Iris 데이터셋 k-means clustering 예제

라이브러리 및 패키지 가져오기

In [1]:
```
# 필요한 패키지 설치
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn import datasets

from sklearn.cluster import KMeans
```

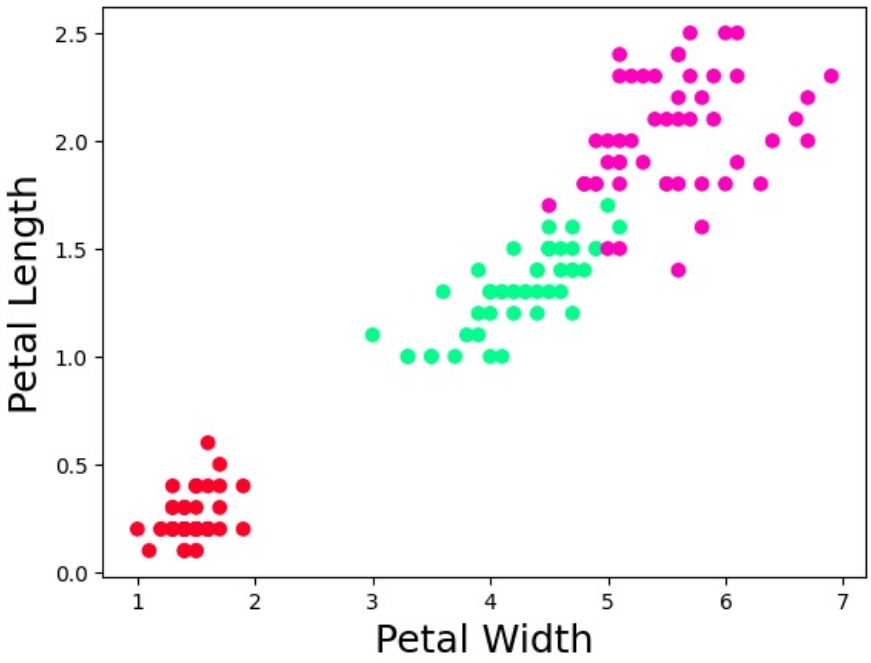
```
C:\Users\user\AppData\Local\Temp\ipykernel_30208\3645585242.py:2: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better interoperability with other lib
raries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd
```

데이터 불러오기

In [2]:
```
iris = datasets.load_iris()

X = iris.data[:, 2:]          # 꽃잎(petal))의 길이(length)와 너비(width) 데이터만 추출
y = iris.target
```

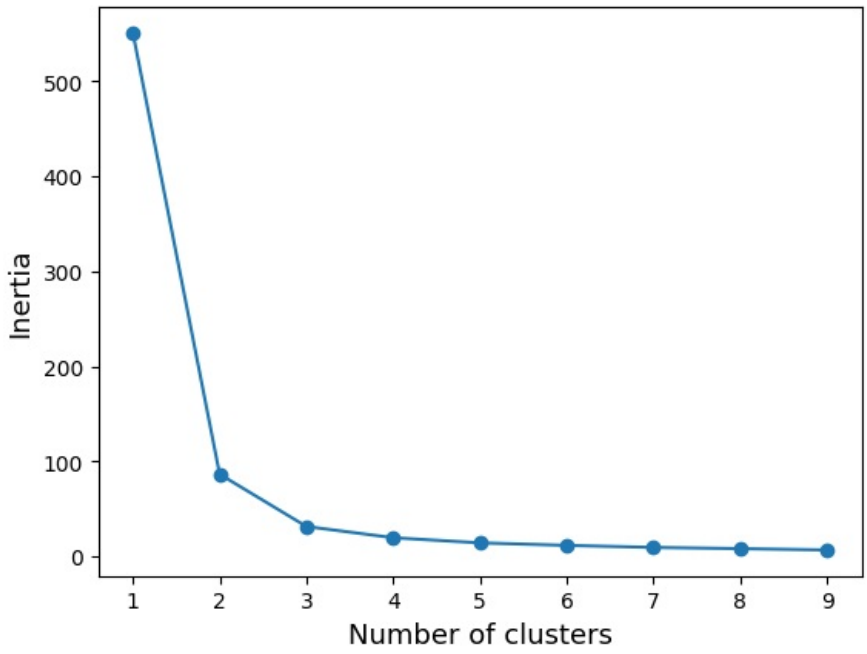
In [3]:
```
# 데이터 산점도로 시각화
plt.scatter(X[:,0], X[:,1], c = y, cmap ='gist_rainbow')      # 산점도 그래프 그리기
plt.xlabel('Petal Width', fontsize = 18)                      # x축 이름 설정
plt.ylabel('Petal Length', fontsize = 18)                     # y축 이름 설정
```

Out[3]:
```
Text(0, 0.5, 'Petal Length')
```



최적의 k값 찾기

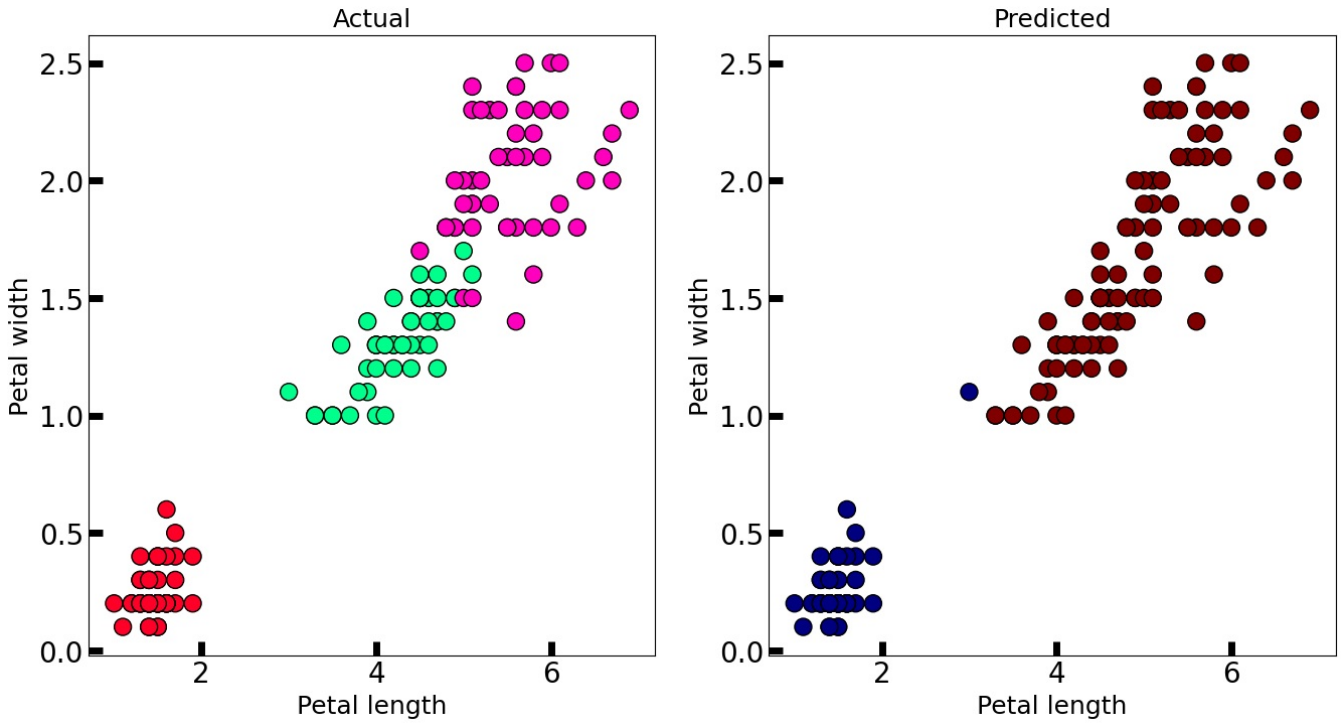
In [4]:
```
inertia_arr = []
k_range = range(1, 10)

for k in k_range:
    kmeans = KMeans(n_clusters= k, random_state= 21)
    kmeans.fit(X)
    inertia = kmeans.inertia_
    inertia_arr.append(inertia)

# Elobw Method 그래프 그리기
plt.plot(k_range, inertia_arr, marker= 'o')
plt.xlabel('Number of clusters', fontsize=13)
plt.ylabel('Inertia', fontsize=13)
plt.show()
```



K-means clustering 적용(k=2 일때)

In [5]:
```
kmeans = KMeans(n_clusters = 2, random_state=21)
kmeans.fit(X)

y_pred = kmeans.labels_

# 비교 그래프 그리기
fig, axes = plt.subplots(1, 2, figsize = (16,8))

# Iris 꽃잎 그래프
axes[0].scatter(X[:, 0], X[:, 1], c = y, cmap = 'gist_rainbow', edgecolor = 'k', s = 150)
axes[0].set_xlabel('Petal length', fontsize = 18)
axes[0].set_ylabel('Petal width', fontsize = 18)
axes[0].tick_params(direction = 'in', length = 10, width = 5, colors = 'k', labelsize = 20)
axes[0].set_title('Actual', fontsize = 18)

# K-Means Clustering을 통해 예측한 결과 그래프
axes[1].scatter(X[:, 0], X[:, 1], c = y_pred, cmap = 'jet', edgecolor = 'k', s = 150)
axes[1].set_xlabel('Petal length', fontsize = 18)
axes[1].set_ylabel('Petal width', fontsize = 18)
axes[1].tick_params(direction = 'in', length = 10, width = 5, colors = 'k', labelsize = 20)
axes[1].set_title('Predicted', fontsize = 18)
```

