In [9]:
```
# Step 1: Import Required Libraries
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report, confusion_matrix
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [10]:
```
# Step 2: Load the Dataset
iris = load_iris()
X = iris.data
y = iris.target
```

In [11]:
```
# Step 3: Split into Training and Test Set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

In [12]:
```
# Step 4: Train the Classification Model
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)
```

Out[12]:

▾ DecisionTreeClassifier

DecisionTreeClassifier()

In [13]:
```
# Step 5: Make Predictions
y_pred = clf.predict(X_test)
```

In [14]:
```
# Step 6: Evaluate the Model
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

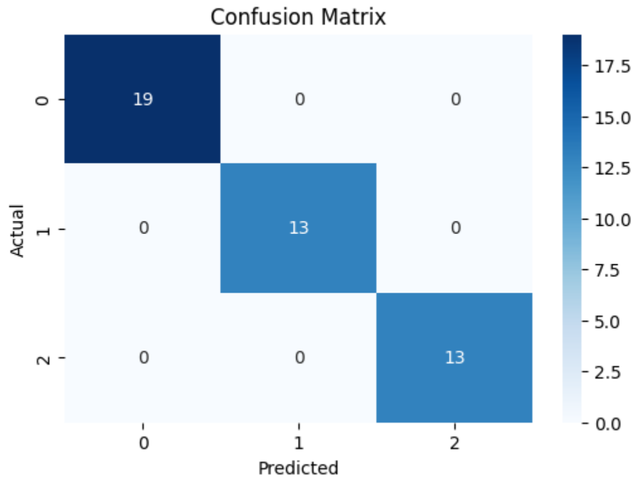
```
Confusion Matrix:
 [[19  0  0]
 [ 0 13  0]
 [ 0  0 13]]

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        19
           1       1.00      1.00      1.00        13
           2       1.00      1.00      1.00        13

    accuracy                           1.00        45
   macro avg       1.00      1.00      1.00        45
weighted avg       1.00      1.00      1.00        45
```

In [15]:
```
# Step 7: Visualize Confusion Matrix
plt.figure(figsize=(6,4))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, cmap='Blues', fmt='d')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```


Confusion Matrix
Actual
Predicted
0
1
2
19
0
0
0
13
0
0
0
13
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0

In [ ]: