Loading the dataset

In [5]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [6]:
```
df=pd.read_csv("C:/Users/aparn/Downloads/Iris.csv")
```

Displaying the data

In [11]:
```
df.shape #no of rows and columns
```

Out[11]:
```
(150, 6)
```

In [14]:
```
df.head() #first 5 rows
```

Out[14]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [8]:
```
df.tail() # last 5 rows
```

Out[8]:

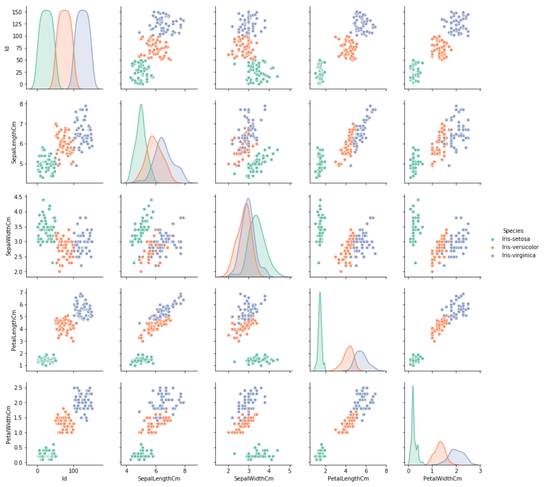
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

In [15]:
```
df.info()   # all non-null and numeric [except the labels]
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
Id               150 non-null int64
SepalLengthCm    150 non-null float64
SepalWidthCm     150 non-null float64
PetalLengthCm    150 non-null float64
PetalWidthCm     150 non-null float64
Species          150 non-null object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.1+ KB
```

Plotting the data

In [18]:
```
sns.pairplot(data=df,hue="Species")
plt.show()
```



Implemeting the K Means Clustering

In [25]:
```
from sklearn.cluster import KMeans
```

Elbow method - To find optimal number of custers

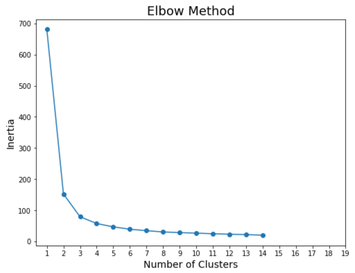
In [26]:
```
x=df.iloc[:,[0,1,2,3]].values
```

In [27]:
```
wcss = []

for k in range(1,15):
    kmeans = KMeans(n_clusters=k) #creating a kmeans instance with k clusters
    kmeans.fit(features) #fitting model to samples
    wcss.append(kmeans.inertia_) #appending the inertia to the list of wcss
```

Plot the inertia to find the optimal number of cluters

In [34]:
```
plt.figure(figsize=(8,6))
plt.title("Elbow Method", fontsize=18)
plt.plot(range(1,15),wcss,"-o")
plt.xlabel("Number of Clusters",fontsize=14)
plt.ylabel("Inertia",fontsize=14)
plt.xticks(range(1,20))
#plt.tight_layout()
plt.show()
```



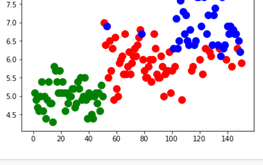
In [ ]:
```
# From the graph we can see that the elbow starts at 3. So number of clusters are 3
```

In [35]:
```
kmeans = KMeans(n_clusters=3)
kmeans_predict = kmeans.fit_predict(features)
```

In [36]:
```
df1 = pd.DataFrame({'labels':kmeans_predict,"Species":df['Species']})
ct1 = pd.crosstab(df1['labels'],df1['Species'])
```

Visualising the clusters

In [39]:
```
plt.scatter(x[kmeans_predict==0,0],x[kmeans_predict==0,1],s=100,c='red',label='Iris-setosa')
plt.scatter(x[kmeans_predict==1,0],x[kmeans_predict==1,1],s=100,c='green',label='Iris-versicolor')
plt.scatter(x[kmeans_predict==2,0],x[kmeans_predict==2,1],s=100,c='blue',label='Iris-virginica')
plt.show()
```



In [ ]:
```
```