In [1]:

```
import pandas as pd
from sklearn.datasets import load_iris
iris = load_iris()
```

In [2]:

```
iris.feature_names
```

Out[2]:

```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

In [3]:

```
iris.target_names
```

Out[3]:

```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

In [4]:

```
df = pd.DataFrame(iris.data,columns=iris.feature_names)
df.head()
```

Out[4]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

In [5]:

```
df['target'] = iris.target
df.head()
```

Out[5]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

In [7]:
```
df[df.target==1].head()
```

Out[7]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
50	7.0	3.2	4.7	1.4	1
51	6.4	3.2	4.5	1.5	1
52	6.9	3.1	4.9	1.5	1
53	5.5	2.3	4.0	1.3	1
54	6.5	2.8	4.6	1.5	1

In [8]:
```
df[df.target==2].head()
```

Out[8]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
100	6.3	3.3	6.0	2.5	2
101	5.8	2.7	5.1	1.9	2
102	7.1	3.0	5.9	2.1	2
103	6.3	2.9	5.6	1.8	2
104	6.5	3.0	5.8	2.2	2

In [9]:
```
df['flower_name'] =df.target.apply(lambda x: iris.target_names[x])
df.head()
```

Out[9]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	flower_name
0	5.1	3.5	1.4	0.2	0	setosa
1	4.9	3.0	1.4	0.2	0	setosa
2	4.7	3.2	1.3	0.2	0	setosa
3	4.6	3.1	1.5	0.2	0	setosa
4	5.0	3.6	1.4	0.2	0	setosa

In [10]:
```
df[45:55]
```

Out[10]:

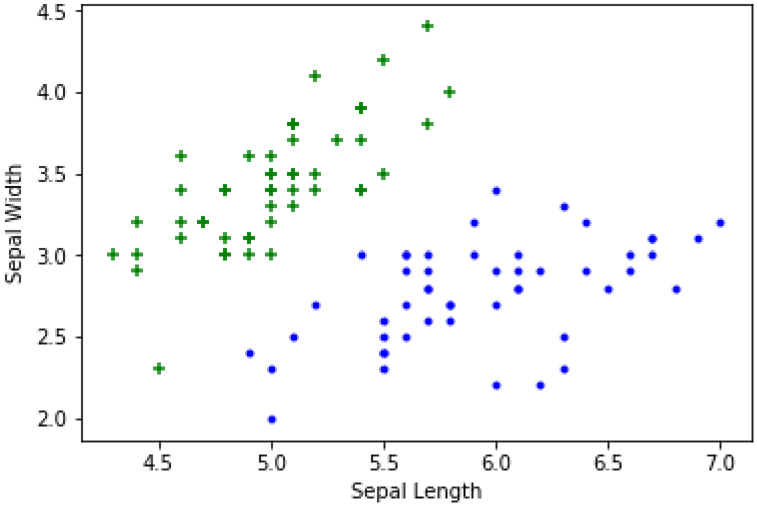
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	flower_name
45	4.8	3.0	1.4	0.3	0	setosa
46	5.1	3.8	1.6	0.2	0	setosa
47	4.6	3.2	1.4	0.2	0	setosa
48	5.3	3.7	1.5	0.2	0	setosa
49	5.0	3.3	1.4	0.2	0	setosa
50	7.0	3.2	4.7	1.4	1	versicolor
51	6.4	3.2	4.5	1.5	1	versicolor
52	6.9	3.1	4.9	1.5	1	versicolor
53	5.5	2.3	4.0	1.3	1	versicolor
54	6.5	2.8	4.6	1.5	1	versicolor

In [11]:
```
df0 = df[:50]
df1 = df[50:100]
df2 = df[100:]
```

In [12]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

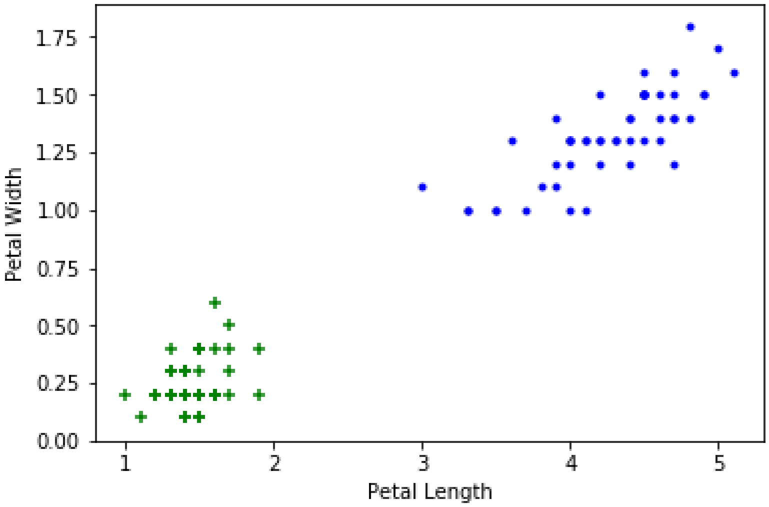
In [13]:
```
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.scatter(df0['sepal length (cm)'], df0['sepal width (cm)'],color="green",marke
plt.scatter(df1['sepal length (cm)'], df1['sepal width (cm)'],color="blue",marke
```

Out[13]:
```
<matplotlib.collections.PathCollection at 0x186ffb0b4e0>
```



In [14]:
```
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
plt.scatter(df0['petal length (cm)'], df0['petal width (cm)'],color="green",marke
plt.scatter(df1['petal length (cm)'], df1['petal width (cm)'],color="blue",marker
```

Out[14]:
```
<matplotlib.collections.PathCollection at 0x186ffb9df98>
```



In [15]:
```
from sklearn.model_selection import train_test_split
```

In [16]:
```
X = df.drop(['target','flower_name'], axis='columns')
y = df.target
```

In [17]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

In [18]:
```
len(X_train)
```

Out[18]:
```
120
```

In [19]:
```
len(X_test)
```

Out[19]:
```
30
```

In [20]:
```
from sklearn.svm import SVC
model = SVC()
```

In [21]:

```
model.fit(X_train, y_train)
```

```
C:\Users\aziz\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: FutureWarnin
g: The default value of gamma will change from 'auto' to 'scale' in version 0.2
2 to account better for unscaled features. Set gamma explicitly to 'auto' or 's
cale' to avoid this warning.
  "avoid this warning.", FutureWarning)
```

Out[21]:

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
  kernel='rbf', max_iter=-1, probability=False, random_state=None,
  shrinking=True, tol=0.001, verbose=False)
```

In [22]:

```
model.score(X_test, y_test)
```

Out[22]:

```
1.0
```

In [23]:

```
model.predict([[4.8,3.0,1.5,0.3]])
```

Out[23]:

```
array([0])
```

In [24]:

```
model_C = SVC(C=1)
model_C.fit(X_train, y_train)
model_C.score(X_test, y_test)
```

```
C:\Users\aziz\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: FutureWarnin
g: The default value of gamma will change from 'auto' to 'scale' in version 0.2
2 to account better for unscaled features. Set gamma explicitly to 'auto' or 's
cale' to avoid this warning.
  "avoid this warning.", FutureWarning)
```

Out[24]:

```
1.0
```

In [26]:

```
model_g = SVC(gamma=10)
model_g.fit(X_train, y_train)
model_g.score(X_test, y_test)
```

Out[26]:

```
0.9666666666666667
```

In [27]:

```
model_linear_kernal = SVC(kernel='linear')
model_linear_kernal.fit(X_train, y_train)
```

Out[27]:

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
  kernel='linear', max_iter=-1, probability=False, random_state=None,
  shrinking=True, tol=0.001, verbose=False)
```

In [29]:

```
model_linear_kernal.score(X_test, y_test)
```

Out[29]:

```
1.0
```