In [14]:

```python
# Dataset Loading

from sklearn.datasets import load_iris

iris = load_iris()

# Feature: The variable of data
# Response: Ouptut of the variable
# Dataset:the collection of data

X = iris.data
y = iris.target

feature_names = iris.feature_names
target_names = iris.target_names

print("Feature Names: ", feature_names)
print("Target Names: ", target_names)
print("\n First 10 rows of X: \n", X[:10])
```

```
Feature Names:  ['sepal length (cm)', 'sepal width (cm)', 'petal length (c
m)', 'petal width (cm)']
Target Names:  ['setosa' 'versicolor' 'virginica']

 First 10 rows of X:
 [[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5.  3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]]
```

In [15]:

```python
# Splitting the Dataset: used for checking the accuracy
#                        of the model

from sklearn.datasets import load_iris

iris = load_iris()

X = iris.data
y = iris.target

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=1
)

print(X_train.shape)
print(X_test.shape)

print(y_train.shape)
print(y_test.shape)
```

```
(105, 4)
(45, 4)
(105,)
(45,)
```

In [20]:

```
# Train the Model

# This method used KNN (K Nearest Neighbors)

from sklearn.datasets import load_iris

iris = load_iris()

X = iris.data
y = iris.target

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=1
)

from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics

classifier_knn = KNeighborsClassifier(n_neighbors=3)
classifier_knn.fit(X_train, y_train)

y_predict = classifier_knn.predict(X_test)

print("Accuracy: ", metrics.accuracy_score(y_test, y_predict))

sample = [[5, 5, 3, 2], [2, 4, 3, 5]]

preds = classifier_knn.predict(sample)

for p in preds:
    print("Predictions: ", iris.target_names[p])
```

```
Accuracy:  0.9777777777777777
Predictions:  versicolor
Predictions:  virginica
```

In [26]:

```
# Binarisation: process convert numerical value to boolean value

import numpy as np
from sklearn import preprocessing

input_data = np.array(
    [[2.1, -1.9, 5.5],
    [-1.5, 2.4, 3.5],
    [0.5, -7.9, 5.6],
    [5.9, 2.3, -5.8]]
)

# threshold=0.5, if > 0.5 = 1, if < 0.5 = 0
data_binarized = preprocessing.Binarizer(threshold=0.5).transform(input_da
print("\Binarized Data: \n", data_binarized)
```

```
\Binarized Data:
 [[1. 0. 1.]
 [0. 1. 1.]
 [0. 0. 1.]
 [1. 1. 0.]]
```

In [27]:

```
# Data as table

import seaborn as sns

iris = sns.load_dataset('iris')
iris.head()
```

Out[27]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [28]:

```
# Data as Target Array

import seaborn as sns

iris = sns.load_dataset('iris')

%matplotlib inline

import seaborn as sns; sns.set()
sns.pairplot(iris, hue='species', height=3)
```

Out[28]:

```
<seaborn.axisgrid.PairGrid at 0x7ffb8112d3c8>
```

```
In [ ]:
```