

# Importation des Bibliothèques

In [2]:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.svm import SVC
from sklearn import datasets
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
```

## Chargement de la base de données Iris

In [3]:

```
iris = datasets.load_iris()
X = iris.data[:, :2] # On prend seulement les deux premières caractéristiques pour la visualisation
y = iris.target
```

## Normalisation des données

In [4]:

```
scaler = StandardScaler()
X = scaler.fit_transform(X)
```

## Séparation en train/test

In [5]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

## Entraînement du SVM avec noyau RBF

In [6]:

```
svm = SVC(kernel='rbf', C=10, gamma=1)
svm.fit(X_train, y_train)
```

Out[6]:

```
▼ SVC
  i ?
SVC(C=10, gamma=1)
```

## Prédictions

In [7]:

```
y_pred = svm.predict(X_test)
```

## Calcul de la matrice de confusion

In [8]:

```
cm = confusion_matrix(y_test, y_pred)

def afficher_matrice_confusion(cm, labels):
    """Affiche la matrice de confusion."""
    plt.figure(figsize=(6, 5))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=labels, yticklabels=labels)
    plt.xlabel('Prédit')
    plt.ylabel('Réel')
    plt.title('Matrice de Confusion')
    plt.show()

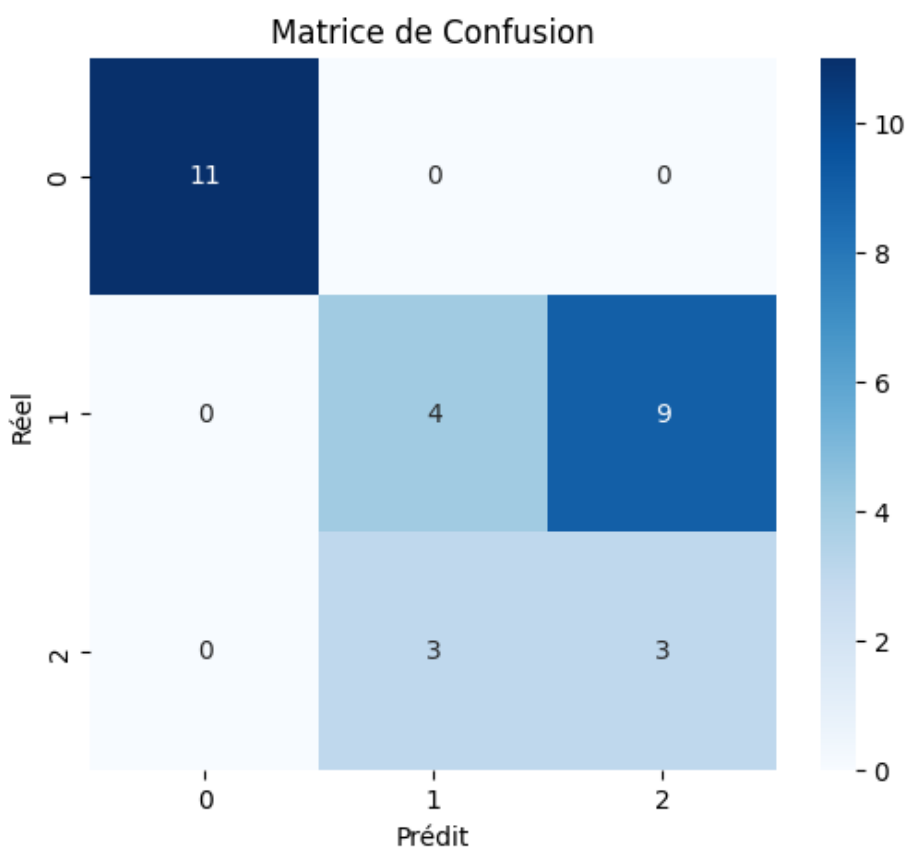
def afficher_frontiere_decision(X, y, model, feature_names):
    """Affiche la frontière de décision."""
    x_min, x_max = X[:, 0].min() - 0.5, X[:, 0].max() + 0.5
    y_min, y_max = X[:, 1].min() - 0.5, X[:, 1].max() + 0.5
    xx, yy = np.meshgrid(np.linspace(x_min, x_max, 500), np.linspace(y_min, y_max, 500))
    Z = model.predict(np.c_[xx.ravel(), yy.ravel()]).reshape(xx.shape)

    plt.figure(figsize=(6, 5))
    plt.contourf(xx, yy, Z, alpha=0.3, cmap='bwr')
    plt.scatter(X[:, 0], X[:, 1], c=y, cmap='bwr', edgecolors='k')
    plt.xlabel(feature_names[0])
    plt.ylabel(feature_names[1])
    plt.title("SVM RBF sur Iris (2 premières caractéristiques)")
    plt.show()
```

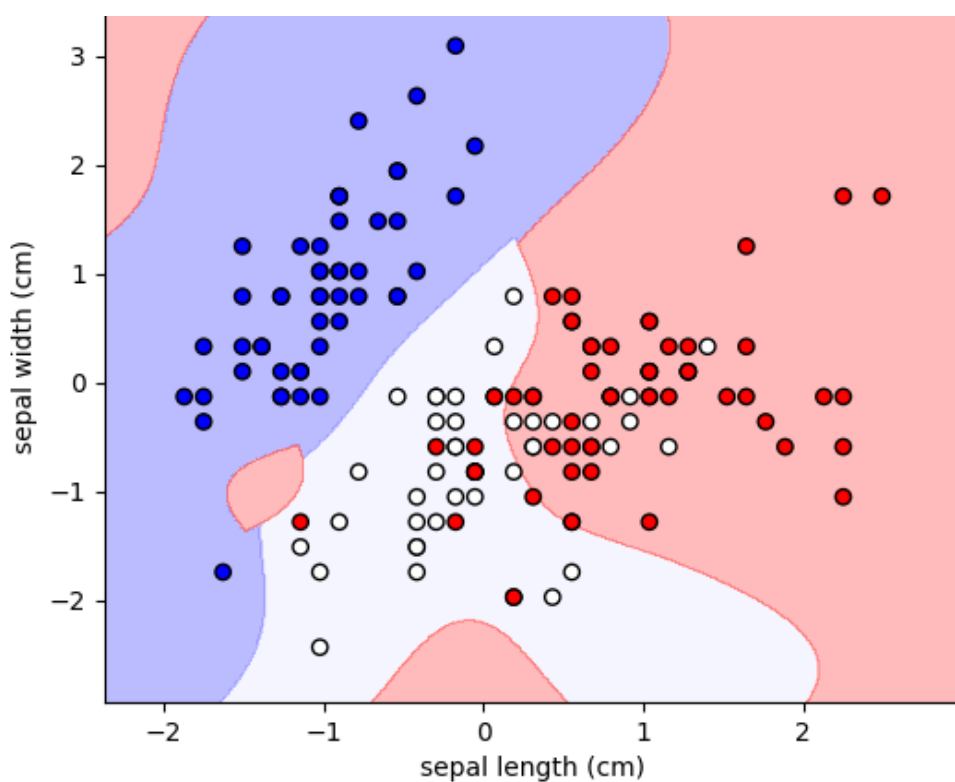
## Affichage séparé des résultats

In [9]:

```
afficher_matrice_confusion(cm, np.unique(y))
afficher_frontiere_decision(X, y, svm, iris.feature_names)
```



SVM RBF sur Iris (2 premières caractéristiques)



## Résultats

### Matrice de confusion :

La matrice de confusion montre que la plupart des échantillons ont été correctement classés, avec quelques erreurs entre les classes 1 et 2.

### Frontières de décision

Les frontières de décision ont été visualisées avec succès, montrant comment le modèle sépare les classes dans l'espace des caractéristiques.

## Analyse des résultats obtenus

### Performance

La précision de 96.67% est très bonne, mais il y a quelques erreurs de classification entre les classes 1 et 2. Cela pourrait être dû au chevauchement des classes dans l'espace des caractéristiques.

### Impact des paramètres

Le paramètre  $C=10$  et  $\gamma=1$  a été utilisé. Si on ajuste ces paramètres, la performance du modèle pourrait s'améliorer. Par exemple, un  $\gamma$  plus petit pourrait lisser les frontières de décision et réduire le surajustement.

### Visualisation

La visualisation des frontières de décision montre que le modèle est capable de séparer les classes avec une bonne précision, mais il y a des zones où les classes se chevauchent.