In [1]:
```
# import necessary libraries
# sklearn contains some predefined datasets. For this exercise we are using Load_
# Load digits contains a dataset of 1797 images of 8*8 size.
from sklearn.datasets import load_digits
%matplotlib inline
import matplotlib.pyplot as plt
digits = load_digits()
```

In [3]:
```
# discription of the data
dir(digits)
```

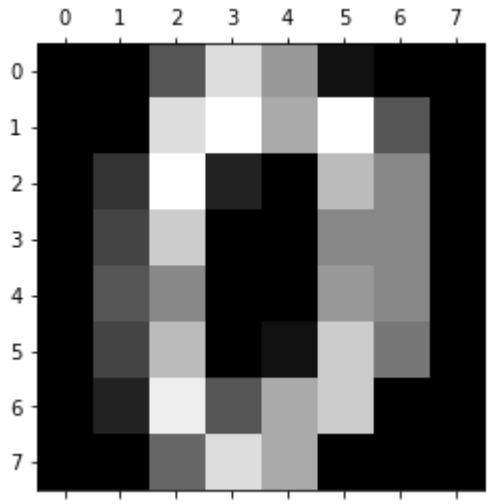
Out[3]:
```
['DESCR', 'data', 'images', 'target', 'target_names']
```

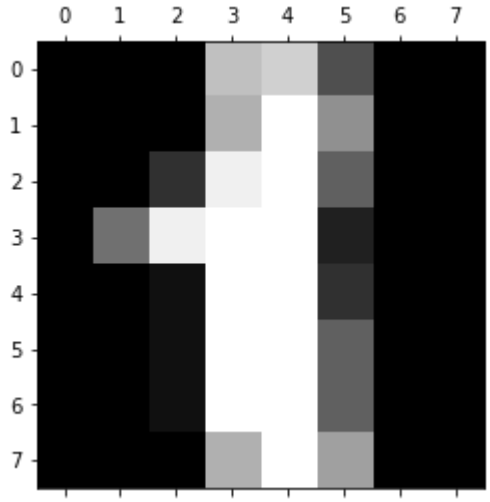
In [4]:
```
# see the data
digits.data[0]
```

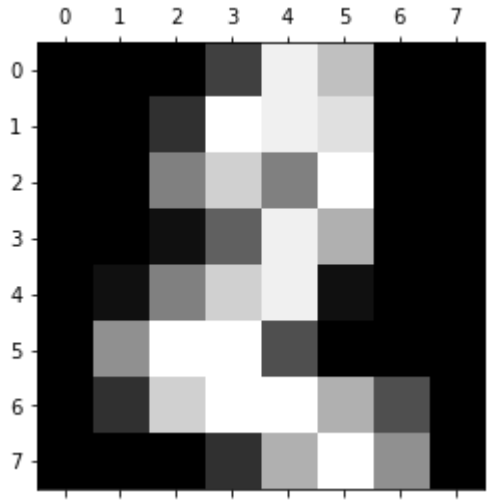
Out[4]:
```
array([ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.,  0.,  0., 13., 15., 10.,
       15.,  5.,  0.,  0.,  3., 15.,  2.,  0., 11.,  8.,  0.,  0.,  4.,
       12.,  0.,  0.,  8.,  8.,  0.,  0.,  5.,  8.,  0.,  0.,  9.,  8.,
        0.,  0.,  4., 11.,  0.,  1., 12.,  7.,  0.,  0.,  2., 14.,  5.,
       10., 12.,  0.,  0.,  0.,  0.,  6., 13., 10.,  0.,  0.,  0.])
```

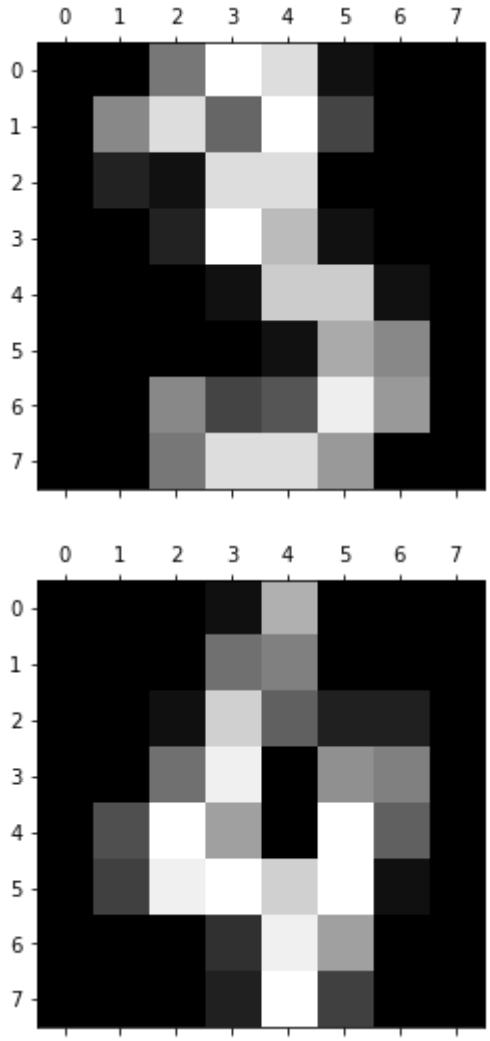
In [5]:
```
# see the image data for first 5 data
plt.gray()
for i in range(5):
    plt.matshow(digits.images[i])
```

```
<matplotlib.figure.Figure at 0x20e79c0c320>
```





In [6]:
```
# Create and train logistic regression model
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
```

In [7]:
```
# separate the data betwen train and test data
from sklearn.model_selection import train_test_split
```

In [8]:
```
# Divide the data between training and test dataset

X_train, X_test, y_train, y_test = train_test_split(digits.data,digits.target, te
```

In [9]:
```
# train the data
model.fit(X_train, y_train)
```

Out[9]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
```

In [10]:
```
# Measure accuracy of the model

model.score(X_test, y_test)
```

Out[10]:
```
0.9611111111111111
```

In [11]:
```
# predict the first five images
model.predict(digits.data[0:5])
```

Out[11]:
```
array([0, 1, 2, 3, 4])
```

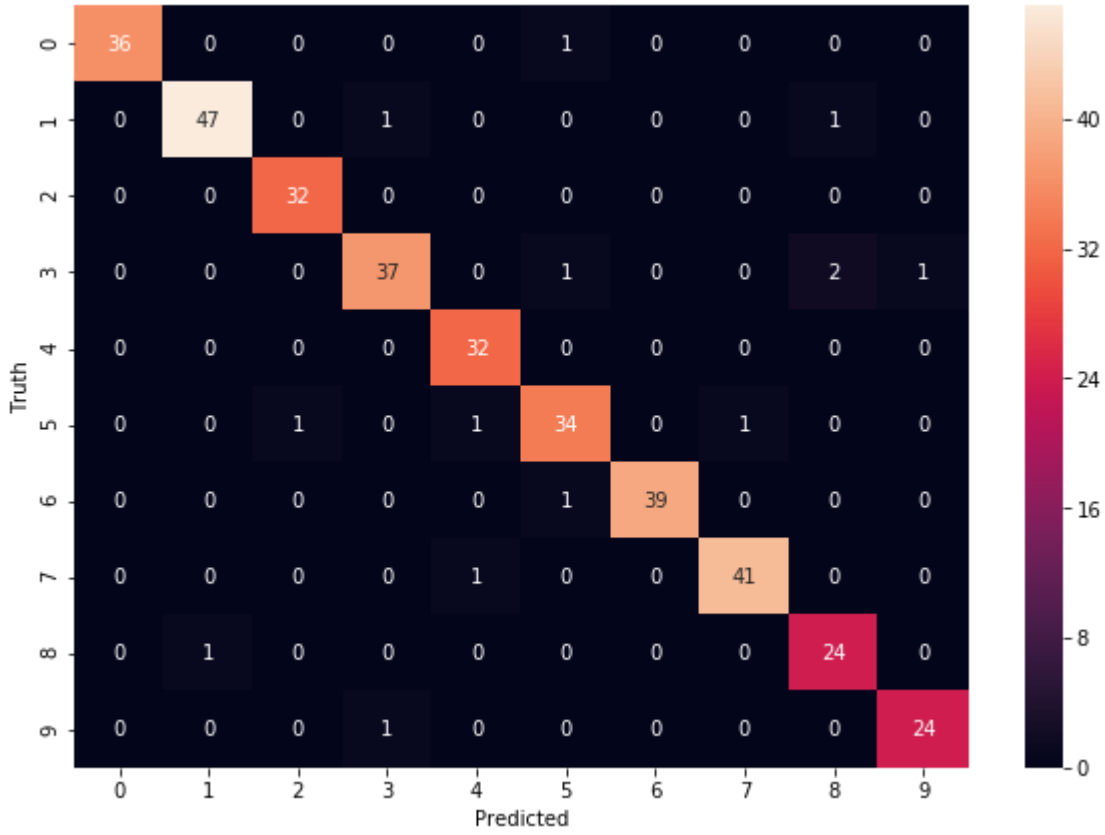
In [12]:
```
# Evaluate the confusion matrix

y_predicted = model.predict(X_test)
```

In [13]:
```
# Evaluate the confusion matrix for the datasex
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_predicted)
cm
```

Out[13]:
```
array([[36,  0,  0,  0,  0,  1,  0,  0,  0,  0],
       [ 0, 47,  0,  1,  0,  0,  0,  0,  1,  0],
       [ 0,  0, 32,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  0, 37,  0,  1,  0,  0,  2,  1],
       [ 0,  0,  0,  0, 32,  0,  0,  0,  0,  0],
       [ 0,  0,  1,  0,  1, 34,  0,  1,  0,  0],
       [ 0,  0,  0,  0,  0,  1, 39,  0,  0,  0],
       [ 0,  0,  0,  0,  1,  0,  0, 41,  0,  0],
       [ 0,  1,  0,  0,  0,  0,  0,  0, 24,  0],
       [ 0,  0,  0,  1,  0,  0,  0,  0,  0, 24]], dtype=int64)
```

In [14]:
```
# Use seaborn library to plot the confusion Matrix
import seaborn as sn
plt.figure(figsize = (10,7))
sn.heatmap(cm, annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

Out[14]: Text(69,0.5,'Truth')




In [ ]:
```

```