In [52]:

```
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn.metrics import accuracy_score
from sklearn.datasets import load_digits
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
import random
```

有时候

In [53]:

```
## 数据加载
digits=load_digits()
digits
```

Out[53]:

```
{'data': 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.]]),
 'target': array([0, 1, 2, ..., 8, 9, 8]),
 'target_names': array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 'images': array([[[ 0.,  0.,  5., ...,  1.,  0.,  0.],
         [ 0.,  0., 13., ..., 15.,  5.,  0.],
         [ 0.,  3., 15., ..., 11.,  8.,  0.],
         ...,
         [ 0.,  4., 11., ..., 12.,  7.,  0.],
         [ 0.,  2., 14., ..., 12.,  0.,  0.],
         [ 0.,  0.,  6., ...,  0.,  0.,  0.]],

        [[ 0.,  0.,  0., ...,  5.,  0.,  0.],
```

In [54]:

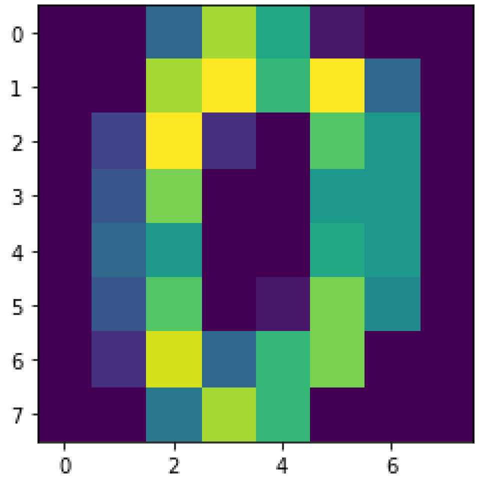
```
## 探索数据
#先看看数据的维度
data=digits.data
images=digits.images
print('data:',data.shape,type(data))
print('images:',images.shape,type(images))
print('target:',digits.target.shape,type(digits.target))
```

```
data: (1797, 64) <class 'numpy.ndarray'>
images: (1797, 8, 8) <class 'numpy.ndarray'>
target: (1797,) <class 'numpy.ndarray'>
```

In [55]:

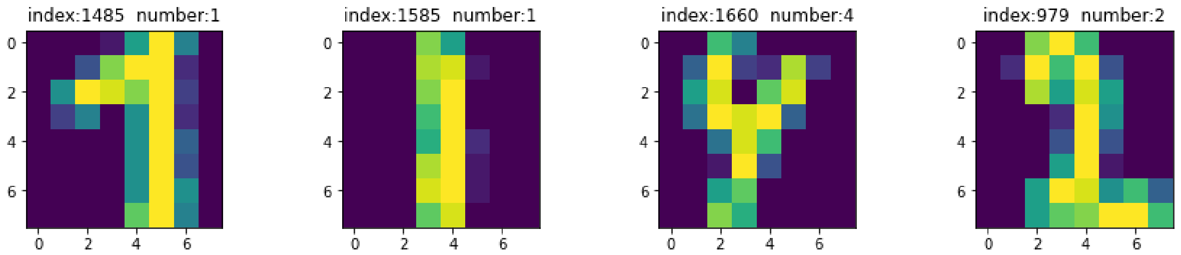
```
#data和images实际上数据是一样的 但是维度不同data中的64x1变成了images里的8x8 可以找个例子看看
print('data:\n',data[0],data[0].shape)
print('images\n',images[0],images[0].shape)
#images[0]是一个二维数组 可以将其用plt.imshow()方法画出来
plt.imshow(images[0])#显示出来的图形 应该跟标签是对应的吧
print('图片对应的target数字:',digits.target[0])
```

```
data:
 [ 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.] (64,)
images
 [[ 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.]] (8, 8)
图片对应的target数字: 0
```



In [56]:

```
# 随机找其中的几副图像 然后显示出来看看
plt.figure(figsize=(15,2.5))
i=1
for e in [random.randint(1,data.shape[0]) for i in range(4)]:#在（1,1797）中间随机的找4个数 然后将图
    ax=plt.subplot(140+i)#子图141 142 143 144
    plt.title('index:'+str(e)+'  number:'+str(digits.target[e]))
    plt.imshow(digits.images[e])
    i+=1
plt.show()
```



In [57]:

```
## 划分数据
# 分割数据，将25%的数据作为测试集，其余作为训练集
# 要训练的特征的数据是data  目标是target  只不过感觉这里的data维度有点高1794x64
train_x, test_x, train_y, test_y = train_test_split(data, digits.target, test_size=0.25, random_stat
```

In [58]:

```
## 模型创建与训练
model_dt=DecisionTreeClassifier()
model_dt.fit(train_x,train_y)
```

Out[58]:

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=None, max_features=None, 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, presort='deprecated',
                       random_state=None, splitter='best')
```

In [59]:

```
## 模型评估
predict_y=model_dt.predict(test_x)
print('模型的分数：',model_dt.score(test_x, test_y))
```

```
模型的分数： 0.8533333333333334
```