In [81]:

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

Out[81]:

```
['DESCR', 'data', 'images', 'target', 'target_names']
```

In [82]:

```
digits.data
```

Out[82]:

```
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 [83]:

```
digits.target_names
```

Out[83]:

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [84]:

```
df = pd.DataFrame(digits.data,digits.target)
df.head()
```

Out[84]:

	0	1	2	3	4	5	6	7	8	9	...	54	55	56	57	58	59	60	61	62	63
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	0.0	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	0.0	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	9.0	0.0
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	0.0	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	0.0	0.0

5 rows × 64 columns

In [85]:

```
df['target'] = digits.target
df.head(20)
```

Out[85]:

	0	1	2	3	4	5	6	7	8	9	...	55	56	57	58	59	60	61	62	63	target
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	0.0	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	0.0	0.0	1
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	9.0	0.0	2
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	0.0	0.0	3

	0	1	2	3	4	5	6	7	8	9	...	55	56	57	58	59	60	61	62	63	target
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	0.0	0.0	4
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	0.0	0.0	5
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	3.0	0.0	6
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	0.0	0.0	7
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	1.0	0.0	8
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	0.0	0.0	9
0	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	0.0	0.0	0
1	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	1.0	0.0	1
2	0.0	0.0	5.0	12.0	1.0	0.0	0.0	0.0	0.0	0.0	...	2.0	0.0	0.0	3.0	11.0	8.0	13.0	12.0	4.0	2
3	0.0	2.0	9.0	15.0	14.0	9.0	3.0	0.0	0.0	4.0	...	0.0	0.0	2.0	12.0	12.0	13.0	11.0	0.0	0.0	3
4	0.0	0.0	0.0	8.0	15.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	10.0	15.0	4.0	0.0	0.0	4
5	0.0	5.0	12.0	13.0	16.0	16.0	2.0	0.0	0.0	11.0	...	0.0	0.0	4.0	15.0	16.0	2.0	0.0	0.0	0.0	5
6	0.0	0.0	0.0	8.0	15.0	1.0	0.0	0.0	0.0	0.0	...	2.0	0.0	0.0	0.0	7.0	15.0	16.0	11.0	0.0	6
7	0.0	0.0	1.0	8.0	15.0	10.0	0.0	0.0	0.0	3.0	...	0.0	0.0	0.0	0.0	11.0	9.0	0.0	0.0	0.0	7
8	0.0	0.0	10.0	7.0	13.0	9.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	11.0	14.0	5.0	0.0	0.0	0.0	8
9	0.0	0.0	6.0	14.0	4.0	0.0	0.0	0.0	0.0	0.0	...	2.0	0.0	0.0	7.0	16.0	16.0	13.0	11.0	1.0	9

20 rows × 65 columns

In [86]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(digits.data, digits.target, test_size=0.2, rand
om_state=42)
```

In [87]:

```
len(X_train)
```

Out[87]:

```
1437
```

In [88]:

```
len(X_test)
```

Out[88]:

```
360
```

In [140]:

```
import numpy as np
from sklearn.svm import SVC
clf = SVC(C=10, gamma='scale', kernel='rbf')
clf.fit(X_train, y_train)
```

Out[140]:

```
SVC(C=10, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

In [141]:

```
clf.score(X_test, y_test)
```

Out[141]:

```
0.9861111111111112
```

In [142]:

```
import matplotlib.pyplot as plt
```

In [143]:

```
digits.data[0]
```

Out[143]:

```
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 [144]:

```
digits.images[0]
```

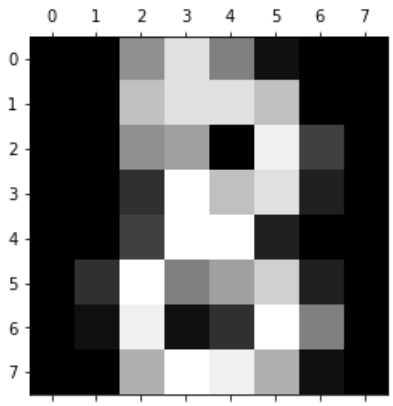
Out[144]:

```
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 [145]:

```
plt.gray()
plt.matshow(digits.images[8])
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



In [146]:

```
print(digits.target.shape)
print(digits.target)
```

```
(1797,)
[0 1 2 ... 8 9 8]
```

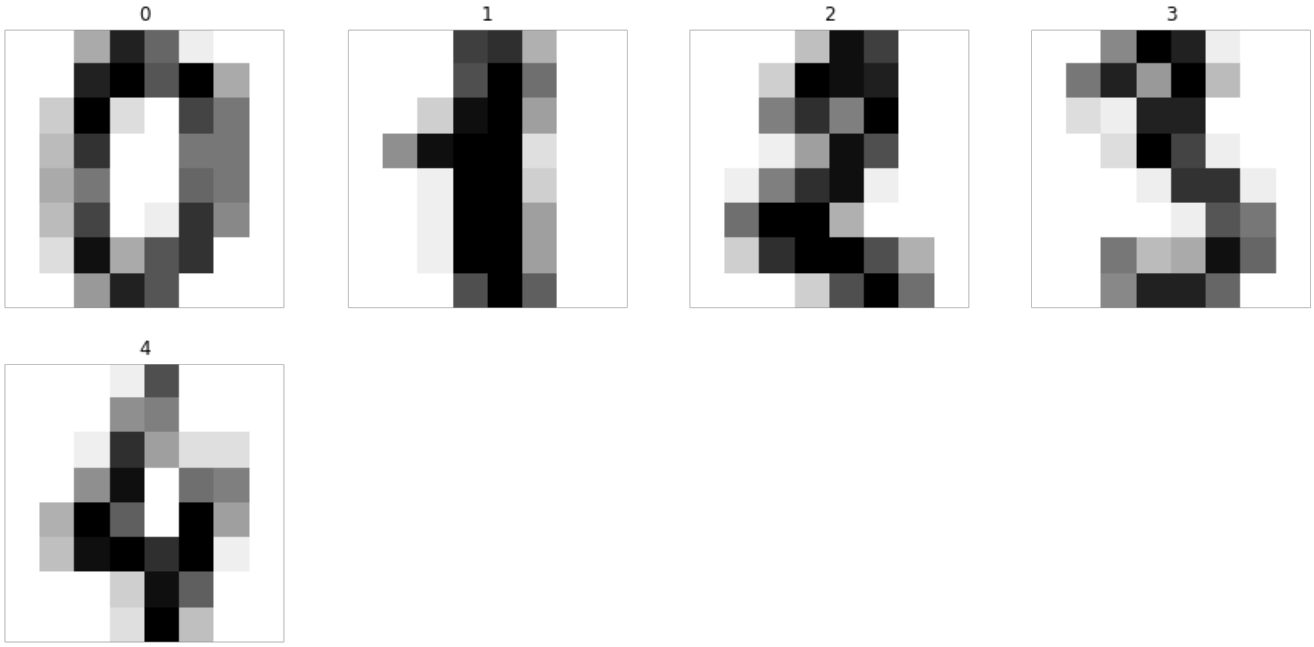
In [153]:

```
def plot_multi(i):
    '''Plots 16 digits, starting with digit i'''
    nplots = 5
```

```
    fig = plt.figure(figsize=(15,15))
    for j in range(nplots):
        plt.subplot(4,4,j+1)
        plt.imshow(digits.images[i+j], cmap='binary')
        plt.title(digits.target[i+j])
        plt.axis('off')
    plt.show()
```

In [154]:

```
plot_multi(0)
```



In [ ]: