In [1]:
```
# importing necessary libraries and use digits datasets from sklearn

import pandas as pd
from sklearn.datasets import load_digits
digits = load_digits()
```

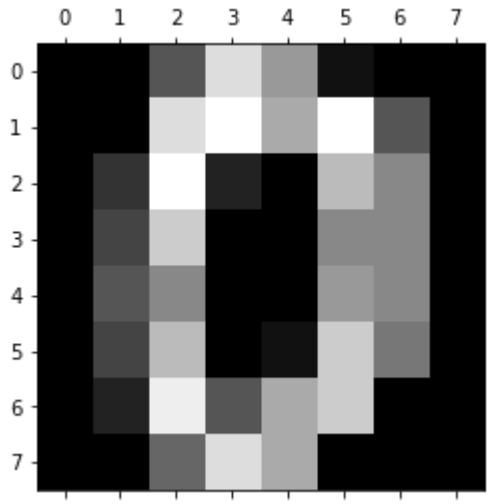
In [2]:
```
dir(digits)
```

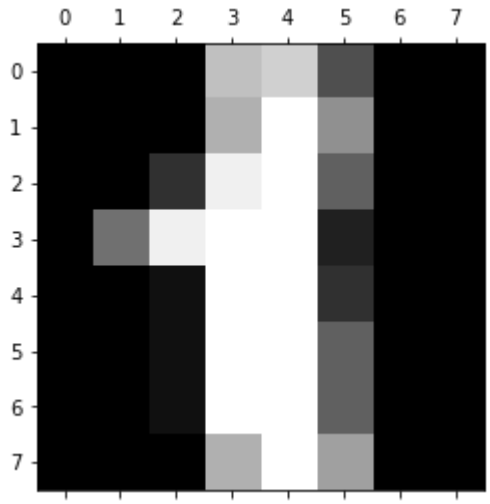
Out[2]:
```
['DESCR', 'data', 'images', 'target', 'target_names']
```

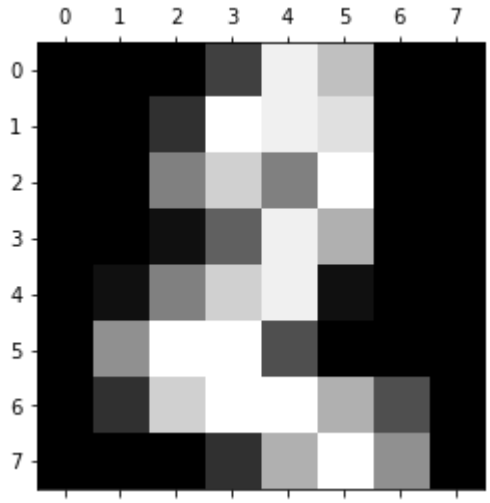
In [3]:
```
%matplotlib inline
import matplotlib.pyplot as plt
```

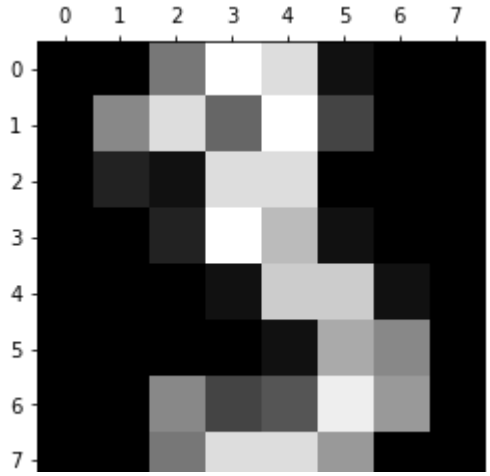
In [4]:
```
# plot the data
plt.gray()
for i in range(4):
    plt.matshow(digits.images[i])
```

```
<matplotlib.figure.Figure at 0x2378ed54048>
```





In [5]:
```
df = pd.DataFrame(digits.data)
df.head()
```

Out[5]:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 6 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.0 | 0.0 | 5.0 | 13.0 | 9.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 6.0 | 13.0 | 10.0 | 0.0 | 0. |
| **1** | 0.0 | 0.0 | 0.0 | 12.0 | 13.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 11.0 | 16.0 | 10.0 | 0. |
| **2** | 0.0 | 0.0 | 0.0 | 4.0 | 15.0 | 12.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 | 3.0 | 11.0 | 16.0 | 9. |
| **3** | 0.0 | 0.0 | 7.0 | 15.0 | 13.0 | 1.0 | 0.0 | 0.0 | 0.0 | 8.0 | ... | 9.0 | 0.0 | 0.0 | 0.0 | 7.0 | 13.0 | 13.0 | 9.0 | 0. |
| **4** | 0.0 | 0.0 | 0.0 | 1.0 | 11.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 16.0 | 4.0 | 0. |

5 rows × 64 columns

In [6]:
```
df['target'] = digits.target
```

In [7]:
```
df[0:12]
```

Out[7]:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 6 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.0 | 0.0 | 5.0 | 13.0 | 9.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 6.0 | 13.0 | 10.0 | 0.0 | 0. |
| **1** | 0.0 | 0.0 | 0.0 | 12.0 | 13.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 11.0 | 16.0 | 10.0 | 0. |
| **2** | 0.0 | 0.0 | 0.0 | 4.0 | 15.0 | 12.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 3.0 | 11.0 | 16.0 | 9. |
| **3** | 0.0 | 0.0 | 7.0 | 15.0 | 13.0 | 1.0 | 0.0 | 0.0 | 0.0 | 8.0 | ... | 0.0 | 0.0 | 0.0 | 7.0 | 13.0 | 13.0 | 9.0 | 0. |
| **4** | 0.0 | 0.0 | 0.0 | 1.0 | 11.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 16.0 | 4.0 | 0. |
| **5** | 0.0 | 0.0 | 12.0 | 10.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 9.0 | 16.0 | 16.0 | 10.0 | 0. |
| **6** | 0.0 | 0.0 | 0.0 | 12.0 | 13.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 1.0 | 9.0 | 15.0 | 11.0 | 3. |
| **7** | 0.0 | 0.0 | 7.0 | 8.0 | 13.0 | 16.0 | 15.0 | 1.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 13.0 | 5.0 | 0.0 | 0.0 | 0. |
| **8** | 0.0 | 0.0 | 9.0 | 14.0 | 8.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 11.0 | 16.0 | 15.0 | 11.0 | 1. |
| **9** | 0.0 | 0.0 | 11.0 | 12.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 0.0 | 0.0 | 0.0 | 9.0 | 12.0 | 13.0 | 3.0 | 0. |
| **10** | 0.0 | 0.0 | 1.0 | 9.0 | 15.0 | 11.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 1.0 | 10.0 | 13.0 | 3.0 | 0. |
| **11** | 0.0 | 0.0 | 0.0 | 0.0 | 14.0 | 13.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 13.0 | 16.0 | 1. |

12 rows × 65 columns

In [9]:
```
#Train and the model and prediction

X = df.drop('target',axis='columns')
y = df.target

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 [10]:
```
# Import Random forest classificer from sklearn

from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=20)
model.fit(X_train, y_train)
```

Out[10]:
```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', 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, n_estimators=20, n_jobs=1,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False)
```

In [11]:
```
# predict the accuracy of the model

model.score(X_test, y_test)
```

Out[11]:
```
0.9694444444444444
```

In [12]:
```
y_predicted = model.predict(X_test)
```

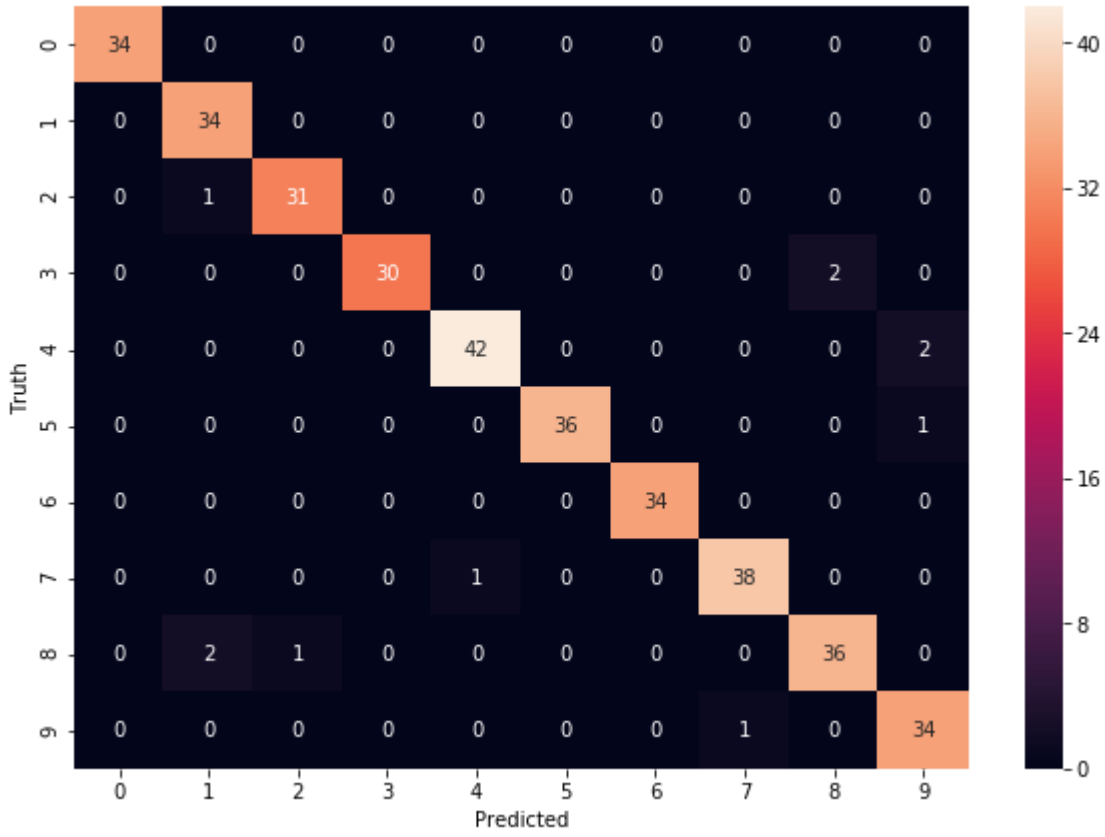
In [13]:
```
# Evaluate the confusion matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_predicted)
cm
```

Out[13]:
```
array([[34,  0,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0, 34,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  1, 31,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  0, 30,  0,  0,  0,  0,  2,  0],
       [ 0,  0,  0,  0, 42,  0,  0,  0,  0,  2],
       [ 0,  0,  0,  0,  0, 36,  0,  0,  0,  1],
       [ 0,  0,  0,  0,  0,  0, 34,  0,  0,  0],
       [ 0,  0,  0,  0,  1,  0,  0, 38,  0,  0],
       [ 0,  2,  1,  0,  0,  0,  0,  0, 36,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  1,  0, 34]], dtype=int64)
```

In [14]:
```
# use seaborn library to plot the data

%matplotlib inline
import matplotlib.pyplot as plt
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 [ ]: