In [1]:
```
from sklearn.cluster import KMeans
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from matplotlib import pyplot as plt
%matplotlib inline
```

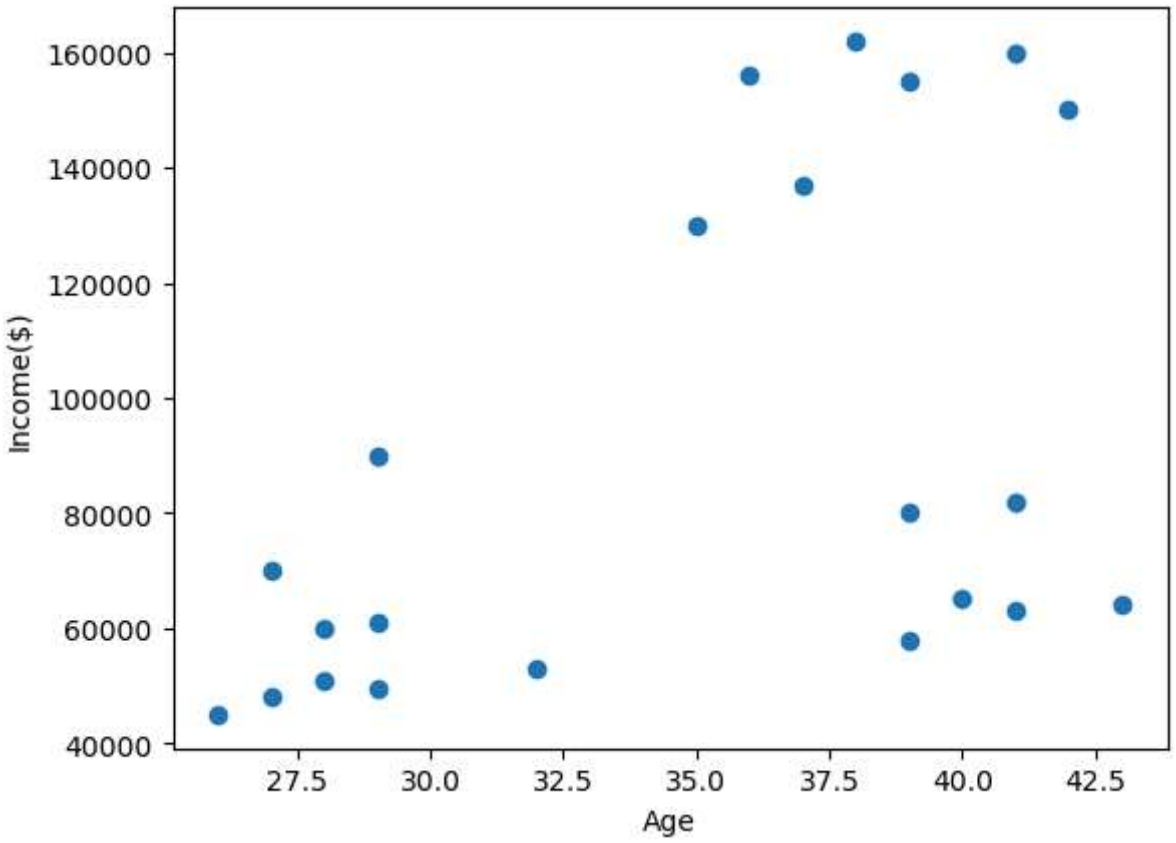
In [13]:
```
df = pd.read_csv('C:\\Users\\aiml\\Desktop\\income.csv')
df.head()
```

Out[13]:

	Name	Age	Income($)
0	Rob	27	70000
1	Michael	29	90000
2	Mohan	29	61000
3	Ismail	28	60000
4	Kory	42	150000

In [14]:
```
plt.scatter(df.Age,df['Income($)'])
plt.xlabel('Age')
plt.ylabel('Income($)')
```

