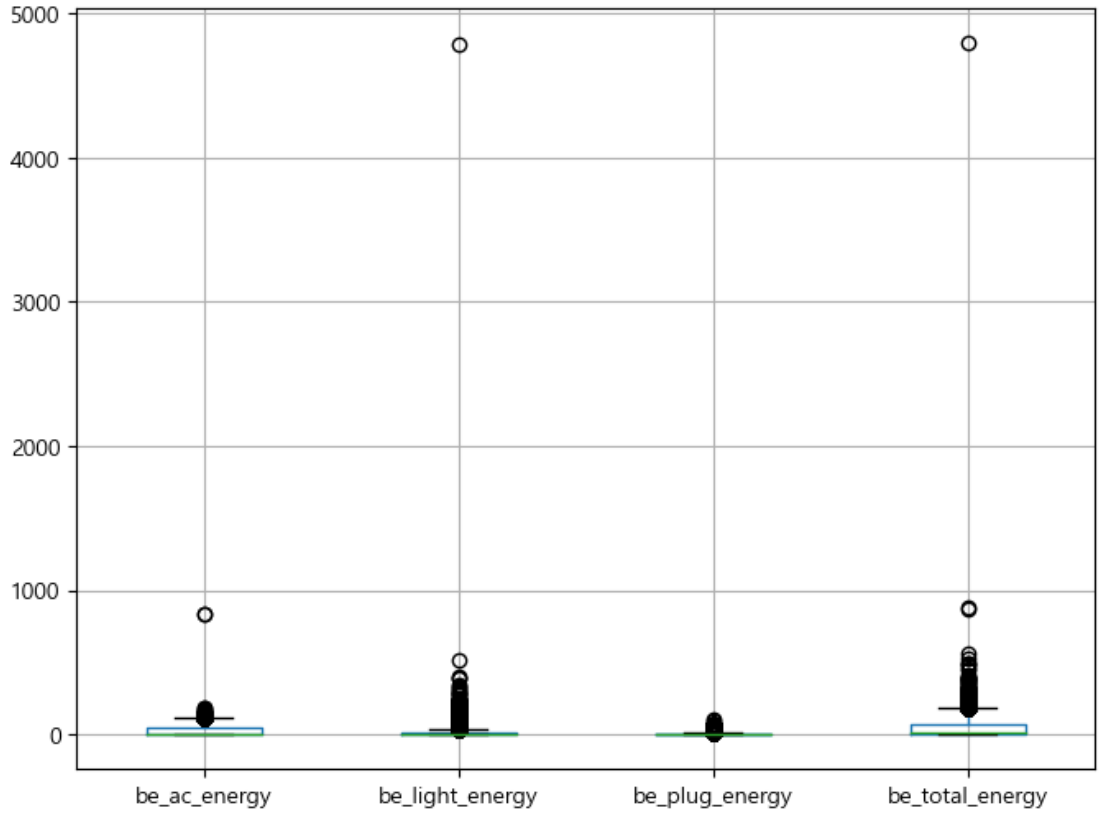
In [2]:
```
#한글깨짐
import matplotlib.pyplot as plt
from matplotlib import rc
%matplotlib inline
from matplotlib import font_manager
f_path = "C:/windows/Fonts/malgun.ttf"
font_manager.FontProperties(fname=f_path).get_name()
rc('font', family='Malgun Gothic')
```

In [10]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
import numpy as np
```

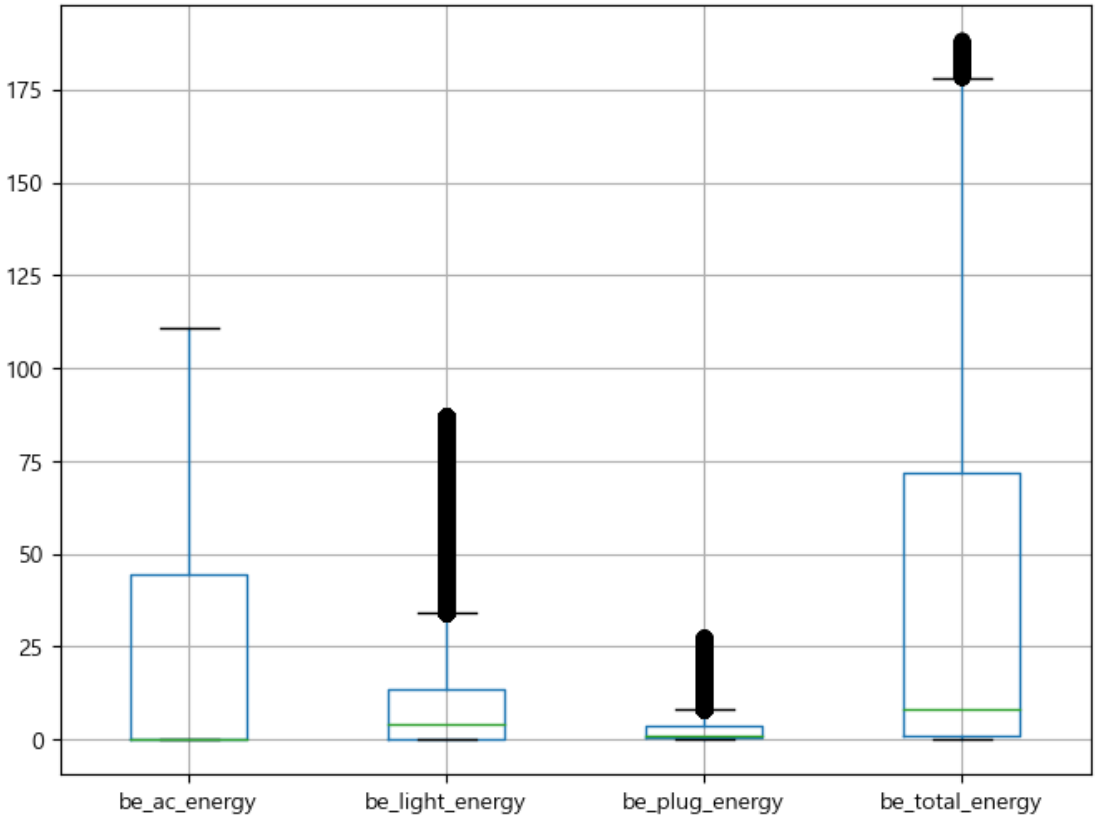
In [26]:
```

# CSV 파일을 읽어들임
df = pd.read_csv('total_data.csv')

# 'be_date' 칼럼을 datetime 형식으로 변환
df['be_date'] = pd.to_datetime(df['be_date'])

# 'be_date' 칼럼을 인덱스로 설정
#df.set_index('be_date', inplace=True)

# 데이터프레임 구조 확인
print(df.head())
```

```
              be_date  be_ac_energy  be_light_energy  be_plug_energy  ₩
0 2018-07-01 00:00:00         45.26            73.38           37.09
1 2018-07-01 00:01:00         45.32            73.49           36.63
2 2018-07-01 00:02:00         45.27            73.48           36.59
3 2018-07-01 00:03:00         45.29            73.56           36.07
4 2018-07-01 00:04:00         45.26            73.67           35.14

