

In [1]:

```
from sklearn.datasets import load_iris
```

In [2]:

```
iris=load_iris()
```

In [3]:

```
X=iris.data
```

In [4]:

```
y=iris.target  
y
```

Out[4]:

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

In [5]:

```
from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test=train_test_split(X,y, test_size=0.2)
```

In [6]:

```
X_train.shape
```

Out[6]:

```
(120, 4)
```

In [7]:

```
X_test.shape
```

Out[7]:

```
(30, 4)
```

In [8]:

```
from sklearn.tree import DecisionTreeClassifier
```

In [9]:

```
clf=DecisionTreeClassifier()
```

In [10]:

```
clf.fit(X_train,y_train)
```

Out[10]:

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',  
                       max_depth=None, max_features=None, max_leaf_nodes=None,
```

```
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, presort='deprecated',
random_state=None, splitter='best')
```

In [11]:

```
y_pred=clf.predict(X_test)
```

In [12]:

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

Out[12]:

0.9

In [13]:

```
from sklearn.tree import plot_tree
```

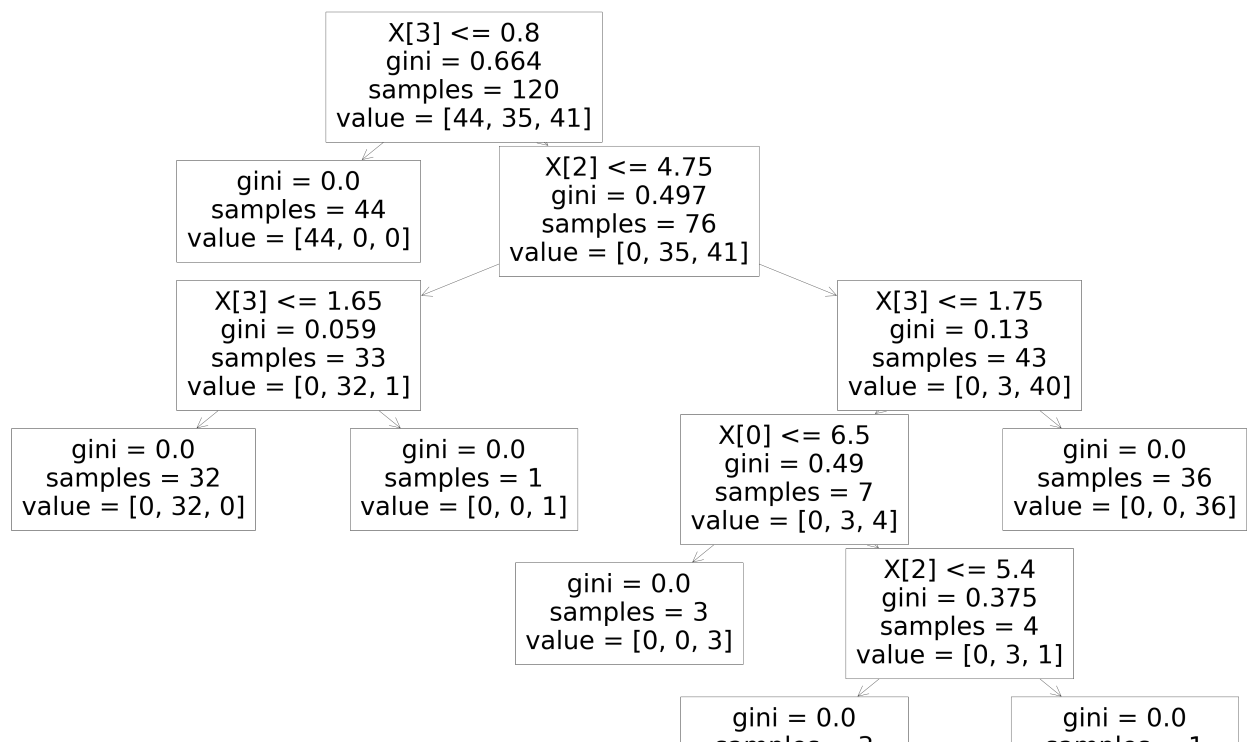
In [14]:

```
from matplotlib.pylab import rcParams
rcParams['figure.figsize'] = 80,50

plot_tree(clf)
```

Out[14]:

```
[Text(1674.0, 2491.5, 'X[3] <= 0.8\ngini = 0.664\nsamples = 120\nvalue = [44, 35, 41]'),
Text(1116.0, 2038.5, 'gini = 0.0\nsamples = 44\nvalue = [44, 0, 0]'),
Text(2232.0, 2038.5, 'X[2] <= 4.75\ngini = 0.497\nsamples = 76\nvalue = [0, 35, 41]'),
Text(1116.0, 1585.5, 'X[3] <= 1.65\ngini = 0.059\nsamples = 33\nvalue = [0, 32, 1]'),
Text(558.0, 1132.5, 'gini = 0.0\nsamples = 32\nvalue = [0, 32, 0]'),
Text(1674.0, 1132.5, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
Text(3348.0, 1585.5, 'X[3] <= 1.75\ngini = 0.13\nsamples = 43\nvalue = [0, 3, 40]'),
Text(2790.0, 1132.5, 'X[0] <= 6.5\ngini = 0.49\nsamples = 7\nvalue = [0, 3, 4]'),
Text(2232.0, 679.5, 'gini = 0.0\nsamples = 3\nvalue = [0, 0, 3]'),
Text(3348.0, 679.5, 'X[2] <= 5.4\ngini = 0.375\nsamples = 4\nvalue = [0, 3, 1]'),
Text(2790.0, 226.5, 'gini = 0.0\nsamples = 3\nvalue = [0, 3, 0]'),
Text(3906.0, 226.5, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
Text(3906.0, 1132.5, 'gini = 0.0\nsamples = 36\nvalue = [0, 0, 36]')]
```



```
samples = 3  
value = [0, 3, 0]
```

```
samples = 1  
value = [0, 0, 1]
```

```
In [ ]:
```