

In []:

```
1 '''
2
3 Support Vector Machin SVM is a supervised learning algorithm
4 used for classification as well as Regression problems
5
6 used to create a hyperline or best line or decision boundary that segregate n-dimension
7 into classes for future entries
8
9 2 types linear and non-linear
10
11 SVM can be used for face detection,image classification and categorization etc.
12
13 '''
```

In [1]:

```
1 from sklearn.datasets import load_iris
2
3 X,y = load_iris(return_X_y=True) # splitting input and output
```

In [2]:

```
1 from sklearn.model_selection import train_test_split
2
3 xtrain,xtest,ytrain,ytest = train_test_split(X,y,test_size=0.3,random_state=0)
4 # splitting training and testing data into 70% for training and 30% for testing
```

In [3]:

```
1 from sklearn.svm import SVC
2
3 clf = SVC(kernel='linear',random_state=0)
4
5 clf.fit(xtrain,ytrain)
```

Out[3]:

```
SVC(kernel='linear', random_state=0)
```

In [4]:

```
1 ypred = clf.predict(xtest)
2
3 ypred
```

Out[4]:

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

In [5]:

```
1 ypred = clf.predict(xtest)
2
3 ypred
```

Out[5]:

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

In [6]:

```
1 from sklearn.metrics import confusion_matrix
2
3 cm = confusion_matrix(ytest,ypred)
4 cm
```

Out[6]:

```
array([[16,  0,  0],
       [ 0, 17,  1],
       [ 0,  0, 11]], dtype=int64)
```

In [10]:

```
1 from sklearn.metrics import classification_report
2
3 cr = classification_report(ytest,ypred)
4 print(cr)
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	16
1	1.00	0.94	0.97	18
2	0.92	1.00	0.96	11
accuracy			0.98	45
macro avg	0.97	0.98	0.98	45
weighted avg	0.98	0.98	0.98	45

In [7]:

```
1 from sklearn.metrics import accuracy_score
2
3 ac = accuracy_score(ytest,ypred)
4 ac = ac*100
5 ac
```

Out[7]:

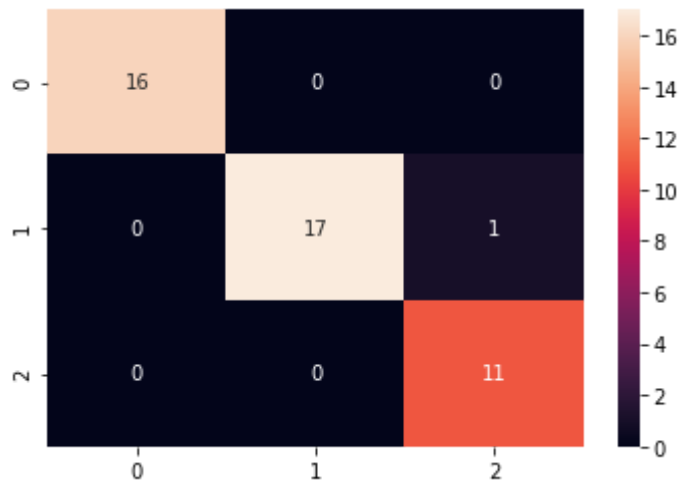
```
97.77777777777777
```

In [8]:

```
1 import seaborn as sns
2 sns.heatmap(cm,annot=True)
```

Out[8]:

<matplotlib.axes._subplots.AxesSubplot at 0x1f1783fef40>



In []:

1