   be_total_energy  be_floor
0           155.73         1
1           155.44         1
2           155.34         1
3           154.92         1
4           154.07         1
```

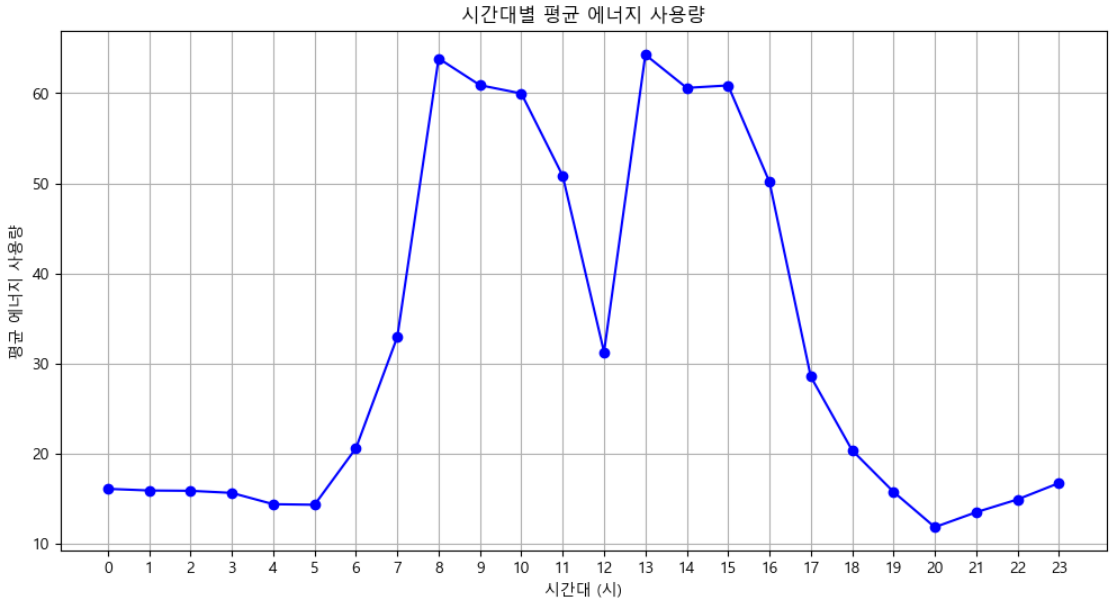
In [21]:
```
# 시간대 및 요일 특성 생성
df['hour_of_day'] = df['be_date'].dt.hour
df['day_of_week'] = df['be_date'].dt.dayofweek



# 상자 수염 그림을 사용한 이상치 시각화
plt.figure(figsize=(8, 6))
df.boxplot(column=['be_ac_energy', 'be_light_energy', 'be_plug_energy', 'be_tota
plt.show()
```

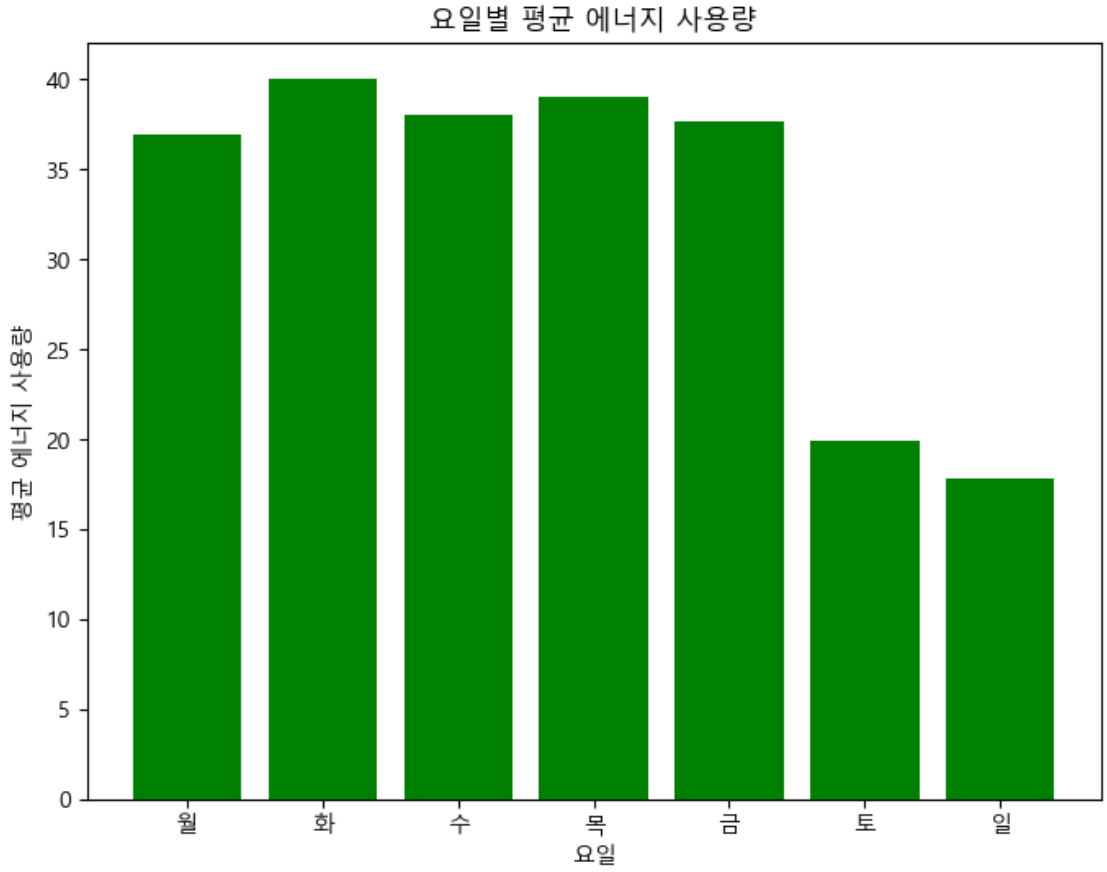


In [22]:
```
# # Z-점수를 이용한 이상치 제거
z_scores = stats.zscore(df[['be_ac_energy', 'be_light_energy', 'be_plug_energy',
abs_z_scores = np.abs(z_scores)
filtered_entries = (abs_z_scores < 3).all(axis=1)  # Z-점수가 3보다 작은 데이터
df_no_outliers = df[filtered_entries]

# 상자 수염 그림을 사용한 이상치 시각화
plt.figure(figsize=(8, 6))
df_no_outliers.boxplot(column=['be_ac_energy', 'be_light_energy', 'be_plug_energy
plt.show()
```



In [23]:
```
# 시간대별 평균 에너지 사용량 계산
hourly_energy = df_no_outliers.groupby('hour_of_day')['be_total_energy'].mean()

# 요일별 평균 에너지 사용량 계산
daily_energy = df_no_outliers.groupby('day_of_week')['be_total_energy'].mean()

# 시간대별 에너지 사용량 시각화
plt.figure(figsize=(12, 6))
plt.plot(hourly_energy.index, hourly_energy.values, marker='o', linestyle='-', co
plt.xlabel('시간대 (시)')
plt.ylabel('평균 에너지 사용량')
plt.title('시간대별 평균 에너지 사용량')
plt.xticks(range(24))  # x축 레이블을 0부터 23까지 표시
plt.grid(True)
plt.show()

# 요일별 에너지 사용량 시각화
days = ['월', '화', '수', '목', '금', '토', '일']
plt.figure(figsize=(8, 6))
plt.bar(days, daily_energy.values, color='g')
plt.xlabel('요일')
plt.ylabel('평균 에너지 사용량')
plt.title('요일별 평균 에너지 사용량')
plt.show()
```



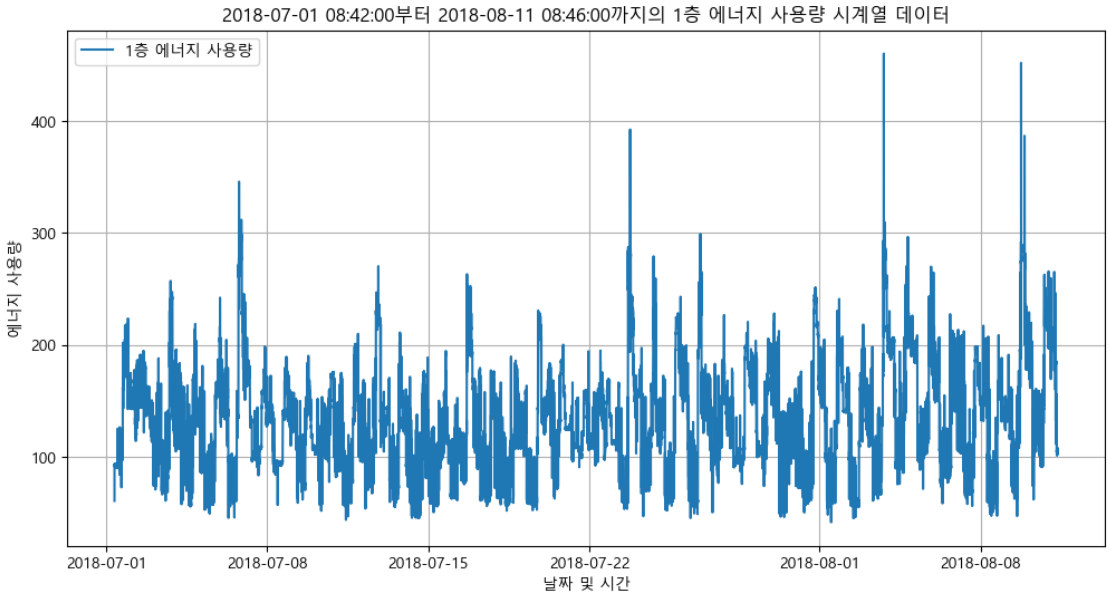


In [31]:
```
print(df_no_outliers.head())
```

	be_date	be_ac_energy	be_light_energy	be_plug_energy	W
522	2018-07-01 08:42:00	23.13	18.15	19.34	
523	2018-07-01 08:43:00	46.54	27.00	19.52	
524	2018-07-01 08:44:00	46.61	27.23	19.57	
525	2018-07-01 08:45:00	46.60	27.25	19.43	
526	2018-07-01 08:46:00	46.47	27.23	18.96	

	be_total_energy	be_floor	hour_of_day	day_of_week
522	60.62	1	8	6
523	93.06	1	8	6
524	93.41	1	8	6
525	93.28	1	8	6
526	92.66	1	8	6

In [46]:
```
# 특정기간 데이터 선택
start_time = '2018-07-01 08:42:00'
end_time = '2018-08-11 08:46:00'
filtered_data = filtered_data.loc[start_time:end_time]
filtered_floor_1_data = df[(df['be_floor'] == 1) & (df.index >= start_time) & (d

# 1층의 특정 기간 에너지 사용량 시각화
plt.figure(figsize=(12, 6))
plt.plot(filtered_floor_1_data.index, filtered_floor_1_data['be_total_energy'], l
plt.xlabel('날짜 및 시간')
plt.ylabel('에너지 사용량')
plt.title(f'{start_time}부터 {end_time}까지의 1층 에너지 사용량 시계열 데이터')
plt.legend()
plt.grid(True)
plt.show()
```



In [59]:
```
# 특정 기간 동안 건물 전체의 에너지 사용량 데이터 선택
filtered_building_data = df[(df.index >= start_time) & (df.index <= end_time)]

# 각 층의 에너지 사용량을 더한 후 건물 전체의 에너지 사용량 계산
total_energy_by_time = filtered_building_data.groupby(filtered_building_data.inde

# 건물 전체의 에너지 사용량 시각화
plt.figure(figsize=(12, 6))
plt.plot(total_energy_by_time.index, total_energy_by_time.values, color='b', labe
plt.xlabel('날짜 및 시간')
plt.ylabel('에너지 사용량')
plt.title(f'{start_time}부터 {end_time}까지의 건물 에너지 사용량 시계열 데이터')
plt.legend()
plt.grid(True)
plt.show()
```