Out[5]:
```
Text(0.5, 1.0, 'Predicted')
```



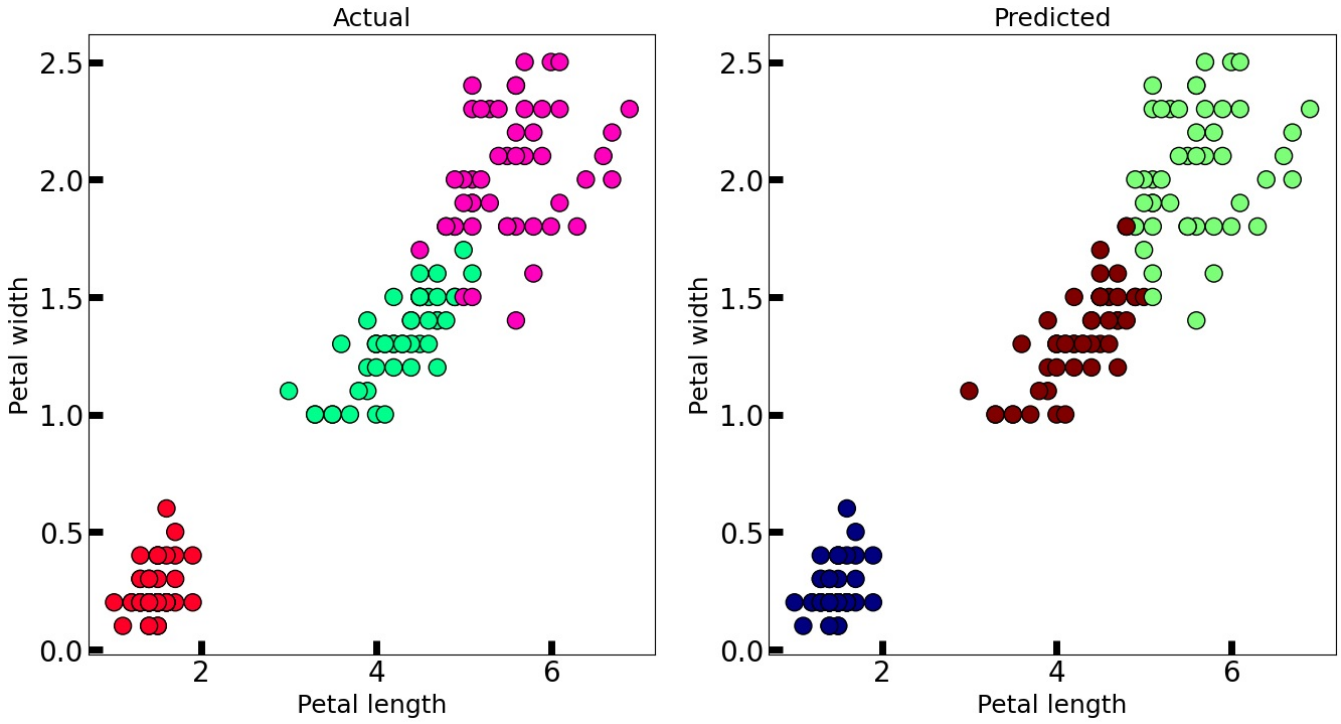
K-means clustering 적용(k=3 일때)

In [6]:
```
kmeans = KMeans(n_clusters = 3, random_state=21)
```

```
kmeans.fit(X)

y_pred = kmeans.labels_

# 비교 그래프 그리기
fig, axes = plt.subplots(1, 2, figsize = (16,8))

# Iris 꽃잎 그래프
axes[0].scatter(X[:, 0], X[:, 1], c = y, cmap = 'gist_rainbow', edgecolor = 'k', s = 150)
axes[0].set_xlabel('Petal length', fontsize = 18)
axes[0].set_ylabel('Petal width', fontsize = 18)
axes[0].tick_params(direction = 'in', length = 10, width = 5, colors = 'k', labelsize = 20)
axes[0].set_title('Actual', fontsize = 18)

# K-Means Clustering을 통해 예측한 결과 그래프
axes[1].scatter(X[:, 0], X[:, 1], c = y_pred, cmap = 'jet', edgecolor = 'k', s = 150)
axes[1].set_xlabel('Petal length', fontsize = 18)
axes[1].set_ylabel('Petal width', fontsize = 18)
axes[1].tick_params(direction = 'in', length = 10, width = 5, colors = 'k', labelsize = 20)
axes[1].set_title('Predicted', fontsize = 18)
```

Out[6]:

```
Text(0.5, 1.0, 'Predicted')
```



In [ ]: