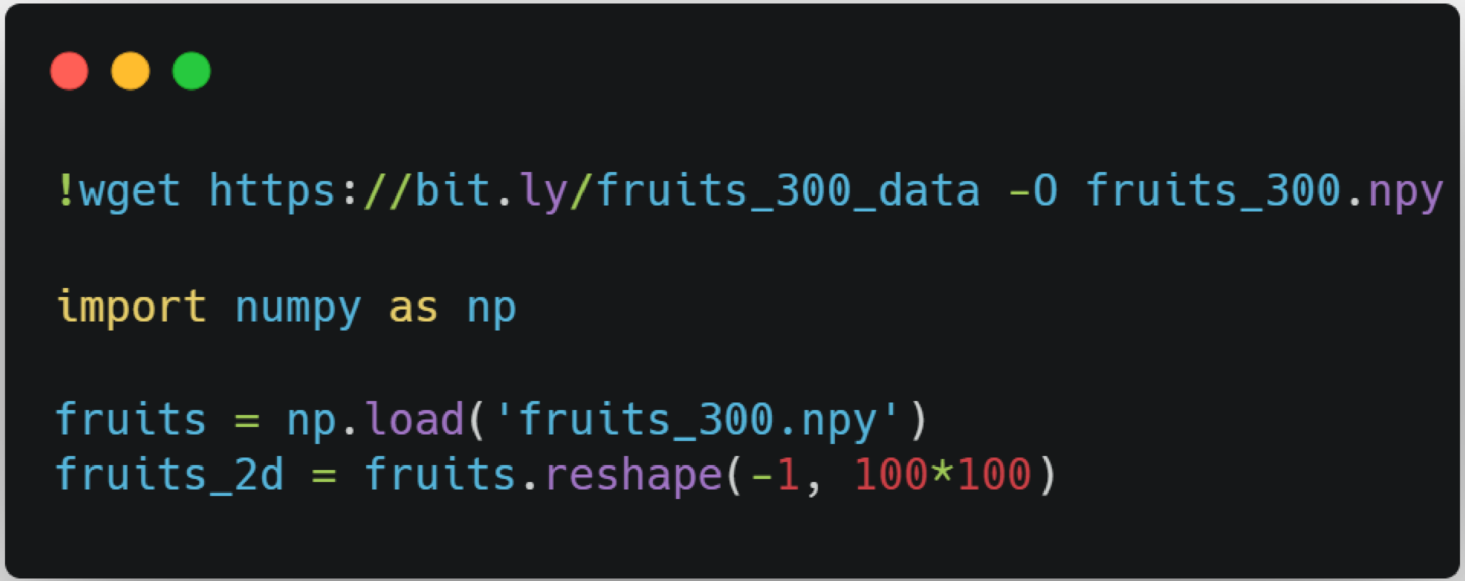
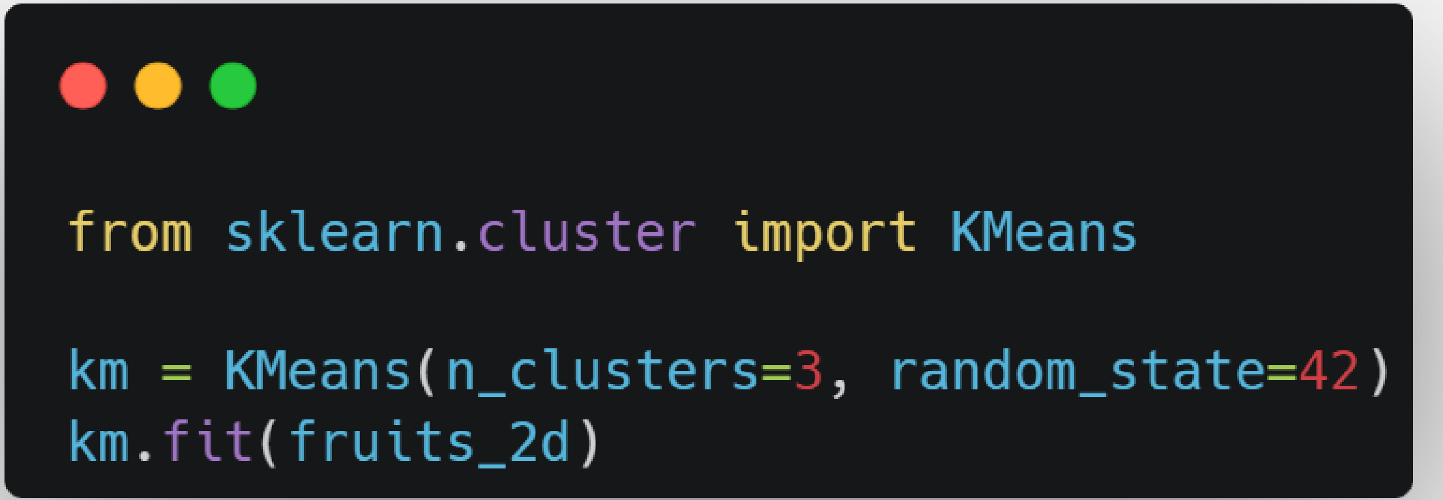
데이터 다운로드 및 불러오기



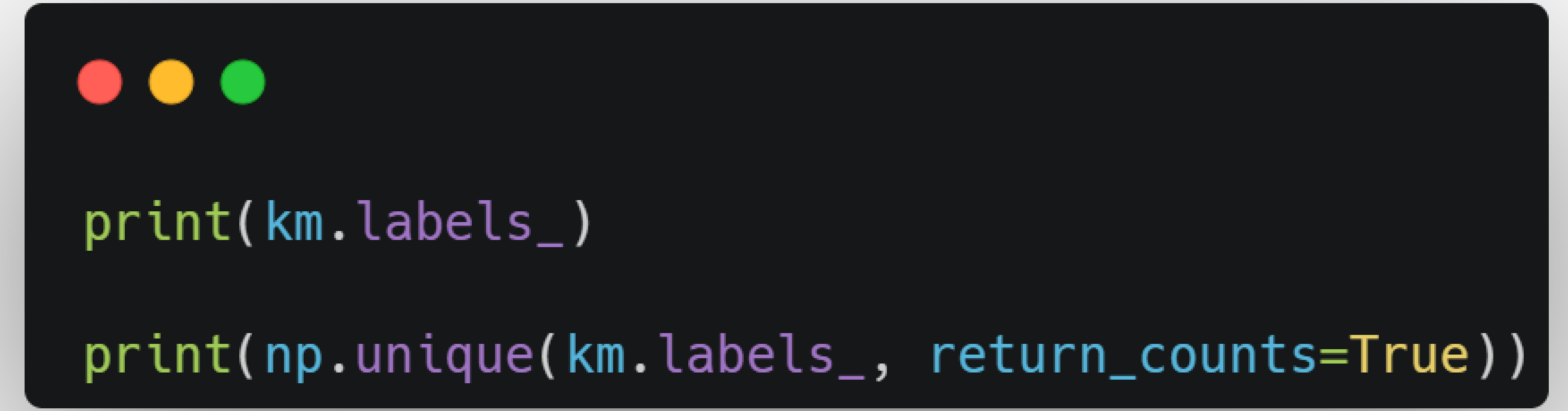
```
!wget https://bit.ly/fruits_300_data -O fruits_300.npy

import numpy as np

fruits = np.load('fruits_300.npy')
fruits_2d = fruits.reshape(-1, 100*100)
```

K-평균 기능 불러오기



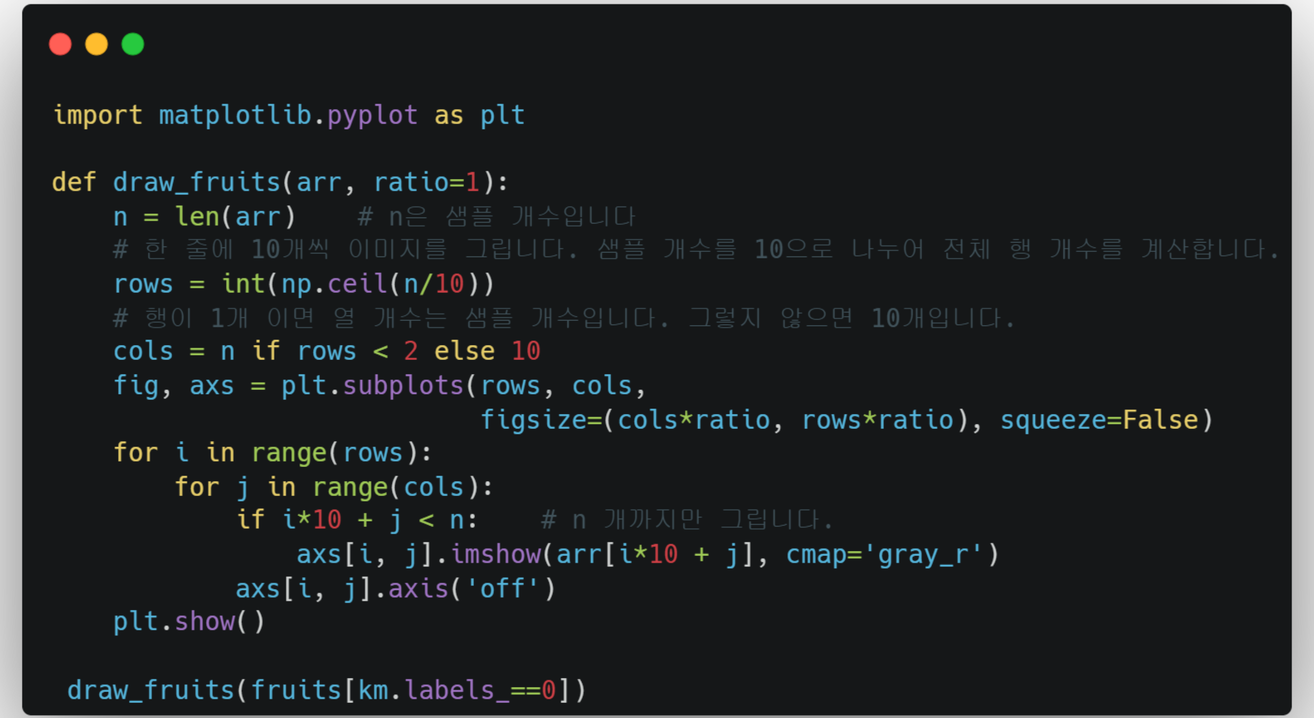
```
from sklearn.cluster import KMeans

km = KMeans(n_clusters=3, random_state=42)
km.fit(fruits_2d)
```

K-평균 클러스터 확인



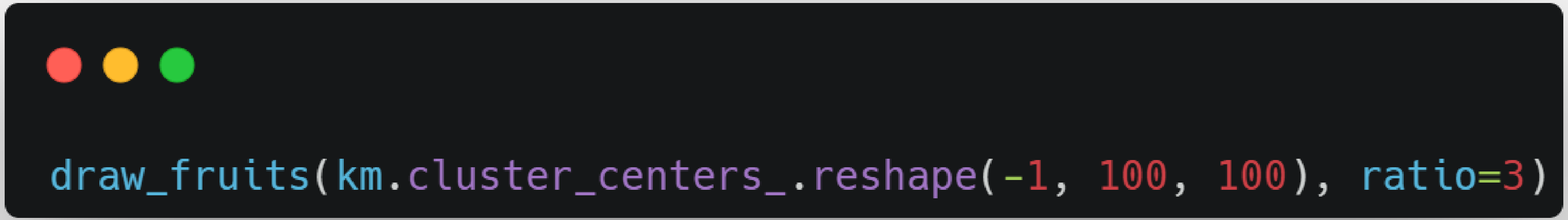
```python
print(km.labels_)

print(np.unique(km.labels_, return_counts=True))
```

군집으로 형성된 각 클러스터별 확인



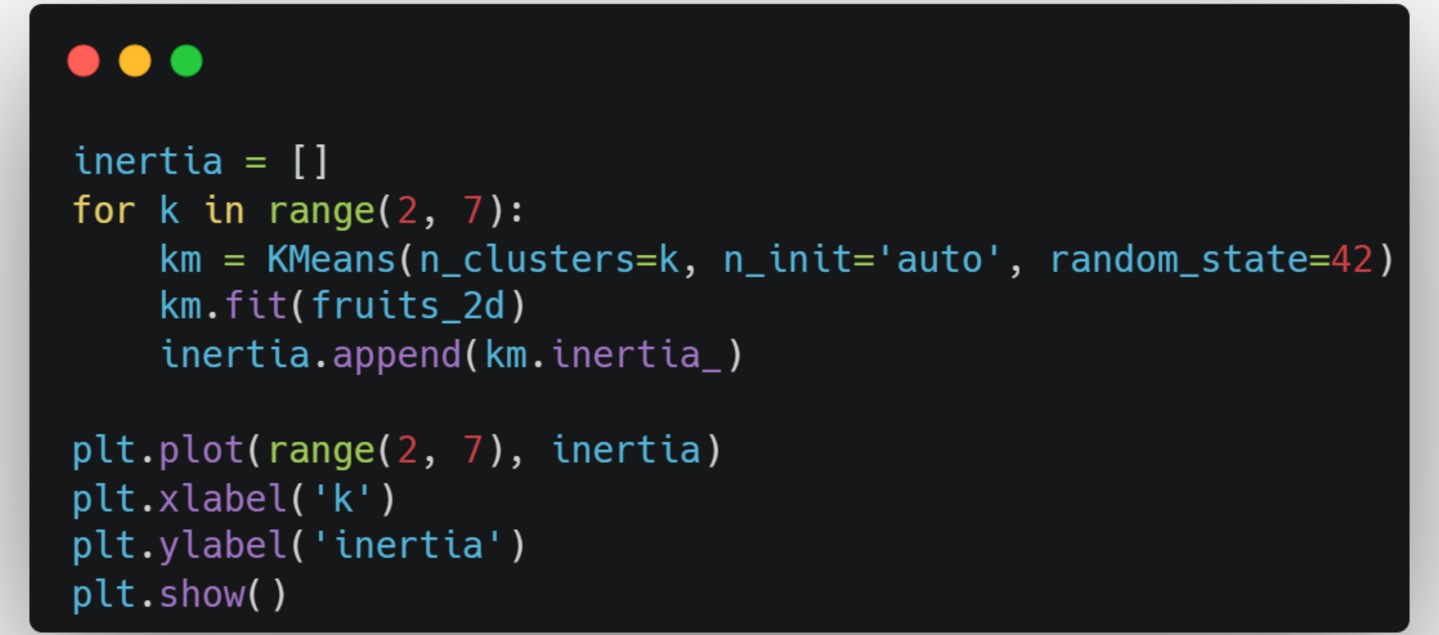
```python
import matplotlib.pyplot as plt

def draw_fruits(arr, ratio=1):
    n = len(arr)    # n은 샘플 개수입니다
    # 한 줄에 10개씩 이미지를 그립니다. 샘플 개수를 10으로 나누어 전체 행 개수를 계산합니다.
    rows = int(np.ceil(n/10))
    # 행이 1개 이면 열 개수는 샘플 개수입니다. 그렇지 않으면 10개입니다.
    cols = n if rows < 2 else 10
    fig, axs = plt.subplots(rows, cols,
                            figsize=(cols*ratio, rows*ratio), squeeze=False)
    for i in range(rows):
        for j in range(cols):
            if i*10 + j < n:    # n 개까지만 그립니다.
                axs[i, j].imshow(arr[i*10 + j], cmap='gray_r')
            axs[i, j].axis('off')
    plt.show()

 draw_fruits(fruits[km.labels_==0])
```

K-평균 클러스터 중심 확인(시각화)

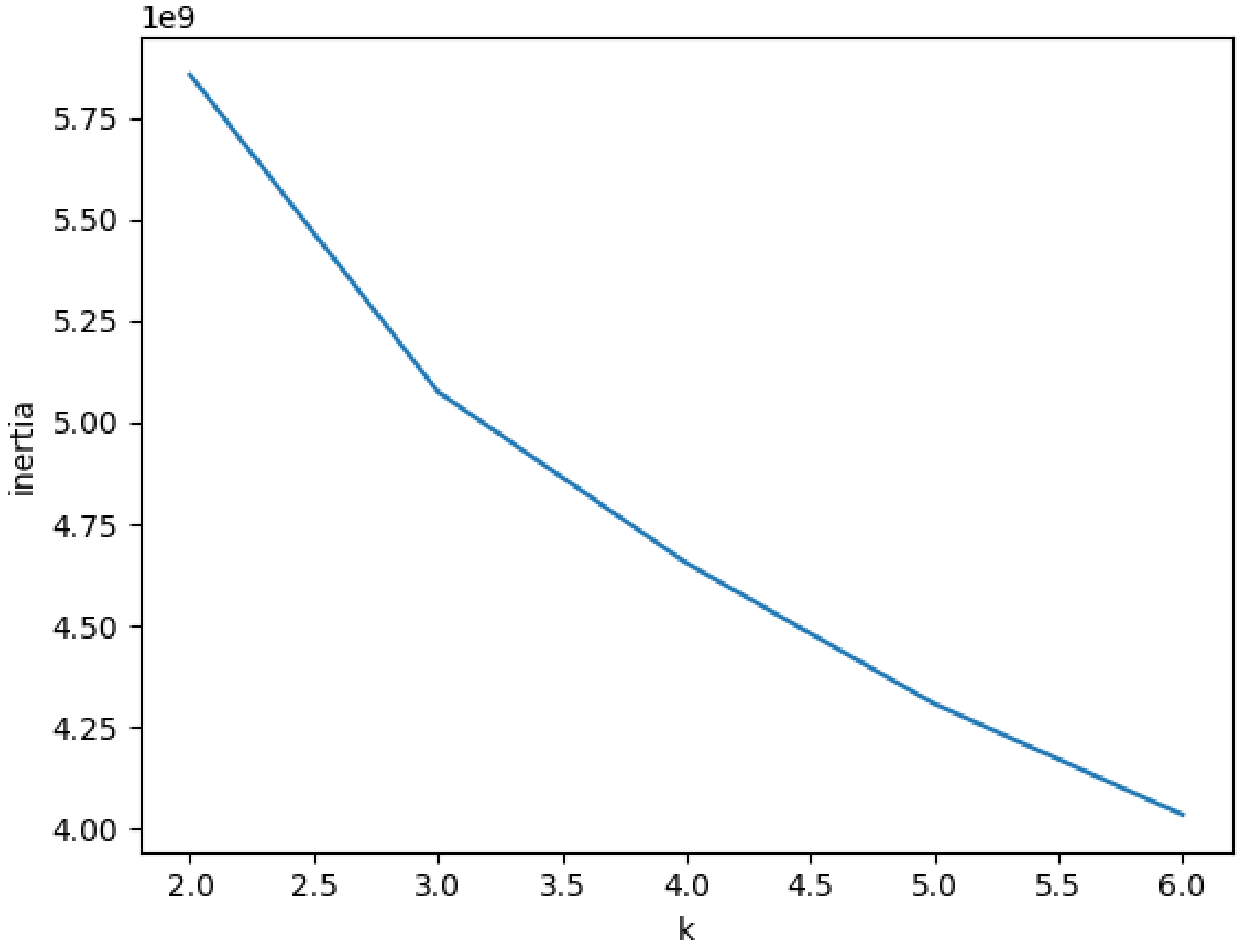


```python
draw_fruits(km.cluster_centers_.reshape(-1, 100, 100), ratio=3)
```

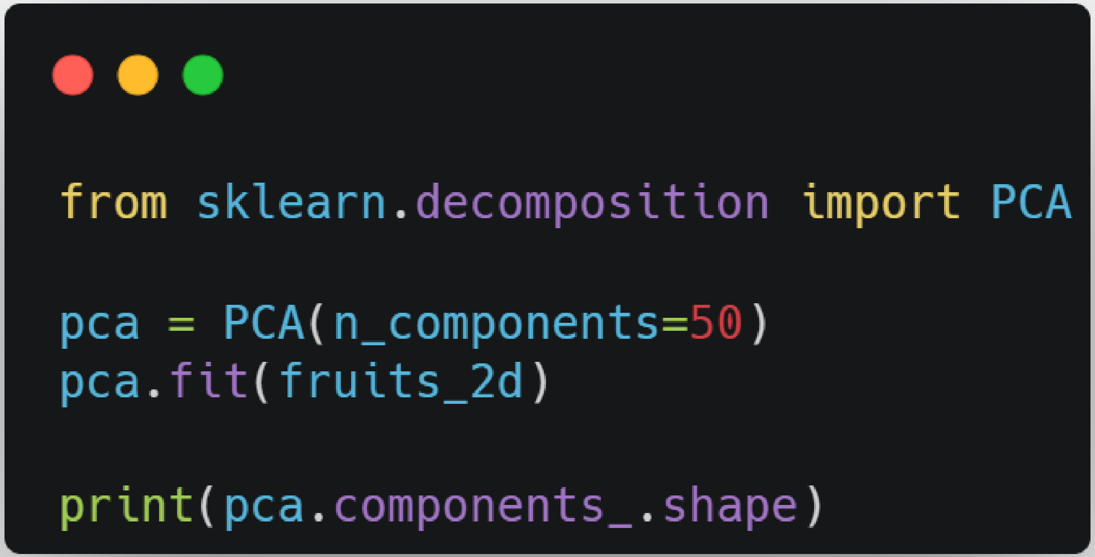
최적의 K 찾기



```python
inertia = []
for k in range(2, 7):
    km = KMeans(n_clusters=k, n_init='auto', random_state=42)
    km.fit(fruits_2d)
    inertia.append(km.inertia_)

plt.plot(range(2, 7), inertia)
plt.xlabel('k')
plt.ylabel('inertia')
plt.show()
```

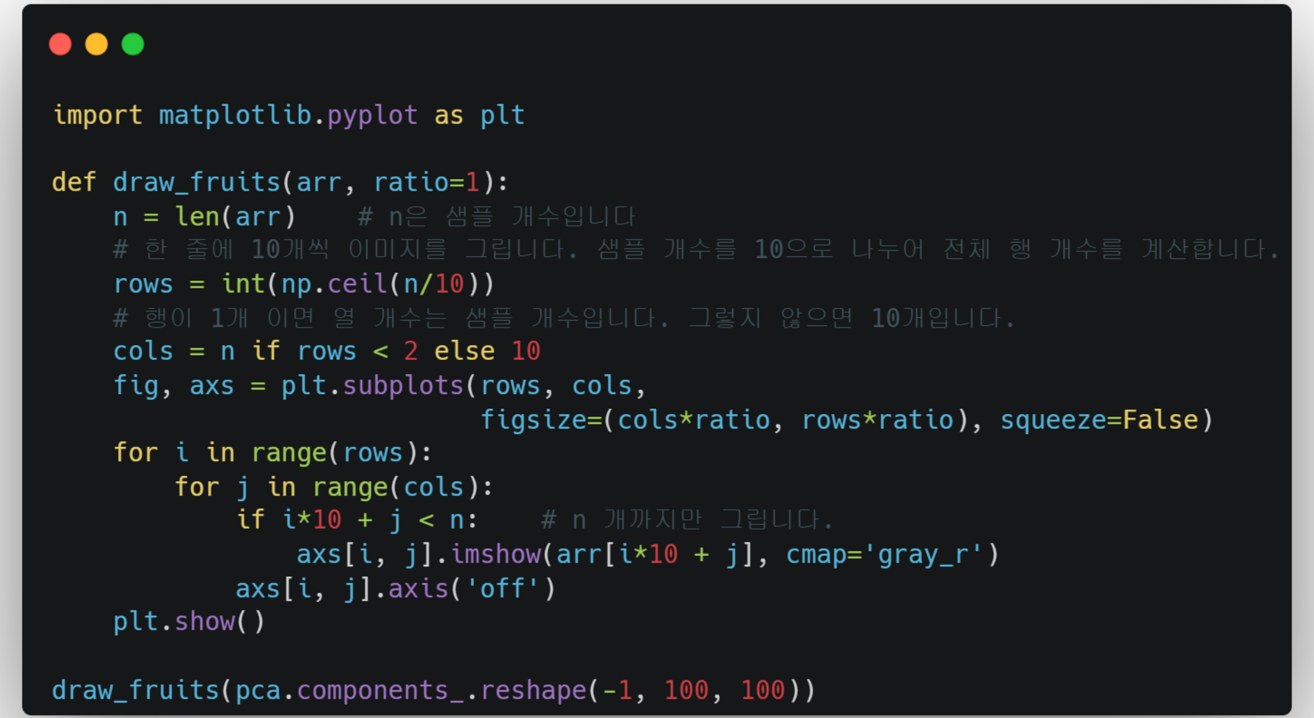

1e9
5.75
5.50
5.25
5.00
4.75
4.50
4.25
4.00
inertia
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
k

PCA 구동 및 차원 확인



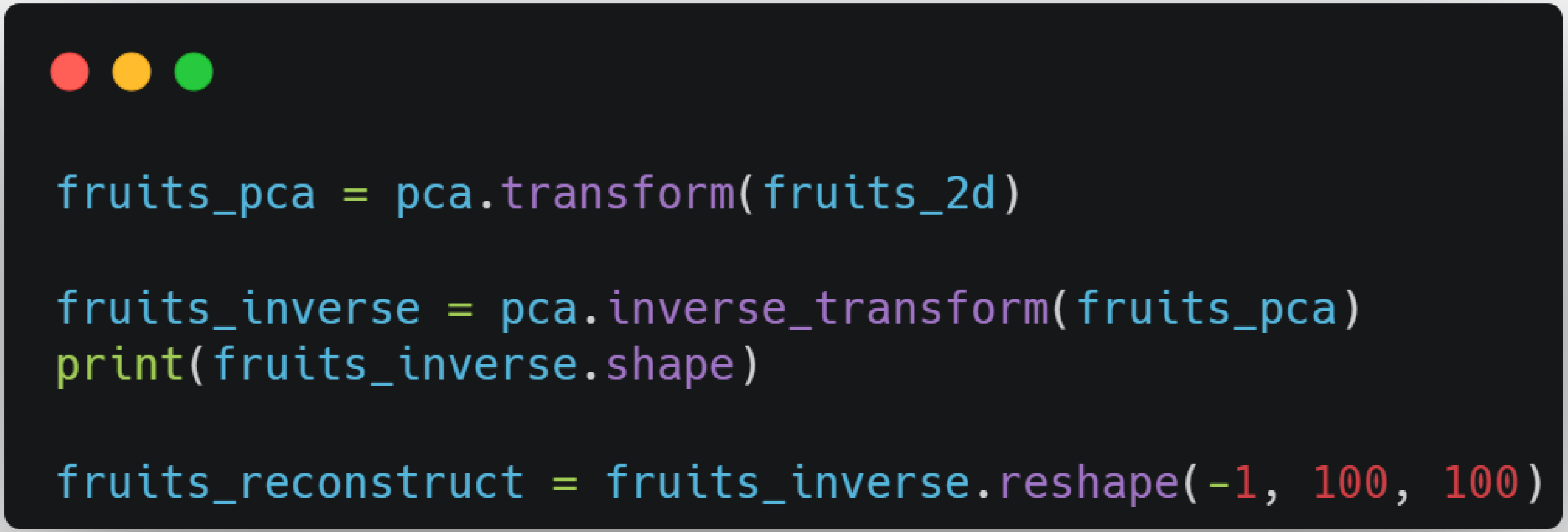
```python
from sklearn.decomposition import PCA

pca = PCA(n_components=50)
pca.fit(fruits_2d)

print(pca.components_.shape)
```

주성분 분석 결과 이미지 출력



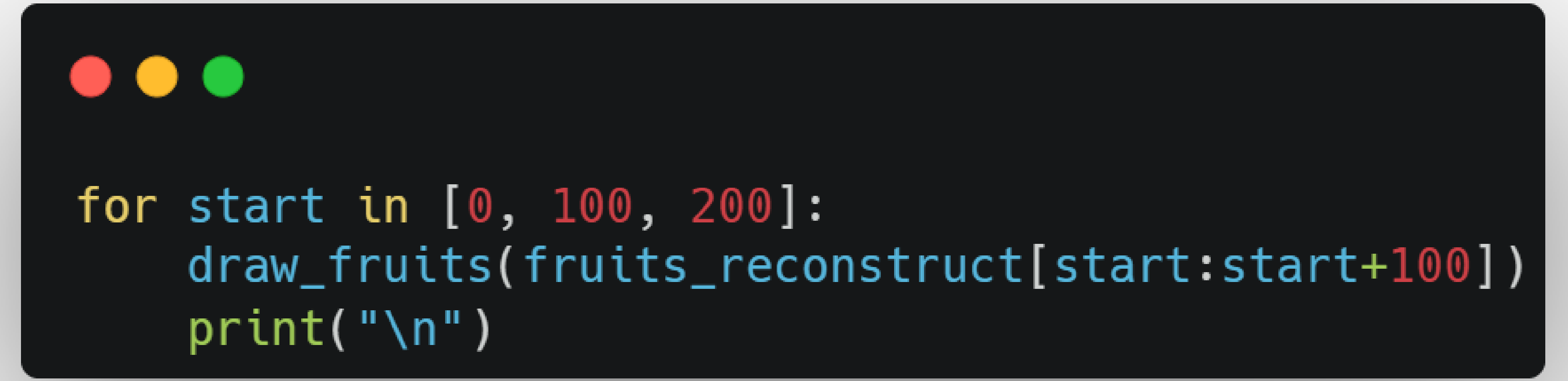
```python
import matplotlib.pyplot as plt

def draw_fruits(arr, ratio=1):
    n = len(arr)    # n은 샘플 개수입니다
    # 한 줄에 10개씩 이미지를 그립니다. 샘플 개수를 10으로 나누어 전체 행 개수를 계산합니다.
    rows = int(np.ceil(n/10))
    # 행이 1개 이면 열 개수는 샘플 개수입니다. 그렇지 않으면 10개입니다.
    cols = n if rows < 2 else 10
    fig, axs = plt.subplots(rows, cols,
                            figsize=(cols*ratio, rows*ratio), squeeze=False)
    for i in range(rows):
        for j in range(cols):
            if i*10 + j < n:    # n 개까지만 그립니다.
                axs[i, j].imshow(arr[i*10 + j], cmap='gray_r')
            axs[i, j].axis('off')
    plt.show()

draw_fruits(pca.components_.reshape(-1, 100, 100))
```

원본 이미지 -> PCA 변환 값

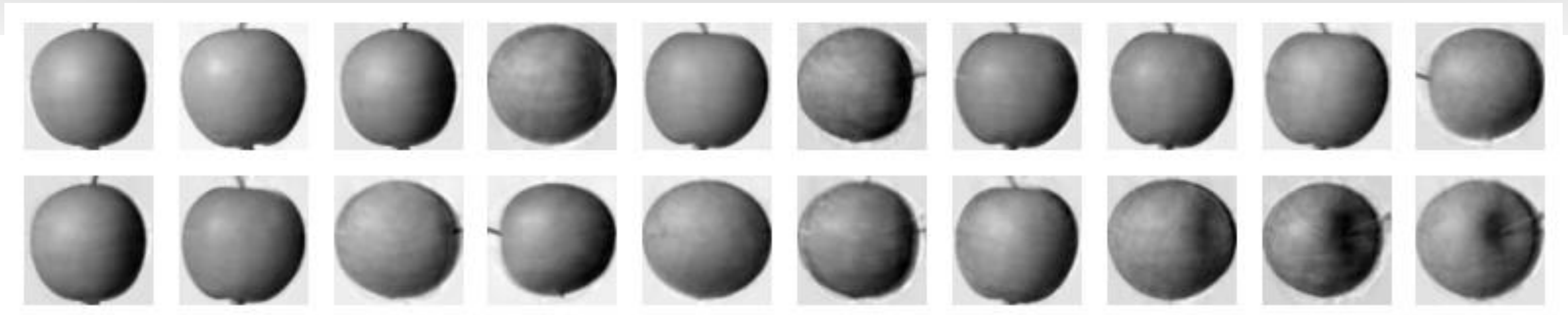


```
fruits_pca = pca.transform(fruits_2d)

fruits_inverse = pca.inverse_transform(fruits_pca)
print(fruits_inverse.shape)

fruits_reconstruct = fruits_inverse.reshape(-1, 100, 100)
```

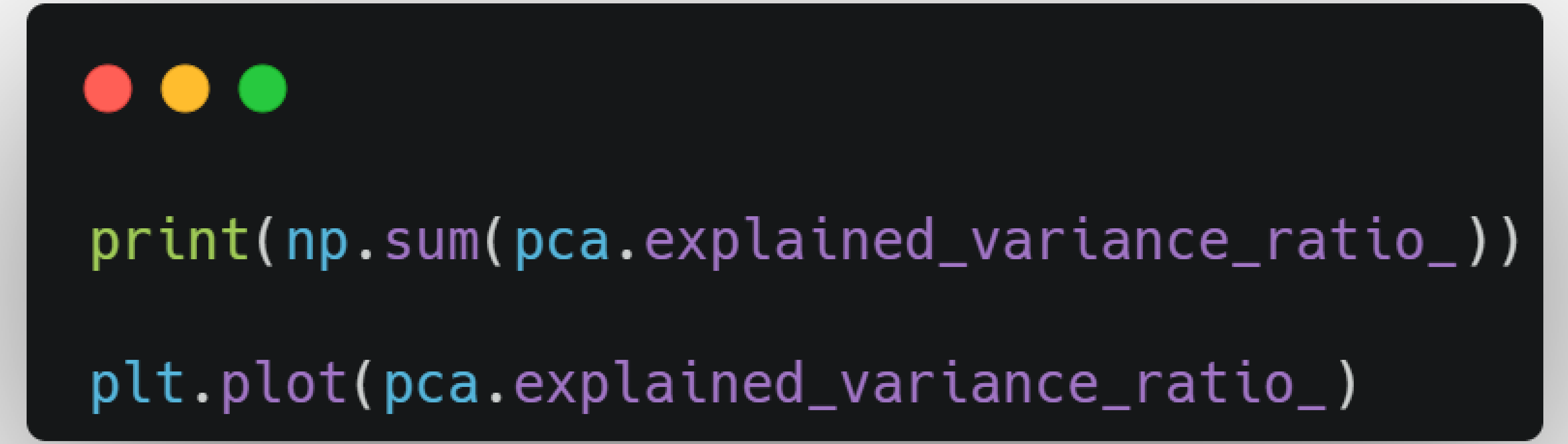
이미지 출력



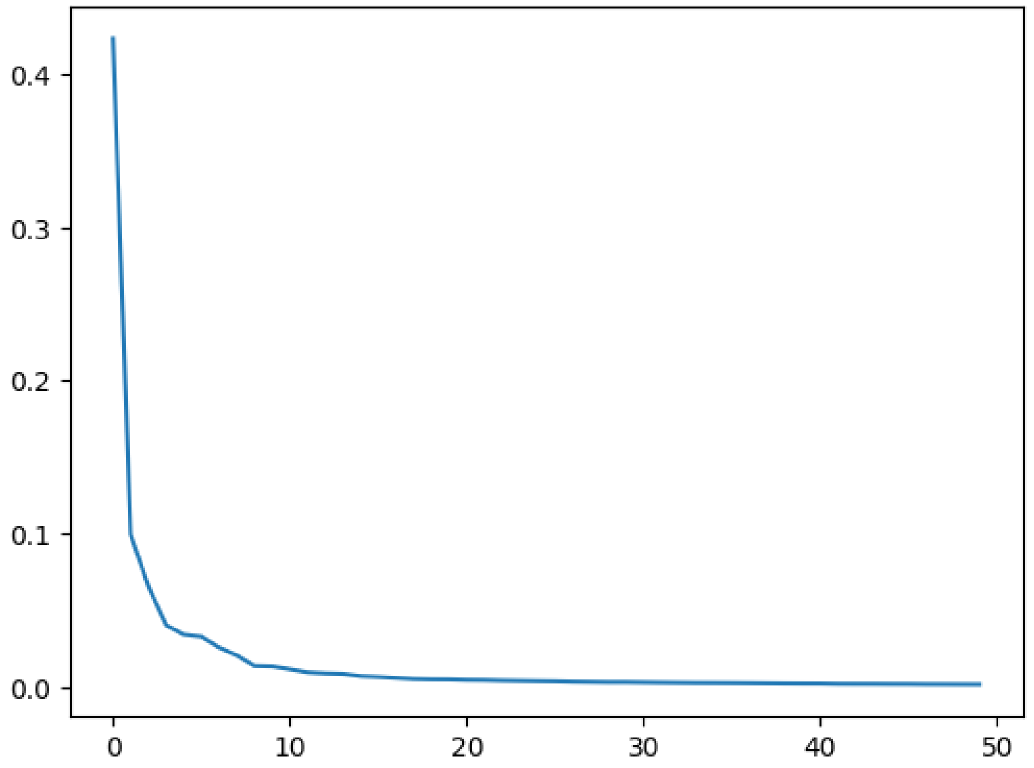
```
for start in [0, 100, 200]:
    draw_fruits(fruits_reconstruct[start:start+100])
    print("\n")
```



PCA 분산 확인 및 그래프 확인



```
print(np.sum(pca.explained_variance_ratio_))

plt.plot(pca.explained_variance_ratio_)
```



0.4
0.3
0.2
0.1
0.0
0
10
20
30
40
50