In [3]:
```
import pandas as pd
```

In [4]:
```
import numpy as np
```

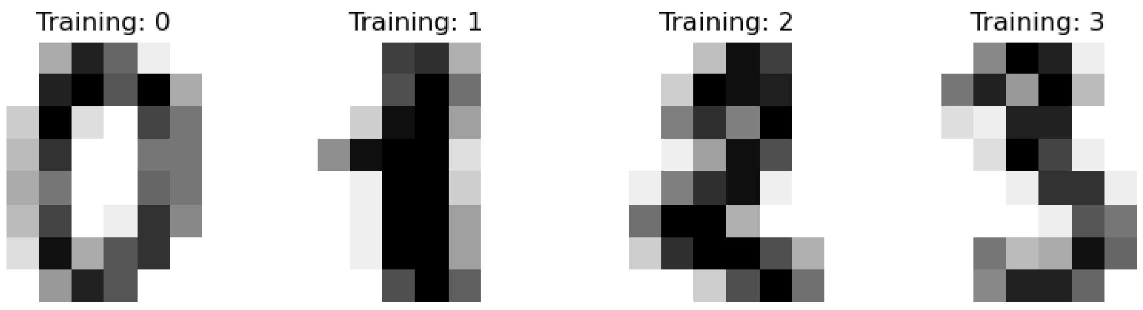
In [5]:
```
import matplotlib.pyplot as plt
```

In [6]:
```
# import Data

from sklearn.datasets import load_digits
```

In [7]:
```
df = load_digits()
```

In [8]:
```
_, axes = plt.subplots(nrows=1, ncols=4, figsize=(10, 3))
for ax, image, label in zip(axes,df.images, df.target):
  ax.set_axis_off()
  ax.imshow(image, cmap=plt.cm.gray_r, interpolation="nearest")
  ax.set_title("Training: %i" % label)
```



In [9]:
```
# Data Preprocessing

df.images.shape
```

Out[9]:
```
(1797, 8, 8)
```

In [10]:
```
df.images[0]
```

Out[10]:
```
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 [11]:
```
df.images[0].shape
```

Out[11]:
```
(8, 8)
```

In [12]:
```
len(df.images)
```

Out[12]:
```
1797
```

In [13]:
```
n_samples = len(df.images)
data = df.images.reshape((n_samples, -1))
```

In [14]:
```
data[0]
```

```
Out[14]: 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 [15]: data[0].shape
```

```
Out[15]: (64,)
```

```
In [16]: data.shape
```

```
Out[16]: (1797, 64)
```

```
In [17]: #Scaling Images Data
         data.min()
```

```
Out[17]: 0.0
```

```
In [18]: data.max()
```

```
Out[18]: 16.0
```

```
In [22]: data=data/16
```

```
In [21]: data.min()
```

```
Out[21]: 0.0
```

```
In [23]: data.max()
```

```
Out[23]: 0.00390625
```

```
In [24]: data[0]
```

```
Out[24]: array([0.        , 0.        , 0.0012207 , 0.00317383, 0.00219727,
                0.00024414, 0.        , 0.        , 0.        , 0.        ,
                0.00317383, 0.00366211, 0.00244141, 0.00366211, 0.0012207 ,
                0.        , 0.        , 0.00073242, 0.00366211, 0.00048828,
                0.        , 0.00268555, 0.00195312, 0.        , 0.        ,
                0.00097656, 0.00292969, 0.        , 0.        , 0.00195312,
                0.00195312, 0.        , 0.        , 0.0012207 , 0.00195312,
                0.        , 0.        , 0.00219727, 0.00195312, 0.        ,
                0.        , 0.00097656, 0.00268555, 0.        , 0.00024414,
                0.00292969, 0.00170898, 0.        , 0.        , 0.00048828,
                0.00341797, 0.0012207 , 0.00244141, 0.00292969, 0.        ,
                0.        , 0.        , 0.        , 0.00146484, 0.00317383,
                0.00244141, 0.        , 0.        , 0.        ])
```

```
In [25]: #Train Test Split Data

         from sklearn.model_selection import train_test_split
```

```
In [26]: X_train, X_test, y_train, y_test = train_test_split(data, df.target, test_size=0.3)
```

```
In [27]: X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
Out[27]: ((1257, 64), (540, 64), (1257,), (540,))
```

In [28]:
```
#Random Forest Model

from sklearn.ensemble import RandomForestClassifier
```

In [29]:
```
rf = RandomForestClassifier()
```

In [30]:
```
rf.fit(X_train, y_train)
```

Out[30]:
```
▾ RandomForestClassifier
RandomForestClassifier()
```

In [31]:
```
y_pred = rf.predict(X_test)
```

In [32]:
```
y_pred
```

Out[32]:
```
array([5, 7, 9, 4, 6, 6, 3, 3, 4, 9, 2, 4, 4, 6, 4, 9, 1, 2, 4, 3, 5, 9,
       1, 0, 0, 0, 5, 0, 1, 0, 5, 4, 9, 8, 6, 8, 2, 7, 5, 4, 7, 6, 8, 7,
       6, 4, 0, 5, 4, 2, 3, 2, 2, 4, 0, 2, 5, 5, 9, 2, 3, 9, 4, 3, 8, 3,
       9, 1, 1, 7, 2, 3, 4, 6, 3, 1, 2, 2, 8, 0, 4, 7, 4, 5, 1, 3, 7, 1,
       7, 6, 0, 2, 5, 8, 2, 7, 7, 1, 9, 2, 6, 3, 9, 4, 1, 2, 8, 7, 2, 5,
       3, 1, 3, 6, 8, 8, 3, 7, 8, 9, 9, 6, 7, 4, 8, 9, 6, 3, 8, 7, 4, 3,
       1, 2, 9, 1, 8, 1, 4, 1, 3, 4, 3, 6, 5, 1, 0, 7, 2, 7, 5, 9, 4, 1,
       7, 8, 5, 4, 7, 3, 4, 4, 6, 7, 1, 0, 1, 3, 1, 7, 6, 6, 9, 7, 7, 9,
       5, 6, 6, 3, 5, 0, 6, 0, 8, 4, 4, 7, 9, 3, 3, 8, 6, 2, 4, 1, 2, 1,
       7, 4, 0, 6, 1, 0, 7, 7, 5, 8, 9, 2, 7, 0, 3, 3, 2, 5, 6, 7, 0, 4,
       9, 8, 0, 0, 2, 2, 8, 6, 1, 3, 9, 9, 3, 5, 0, 1, 5, 3, 1, 1, 1, 2,
       1, 0, 1, 5, 0, 1, 4, 2, 9, 0, 0, 6, 2, 6, 7, 7, 1, 1, 3, 3, 8, 0,
       5, 0, 6, 1, 3, 8, 0, 4, 5, 8, 2, 1, 6, 5, 2, 0, 8, 5, 0, 7, 1, 8,
       1, 8, 6, 2, 0, 5, 2, 6, 3, 7, 8, 4, 6, 4, 0, 0, 3, 9, 8, 7, 6, 1,
       8, 5, 0, 3, 3, 1, 0, 2, 3, 4, 9, 9, 4, 9, 2, 8, 9, 2, 6, 8, 6, 3,
       4, 9, 0, 3, 5, 0, 9, 6, 0, 7, 1, 6, 7, 2, 8, 2, 9, 4, 6, 7, 2, 1,
       7, 0, 2, 7, 4, 1, 9, 1, 2, 4, 2, 3, 1, 7, 1, 5, 7, 4, 6, 7, 2, 8,
       8, 9, 9, 3, 3, 6, 0, 2, 2, 7, 3, 9, 0, 3, 7, 2, 9, 2, 5, 3, 8, 0,
       1, 7, 4, 6, 4, 0, 6, 6, 5, 0, 9, 3, 9, 1, 5, 2, 3, 1, 6, 4, 3, 3,
       2, 7, 4, 0, 5, 5, 3, 2, 6, 3, 9, 0, 3, 4, 8, 2, 6, 9, 2, 0, 0, 5,
       5, 4, 6, 9, 4, 5, 6, 3, 3, 9, 9, 8, 3, 3, 0, 0, 5, 6, 6, 9, 3, 2,
       8, 4, 0, 0, 7, 7, 0, 6, 9, 5, 4, 4, 3, 7, 5, 9, 8, 2, 3, 7, 2, 1,
       7, 3, 6, 7, 9, 1, 8, 8, 0, 0, 9, 0, 0, 0, 1, 2, 5, 8, 3, 0, 0, 2,
       2, 8, 5, 8, 3, 1, 6, 1, 0, 7, 8, 5, 7, 9, 7, 8, 1, 0, 8, 5, 4, 8,
       5, 0, 0, 2, 2, 5, 4, 1, 0, 4, 0, 9])
```

In [33]:
```
from sklearn.metrics import confusion_matrix, classification_report
```

In [34]:
```
confusion_matrix(y_test, y_pred)
```

Out[34]:
```
array([[65,  0,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0, 52,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0, 58,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  0, 59,  0,  0,  0,  0,  0,  1],
       [ 0,  0,  0,  0, 51,  0,  0,  2,  1,  0],
       [ 0,  0,  0,  0,  0, 46,  1,  0,  0,  1],
       [ 0,  1,  0,  0,  1,  0, 50,  0,  0,  0],
       [ 0,  0,  0,  0,  0,  0,  0, 50,  0,  0],
       [ 0,  2,  0,  0,  1,  0,  0,  1, 46,  0],
       [ 0,  0,  0,  2,  0,  0,  0,  1,  0, 48]], dtype=int64)
```

In [35]:
```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        65
           1       0.95      1.00      0.97        52
           2       1.00      1.00      1.00        58
           3       0.97      0.98      0.98        60
           4       0.96      0.94      0.95        54
           5       1.00      0.96      0.98        48
           6       0.98      0.96      0.97        52
           7       0.93      1.00      0.96        50
           8       0.98      0.92      0.95        50
           9       0.96      0.94      0.95        51

    accuracy                           0.97       540
   macro avg       0.97      0.97      0.97       540
weighted avg       0.97      0.97      0.97       540
```

```
In [ ]:
```