In [4]:

```
from sklearn.cluster import KMeans
from matplotlib import pyplot as plt
import pandas as pd
from sklearn.datasets import load_iris
%matplotlib inline
ir=load_iris()
```

In [5]:

```
dir(ir)
```

Out[5]:

```
['DESCR',
 'data',
 'feature_names',
 'filename',
 'frame',
 'target',
 'target_names']
```

In [14]:

```
df=pd.DataFrame(ir.data,columns=ir.feature_names)
df.head()
df.drop(df[['sepal length (cm)','sepal width (cm)']],axis='columns',inplace=Tr
df.head()
```

Out[14]:

| | petal length (cm) | petal width (cm) |
|---|---|---|
| **0** | 1.4 | 0.2 |
| **1** | 1.4 | 0.2 |
| **2** | 1.3 | 0.2 |
| **3** | 1.5 | 0.2 |
| **4** | 1.4 | 0.2 |

In [21]:

```
k_rng=range(1,21)
sse=[]
for k in k_rng:
    km=KMeans(n_clusters=k)
    km.fit(df[['petal length (cm)','petal width (cm)']])
    sse.append(km.inertia_)
sse
```

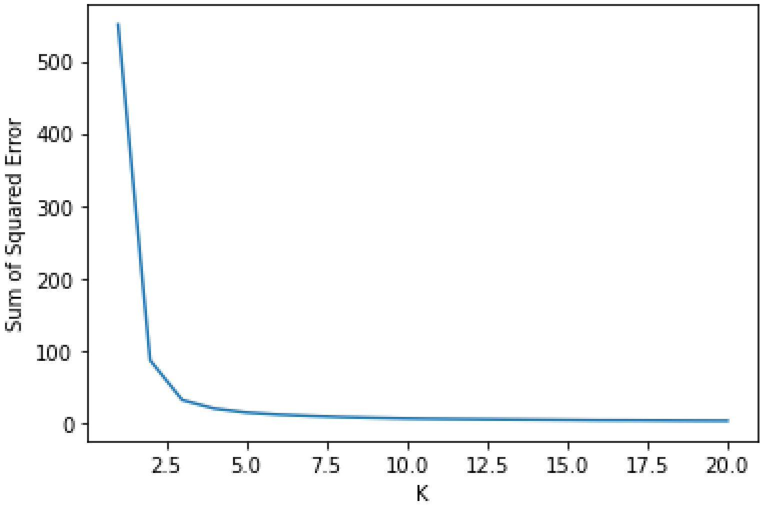
```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:881: Us
erWarning: KMeans is known to have a memory leak on Windows with MKL, when th
ere are less chunks than available threads. You can avoid it by setting the e
nvironment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

Out[21]:

```
[550.8953333333333,
 86.39021984551391,
 31.371358974358966,
 19.48300089968511,
 13.983213141025644,
 11.090892729819197,
 9.203314009661833,
 7.667019523446292,
 6.60300122100122,
 5.637756110418647,
 5.129500771158665,
 4.761637362637363,
 4.373977111639651,
 4.035798701298701,
 3.677049395049394,
 3.2400702183121535,
 3.176466061716062,
 2.926492049617049,
 2.6784797600060757,
 2.518312742812743]
```

In [22]:
```
plt.xlabel("K")
plt.ylabel("Sum of Squared Error")
plt.plot(k_rng,sse)
```

Out[22]:
```
[<matplotlib.lines.Line2D at 0x16b877e6a30>]
```



In [25]:
```
km=KMeans(n_clusters=3)
km.fit(df[['petal length (cm)','petal width (cm)']])
y_predicted=km.predict(df[['petal length (cm)','petal width (cm)']])
y_predicted
```

Out[25]:
```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

In [28]:
```
df['cluster']=y_predicted
df.head()
```

Out[28]:

| | petal length (cm) | petal width (cm) | cluster |
|---|---|---|---|
| **0** | 1.4 | 0.2 | 1 |
| **1** | 1.4 | 0.2 | 1 |
| **2** | 1.3 | 0.2 | 1 |
| **3** | 1.5 | 0.2 | 1 |
| **4** | 1.4 | 0.2 | 1 |

In [31]:

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaler.fit(df[['petal length (cm)']])
df[['petal length (cm)']]=scaler.transform(df[['petal length (cm)']])

scaler.fit(df['petal width (cm)'])
df[['petal width (cm)']]=scaler.transform(df[['petal width (cm)']])
df.head()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_14812/3692464787.py in <module>
      4 df[['petal length (cm)']]=scaler.transform(df[['petal length (cm)']])
      5 
----> 6 scaler.fit(df['petal width (cm)'])
      7 df[['petal width (cm)']]=scaler.transform(df[['petal width (cm)']])
      8 df.head()

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\preprocessing\_data.py in
fit(self, X, y)
    361         # Reset internal state before fitting
    362         self._reset()
--> 363         return self.partial_fit(X, y)
    364 
    365     def partial_fit(self, X, y=None):

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\preprocessing\_data.py in
partial_fit(self, X, y)
    394 
    395         first_pass = not hasattr(self, 'n_samples_seen_')
--> 396         X = self._validate_data(X, reset=first_pass,
    397                                 estimator=self, dtype=FLOAT_DTYPES,
    398                                 force_all_finite="allow-nan")

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py in _validate_data
(self, X, y, reset, validate_separately, **check_params)
    419             out = X
    420         elif isinstance(y, str) and y == 'no_validation':
--> 421             X = check_array(X, **check_params)
    422             out = X
    423         else:

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py in inn
er_f(*args, **kwargs)
     61             extra_args = len(args) - len(all_args)
     62             if extra_args <= 0:
---> 63                 return f(*args, **kwargs)
     64 
     65             # extra_args > 0

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py in che
ck_array(array, accept_sparse, accept_large_sparse, dtype, order, copy, force
_all_finite, ensure_2d, allow_nd, ensure_min_samples, ensure_min_features, es
timator)
    692             # If input is 1D raise error
    693             if array.ndim == 1:
--> 694                 raise ValueError(
    695                     "Expected 2D array, got 1D array instead:\narray=
{}.\n"
    696                     "Reshape your data either using array.reshape(-1,
1) if "

ValueError: Expected 2D array, got 1D array instead:
array=[0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 0.2 0.2 0.1 0.1 0.2 0.4 0.4 0.
3
 0.3 0.3 0.2 0.4 0.2 0.5 0.2 0.2 0.4 0.2 0.2 0.2 0.2 0.4 0.1 0.2 0.2 0.2
 0.2 0.1 0.2 0.2 0.3 0.3 0.2 0.6 0.4 0.3 0.2 0.2 0.2 0.2 1.4 1.5 1.5 1.3
 1.5 1.3 1.6 1.  1.3 1.4 1.  1.5 1.  1.4 1.3 1.4 1.5 1.  1.5 1.1 1.8 1.3
 1.5 1.2 1.3 1.4 1.4 1.7 1.5 1.  1.1 1.  1.2 1.6 1.5 1.6 1.5 1.3 1.3 1.3
 1.2 1.4 1.2 1.  1.3 1.2 1.3 1.3 1.1 1.3 2.5 1.9 2.1 1.8 2.2 2.1 1.7 1.8
 1.8 2.5 2.  1.9 2.1 2.  2.4 2.3 1.8 2.2 2.3 1.5 2.3 2.  2.  1.8 2.1 1.8
 1.8 1.8 2.1 1.6 1.9 2.  2.2 1.5 1.4 2.3 2.4 1.8 1.8 2.1 2.4 2.3 1.9 2.3
 2.5 2.3 1.9 2.  2.3 1.8].
Reshape your data either using array.reshape(-1, 1) if your data has a single
feature or array.reshape(1, -1) if it contains a single sample.
```

In [35]:

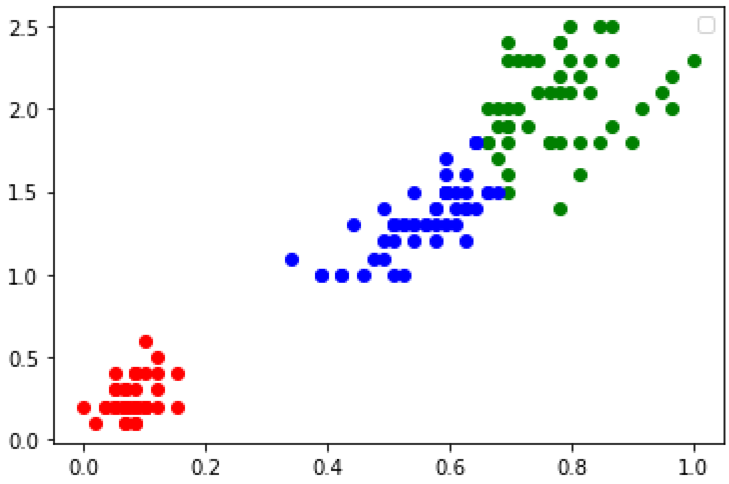
```
%matplotlib inline
df1=df[df.cluster==0]
df2=df[df.cluster==1]
df3=df[df.cluster==2]

plt.scatter(df1['petal length (cm)'],df1['petal width (cm)'],color='green')
plt.scatter(df2['petal length (cm)'],df2['petal width (cm)'],color='red')
plt.scatter(df3['petal length (cm)'],df3['petal width (cm)'],color='blue')
plt.legend()
```

```
No handles with labels found to put in legend.
```

Out[35]:

```
<matplotlib.legend.Legend at 0x16b87aceb50>
```



In [ ]:

In [ ]: