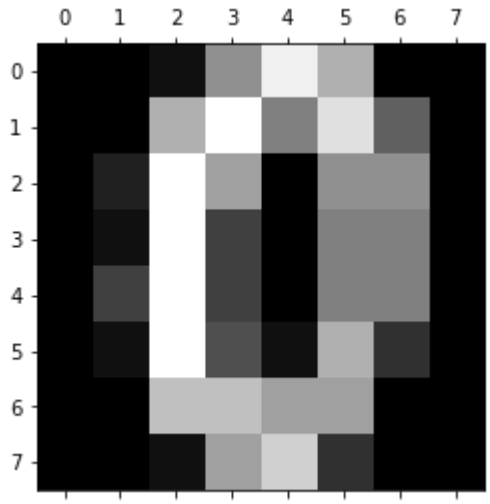
In [1]:
```
from sklearn.datasets import load_digits
```

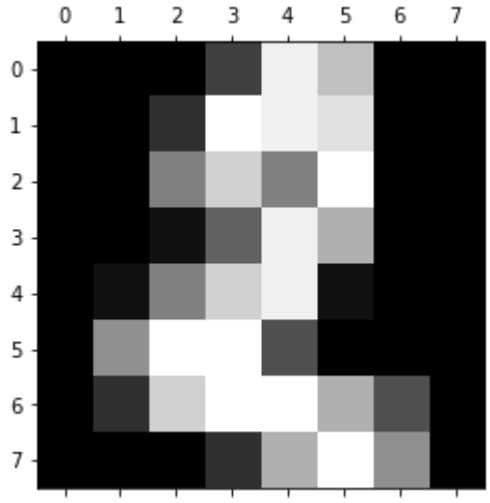
In [2]:
```
digits = load_digits()
import matplotlib.pyplot as plt
plt.gray()
plt.matshow(digits.images[10])
plt.show()
```

```
<matplotlib.figure.Figure at 0x24ab39290f0>
```



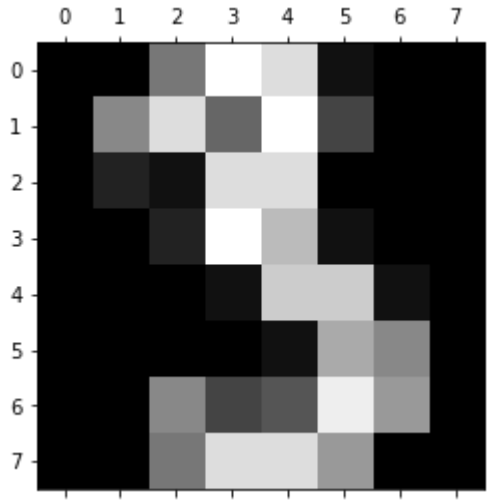
In [4]:
```
digits = load_digits()
import matplotlib.pyplot as plt
plt.gray()
plt.matshow(digits.images[2])
plt.show()
```

```
<matplotlib.figure.Figure at 0x24ab44b5b70>
```



In [5]:
```
digits = load_digits()
import matplotlib.pyplot as plt
plt.gray()
plt.matshow(digits.images[3])
plt.show()
```

```
<matplotlib.figure.Figure at 0x24ab44ac0b8>
```



In [6]:
```
digits.images[10]
```

Out[6]:
```
array([[  0.,   0.,   1.,   9.,  15.,  11.,   0.,   0.],
       [  0.,   0.,  11.,  16.,   8.,  14.,   6.,   0.],
       [  0.,   2.,  16.,  10.,   0.,   9.,   9.,   0.],
       [  0.,   1.,  16.,   4.,   0.,   8.,   8.,   0.],
       [  0.,   4.,  16.,   4.,   0.,   8.,   8.,   0.],
       [  0.,   1.,  16.,   5.,   1.,  11.,   3.,   0.],
       [  0.,   0.,  12.,  12.,  10.,  10.,   0.,   0.],
       [  0.,   0.,   1.,  10.,  13.,   3.,   0.,   0.]])
```

In [7]:
```
n_samples = len(digits.images)
```

In [8]:
```
X = digits.images.reshape((n_samples, -1))
```

In [9]:
```
X
```

Out[9]:
```
array([[  0.,   0.,   5., ...,   0.,   0.,   0.],
       [  0.,   0.,   0., ...,  10.,   0.,   0.],
       [  0.,   0.,   0., ...,  16.,   9.,   0.],
       ...,
       [  0.,   0.,   1., ...,   6.,   0.,   0.],
       [  0.,   0.,   2., ...,  12.,   0.,   0.],
       [  0.,   0.,  10., ...,  12.,   1.,   0.]])
```

In [10]:
```
y = digits.target
```

In [11]:
```
y
```

Out[11]:
```
array([0, 1, 2, ..., 8, 9, 8])
```

In [12]:
```
from sklearn.model_selection import train_test_split
```

In [13]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_
```

In [14]:
```
from sklearn.neural_network import MLPClassifier
```

In [32]:
```
mlp = MLPClassifier(hidden_layer_sizes=(10,10))
```

In [33]:
```
mlp.fit(X_train,y_train)
```

Out[33]:
```
MLPClassifier(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
       beta_2=0.999, early_stopping=False, epsilon=1e-08,
       hidden_layer_sizes=(10, 10), learning_rate='constant',
       learning_rate_init=0.001, max_iter=200, momentum=0.9,
       nesterovs_momentum=True, power_t=0.5, random_state=None,
       shuffle=True, solver='adam', tol=0.0001, validation_fraction=0.1,
       verbose=False, warm_start=False)
```

In [34]:
```
y_pred = mlp.predict(X_test)
```

In [35]:
```
print("Training set score: %f" % mlp.score(X_train, y_train))
print("Test set score: %f" % mlp.score(X_test, y_test))
```

```
Training set score: 0.893599
Test set score: 0.856902
```

In [36]:
```
from sklearn.metrics import accuracy_score
```

In [37]:
```
print('Accuracy: %.2f' %accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.86
```

In [38]:
```
mlp1= MLPClassifier(hidden_layer_sizes=(100,100))
mlp1.fit(X_train,y_train)
y_pred = mlp1.predict(X_test)
```

In [39]:
```
print("Training set score: %f" % mlp1.score(X_train, y_train))
print("Test set score: %f" % mlp1.score(X_test, y_test))
```

```
Training set score: 1.000000
Test set score: 0.971380
```

In [40]:
```
print('Accuracy: %.2f' %accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.97
```

In [ ]: