In [1]:
```python
import matplotlib.pyplot as plt
i = plt.imread('C:/Users/aiml/Desktop/iris.jpg')
plt.axis('off')
plt.title('IRIS')
plt.imshow(i)
plt.show()
```


In [2]:
```python
import pandas as pd
from sklearn.datasets import load_iris
iris = load_iris()
```

In [3]:
```python
iris.feature_names
```

Out[3]:
```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

In [4]:
```python
iris.target_names
```

Out[4]:
```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

In [5]:
```python
df = pd.DataFrame(iris.data,columns=iris.feature_names)
df.head()
```

Out[5]:

	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 [6]:
```python
df['target'] = iris.target
df.head()
```

Out[6]:

	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]:
```python
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]:
```python
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]:
```python
df['flower_name'] =df.target.apply(lambda x: iris.target_names[x])
df.head()
```
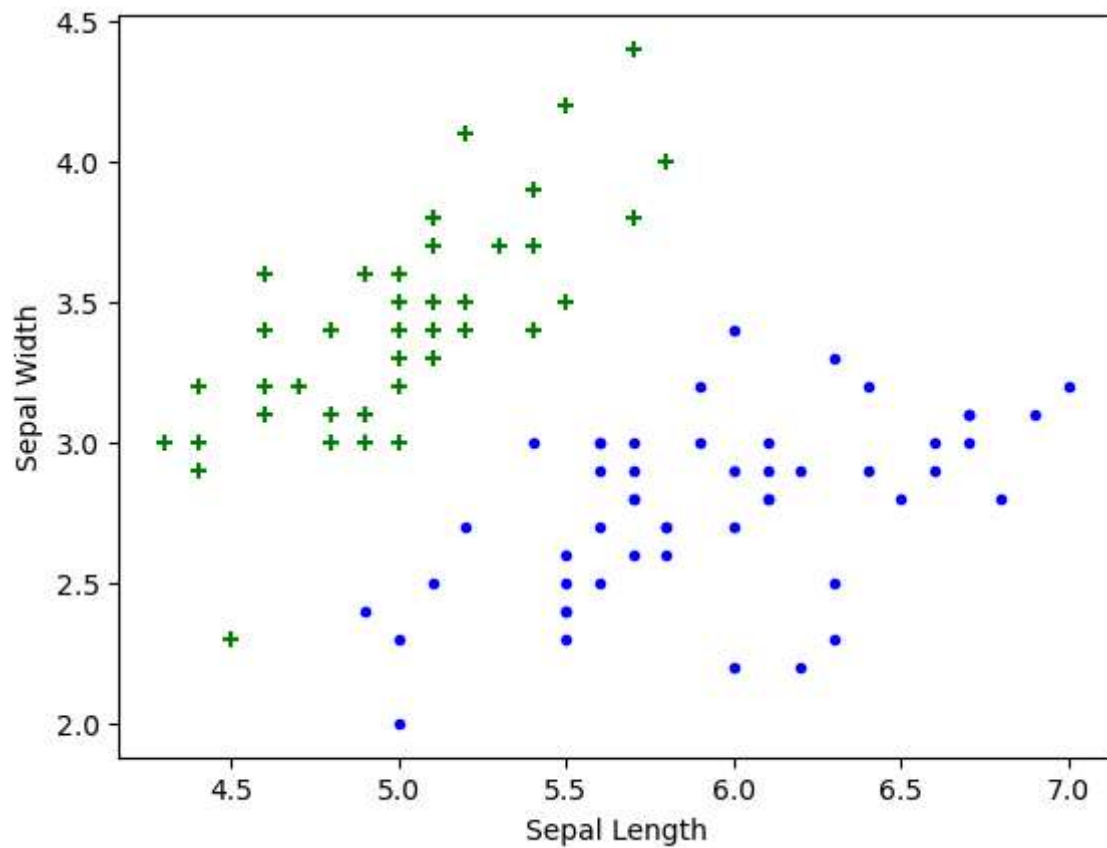
```
In [18]:  plt.xlabel('Petal Length')
          plt.ylabel('Petal Width')
          plt.scatter(df0['petal length (cm)'], df0['petal width (cm)'],color="green",marker='+'
          plt.scatter(df1['petal length (cm)'], df1['petal width (cm)'],color="blue",marker='.')
```

```
Out[18]:  <matplotlib.collections.PathCollection at 0x1583cd9e990>
```

In [31]: `# Print classification report for precesion, recall and f1-score for each classes`
```python
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        11
           1       1.00      0.92      0.96        13
           2       0.86      1.00      0.92         6

    accuracy                           0.97        30
   macro avg       0.95      0.97      0.96        30
weighted avg       0.97      0.97      0.97        30
```

In [ ]: