In [3]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
```

In [18]:
```
dataset = pd.read_csv("C:/Users/Admin-PC/Downlads/iris.csv")
```

In [23]:
```
%matplotlib inline
img=mpimg.imread("C:/Users/Admin-PC/Downlads/iris_types.jpg")
plt.figure(figsize=(20,40))
plt.axis('of')
plt.imshow(img)
```

Out[23]:
```
<matplotlib.image.AxesImage at 0x15d62bfd430>
```

In [19]:
```
dataset.head()
```

Out[19]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [5]:
```
X = dataset.iloc[:,:4].values
y = dataset['species'].values
```

In [6]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 82)
```

In [7]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [8]:
```
from sklearn.naive_bayes import GaussianNB
nvclassifier = GaussianNB()
nvclassifier.fit(X_train, y_train)
```

Out[8]:
```
GaussianNB()
```

In [9]:
```
y_pred = nvclassifier.predict(X_test)
print(y_pred)
```

```
['virginica' 'virginica' 'setosa' 'setosa' 'setosa' 'virginica'
 'versicolor' 'versicolor' 'versicolor' 'versicolor' 'versicolor'
 'virginica' 'setosa' 'setosa' 'setosa' 'setosa' 'virginica' 'versicolor'
 'setosa' 'versicolor' 'setosa' 'virginica' 'setosa' 'virginica'
 'virginica' 'versicolor' 'virginica' 'setosa' 'virginica' 'versicolor']
```

In [10]:
```
y_compare = np.vstack((y_test,y_pred))
#actual value on the left side and predicted value on the right hand side
#printing the top 5 values
y_compare[:5,:]
```

Out[10]:
```
array([['virginica', 'virginica', 'setosa', 'setosa', 'setosa',
        'versicolor', 'versicolor', 'versicolor', 'virginica',
        'versicolor', 'versicolor', 'virginica', 'setosa', 'setosa',
        'setosa', 'setosa', 'virginica', 'versicolor', 'setosa',
        'versicolor', 'setosa', 'virginica', 'setosa', 'virginica',
        'virginica', 'versicolor', 'virginica', 'setosa', 'virginica',
        'versicolor'],
       ['virginica', 'virginica', 'setosa', 'setosa', 'setosa',
        'virginica', 'versicolor', 'versicolor', 'versicolor',
        'versicolor', 'versicolor', 'virginica', 'setosa', 'setosa',
        'setosa', 'setosa', 'virginica', 'versicolor', 'setosa',
        'versicolor', 'setosa', 'virginica', 'setosa', 'virginica',
        'virginica', 'versicolor', 'virginica', 'setosa', 'virginica',
        'versicolor']], dtype=object)
```

In [11]:
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)
```

```
[[11  0  0]
 [ 0  8  1]
 [ 0  1  9]]
```

a = cm.shape corrPred = 0 falsePred = 0

for row in range(a[0]): for c in range(a[1]): if row == c: corrPred +=cm[row,c] else: falsePred += cm[row,c] print('Correct predictions: ', corrPred) print('False predictions', falsePred) print ('\n\nAccuracy of the Naive Bayes Clasification is: ', corrPred/(cm.sum()))

In [24]:
```
a = cm.shape
corrPred = 0
falsePred = 0

for row in range(a[0]):
    for c in range(a[1]):
        if row == c:
            corrPred +=cm[row,c]
        else:
            falsePred += cm[row,c]
print('Correct predictions: ', corrPred)
print('False predictions', falsePred)
print ('\n\nAccuracy of the Naive Bayes Clasification is: ', corrPred/(cm.sum()))
```

```
Correct predictions:  28
False predictions 2


Accuracy of the Naive Bayes Clasification is:  0.9333333333333333
```

In [ ]:
```

```