In [38]:
```
import pandas as pd
from sklearn.datasets import load_iris
iris = load_iris()
```

In [39]:
```
iris.feature_names
```

Out[39]:
```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

In [40]:
```
iris.target_names
```

Out[40]:
```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

In [41]:
```
iris.data
```

```
       [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],
       [5.4, 3.7, 1.5, 0.2],
       [4.8, 3.4, 1.6, 0.2],
       [4.8, 3. , 1.4, 0.1],
       [4.3, 3. , 1.1, 0.1],
       [5.8, 4. , 1.2, 0.2],
       [5.7, 4.4, 1.5, 0.4],
       [5.4, 3.9, 1.3, 0.4],
       [5.1, 3.5, 1.4, 0.3],
       [5.7, 3.8, 1.7, 0.3],
       [5.1, 3.8, 1.5, 0.3],
       [5.4, 3.4, 1.7, 0.2],
       [5.1, 3.7, 1.5, 0.4],
       [4.6, 3.6, 1. , 0.2]
```

In [42]:
```
df = pd.DataFrame(iris.data,columns = iris.feature_names)
df.head()
```

Out[42]:

| | 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 [43]:

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

Out[43]:

| | sepal length (cm) | sepal width (cm) | petal length (cm) | petal width (cm) | flower |
|---|---|---|---|---|---|
| **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 [44]:

```
df.drop(['sepal length (cm)', 'sepal width (cm)', 'flower'],axis='columns',inplace=True
```

In [45]:

```
df.head()
```

Out[45]:

| | 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 [46]:

```
from sklearn.cluster import KMeans
```

In [63]:

```
km = KMeans(n_clusters = 3)
y_predicted = km.fit_predict(df)
y_predicted
```

Out[63]:

```
array([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, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
```

In [73]:

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

Out[73]:

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

In [74]:

```
km.cluster_centers_
```

Out[74]:

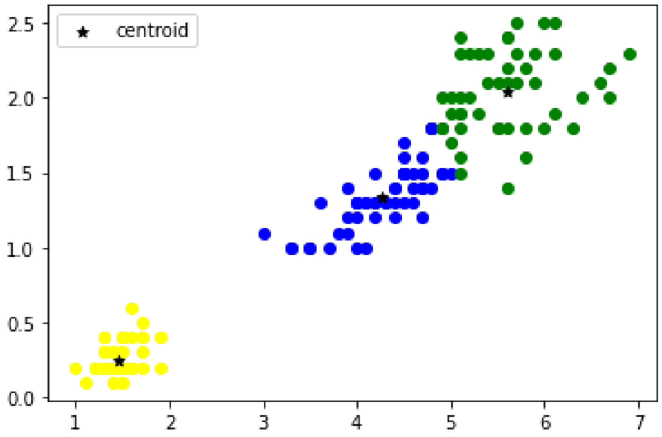
```
array([[4.26923077, 1.34230769, 0.        ],
       [5.59583333, 2.0375    , 0.        ],
       [1.462     , 0.246     , 0.        ]])
```

In [70]:
```
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]
```

In [71]:
```
import matplotlib.pyplot as plt
```

In [76]:
```
plt.scatter(df1['petal length (cm)'],df1['petal width (cm)'],color='blue')
plt.scatter(df2['petal length (cm)'],df2['petal width (cm)'],color='green')
plt.scatter(df3['petal length (cm)'],df3['petal width (cm)'],color='yellow')
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color = 'black',marker =
plt.legend()
```

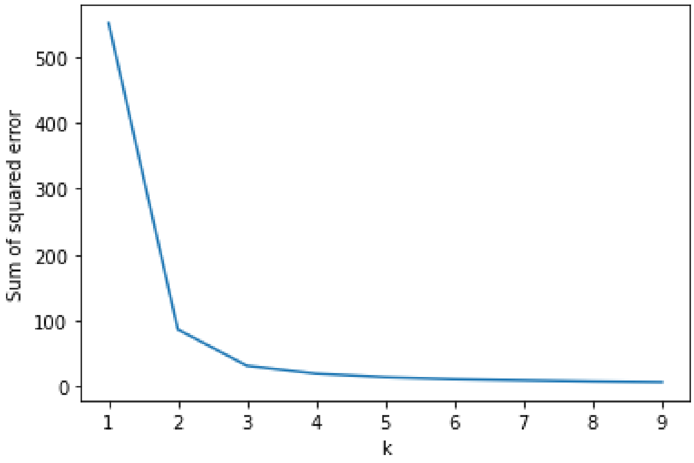
Out[76]:
```
<matplotlib.legend.Legend at 0x2e5a2b2c070>
```



In [80]:
```
sse = []
k_rng = range(1,10)
for k in k_rng:
    km = KMeans(n_clusters = k)
    km.fit(df[['petal length (cm)' , 'petal width (cm)']])
    sse.append(km.inertia_)
```

In [81]:
```
plt.xlabel('k')
plt.ylabel('Sum of squared error')
plt.plot(k_rng,sse)
```

Out[81]:
```
[<matplotlib.lines.Line2D at 0x2e5a2d51cd0>]
```



In [ ]:
```
CONCLUSION -
           The optimal value for k is 2
```