In [2]:

```
import re
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import metrics
%matplotlib inline
digits=load_digits()
```

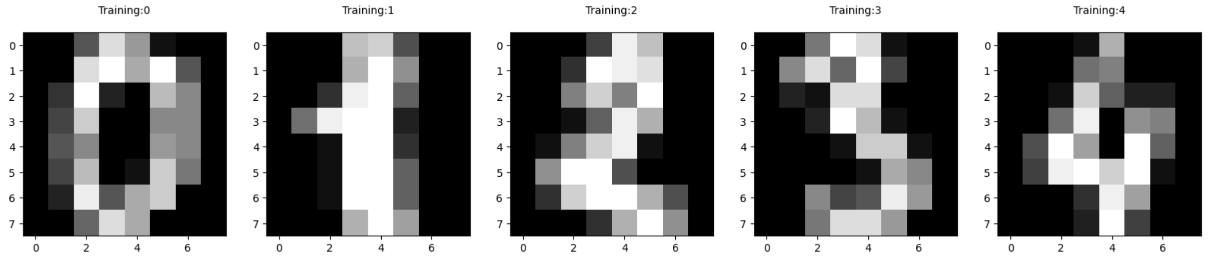
In [3]:

```
print("Image Data Shape",digits.data.shape)
print("Label Data Shape",digits.target.shape)
```

```
Image Data Shape (1797, 64)
Label Data Shape (1797,)
```

In [4]:

```
plt.figure(figsize=(20,4))
for index,(image,label)in enumerate(zip(digits.data[0:5],digits.target[0:5])):
 plt.subplot(1,5,index+1)
 plt.imshow(np.reshape(image,(8,8)),cmap=plt.cm.gray)
 plt.title('Training:%i\n'%label,fontsize=10)
```



In [6]:

```
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=0.30,random_stat
```

In [7]:

```
print(x_train.shape)
```

```
(1257, 64)
```

In [8]:

```
print(y_train.shape)
```

```
(1257,)
```

In [9]:

```
print(y_train.shape)
```

```
(1257,)
```

In [10]:

```
print(y_test.shape)
```

```
(540,)
```

In [11]:

```
print(y_test.shape)
```

```
(540,)
```

In [12]:

```
from sklearn.linear_model import LogisticRegression
logisticRegr=LogisticRegression(max_iter=10000)
logisticRegr.fit(x_test,y_test)
```

Out[12]:

```
▾        LogisticRegression
LogisticRegression(max_iter=10000)
```

In [14]:

```
print(logisticRegr.predict(x_test))
```

```
[5 2 5 5 5 3 2 2 3 5 9 8 7 2 1 1 5 7 0 4 5 9 0 2 1 3 4 7 5 2 1 1 2 9 8 1 4
 5 4 9 5 5 6 0 1 7 2 9 7 7 3 9 5 1 6 6 7 7 8 1 6 1 3 6 1 3 2 6 8 1 4 7 1 6
 0 0 5 1 3 5 1 6 4 0 4 7 5 7 8 3 7 8 5 1 1 7 5 9 7 9 3 0 7 8 7 4 8 3 2 8 5
 2 7 4 4 8 9 7 4 5 0 5 9 8 2 3 2 4 4 8 0 5 2 9 4 8 6 5 9 7 8 0 9 4 3 8 7 5
 5 3 3 5 1 0 8 7 2 8 4 1 0 0 3 6 4 7 7 0 4 9 2 8 7 9 7 2 0 3 3 8 5 7 5 6 8
 4 1 5 1 1 6 9 9 9 8 6 4 6 0 1 6 5 3 9 0 2 7 8 8 7 3 8 3 9 3 0 9 6 0 4 0 3
 5 0 4 3 5 8 8 9 2 5 0 8 3 7 4 3 7 9 2 6 1 2 1 7 0 7 5 0 6 4 1 8 3 0 8 9 2
 2 5 2 6 6 3 4 0 7 1 5 3 8 7 3 4 2 5 1 3 0 0 9 3 8 8 3 9 8 6 6 2 6 7 5 3 1
 5 7 5 4 5 2 6 2 0 6 0 7 2 5 8 8 7 1 4 7 2 0 0 3 7 4 2 5 2 6 1 0 5 7 2 9 1
 6 9 6 5 4 8 8 0 9 3 5 7 1 2 4 1 6 6 2 1 5 8 0 3 2 4 3 9 0 0 3 2 8 9 0 2 5
 2 2 5 8 3 6 7 2 5 9 0 5 9 0 8 3 9 1 4 9 2 1 5 6 0 0 9 6 7 9 9 3 1 4 3 4 7
 2 5 2 6 0 3 4 9 3 1 0 3 7 6 0 7 1 0 5 9 3 5 7 7 0 1 1 5 1 1 1 6 7 7 2 2 2
 7 1 3 6 9 1 7 5 6 7 2 8 3 0 0 3 8 7 6 9 3 0 1 7 8 4 8 2 9 0 9 2 4 9 0 7 0
 4 8 5 7 4 3 6 6 8 7 9 3 2 6 0 3 5 3 6 2 3 6 5 3 0 1 5 2 5 4 2 1 3 1 0 9 6
 4 5 9 1 8 0 8 3 0 2 2 2 9 9 3 8 6 2 4 6 5 5]
```

In [16]:

```
score=logisticRegr.score(x_test,y_test)
print(score)
```

```
1.0
```

In [ ]:

```

```