Out[14]:
```
Text(0, 0.5, 'Income($)')
```



In [15]:
```
km = KMeans(n_clusters=3)
y_predicted = km.fit_predict(df[['Age','Income($)']])
y_predicted
```

```
C:\Users\aiml\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: Fu
tureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\aiml\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: 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[15]:
```
array([0, 0, 2, 2, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 2])
```

In [16]:
```
df['cluster']=y_predicted
df.head()
```

Out[16]:

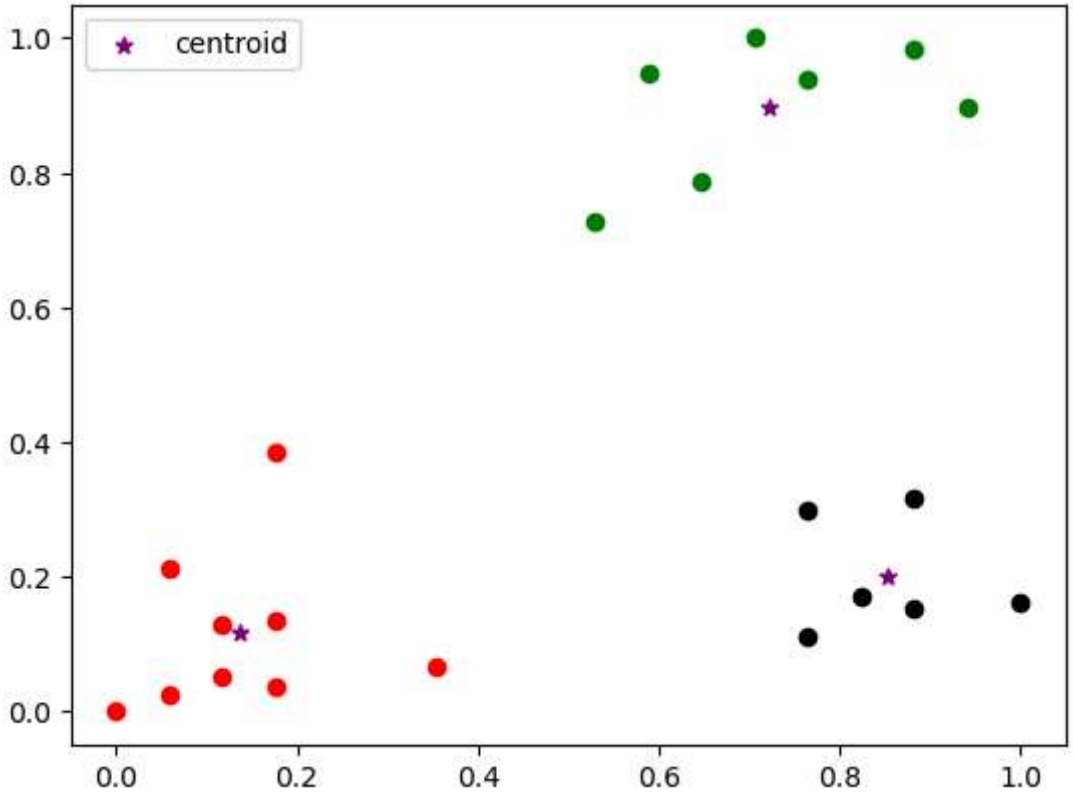
	Name	Age	Income($)	cluster
0	Rob	27	70000	0
1	Michael	29	90000	0
2	Mohan	29	61000	2
3	Ismail	28	60000	2
4	Kory	42	150000	1

In [17]:
```
km.cluster_centers_
```

Out[17]:
```
array([[3.40000000e+01, 8.05000000e+04],
       [3.82857143e+01, 1.50000000e+05],
       [3.29090909e+01, 5.61363636e+04]])
```

In [25]:
```
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]
plt.scatter(df1.Age,df1['Income($)'],color='green')
plt.scatter(df2.Age,df2['Income($)'],color='red')
plt.scatter(df3.Age,df3['Income($)'],color='black')
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='purple',ma
plt.legend()
```

Out[25]:
```
<matplotlib.legend.Legend at 0x1a89c6c09d0>
```



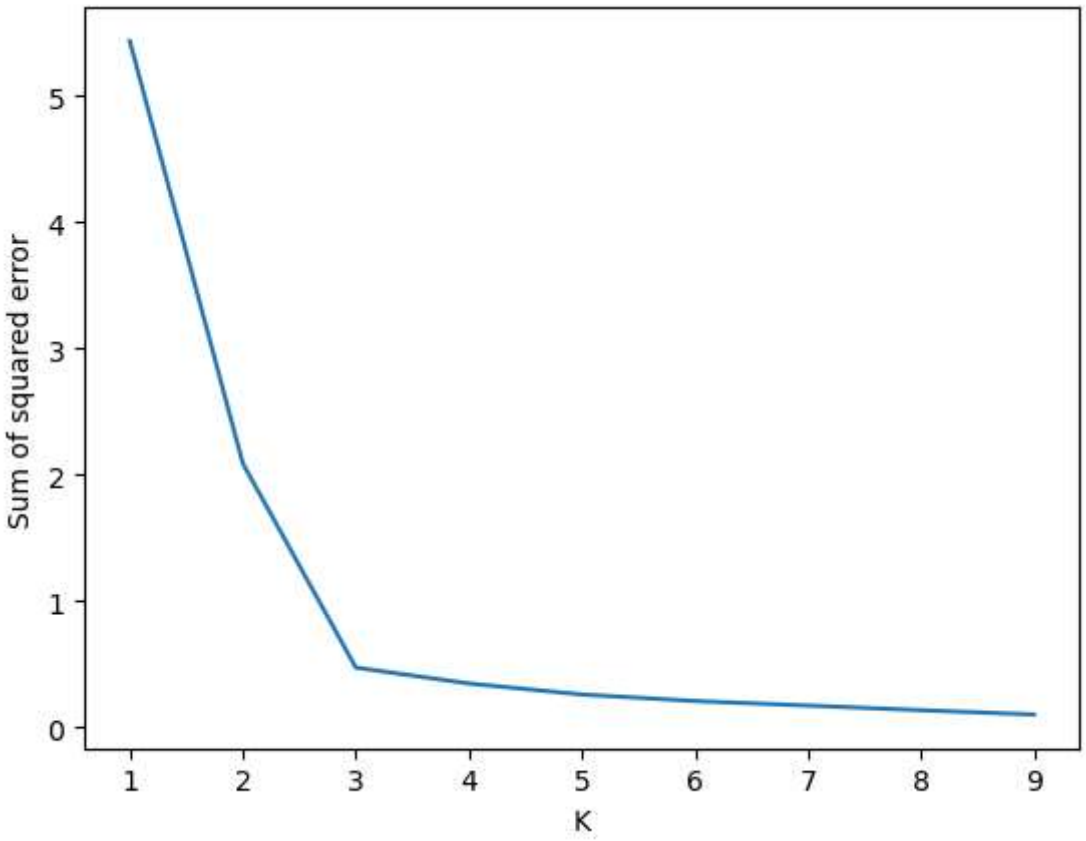
In [26]:
```
# Elbow Plot


sse = []
k_rng = range(1,10)
for k in k_rng:
    km = KMeans(n_clusters=k)
    km.fit(df[['Age','Income($)']])
    sse.append(km.inertia_)
```

```
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\aiml\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: 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(
C:\Users\aiml\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: Fu
tureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\aiml\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: 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(
C:\Users\aiml\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: Fu
tureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\aiml\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: 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(
```

In [27]:
```
plt.xlabel('K')
plt.ylabel('Sum of squared error')
plt.plot(k_rng,sse)
```

Out[27]: [<matplotlib.lines.Line2D at 0x1a89d0e5d10>]



In [ ]: