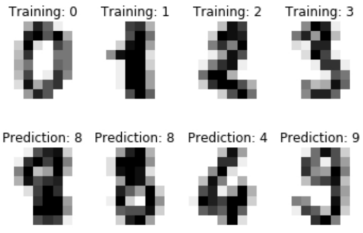
In [4]:
```
# Standat Python importlarimizi yaptik
import matplotlib.pyplot as plt

# Datasetimi importladim
from sklearn import datasets, svm, metrics

# digits degiskenine datasetimi atadim
digits = datasets.load_digits()

# Ilgilendigimiz data 8x8lik formatta
# Once ilk 4 resme bakalim
# Eger imagefile uzerinden calisacaksak oncelikle matplotlib.pyplot.imread. komutunu cagirmaliyiz
# Butun resimlerin ayni formatta olduguna dikkat ediyorum
# Hangi resmin hangi sayiyi temsil ettigini bildigim degiskenlere target ismini atadim
images_and_labels = list(zip(digits.images, digits.target))
for index, (image, label) in enumerate(images_and_labels[:4]):
    plt.subplot(2, 4, index + 1)
    plt.axis('off')
    plt.imshow(image, cmap=plt.cm.gray_r, interpolation='nearest')
    plt.title('Training: %i' % label)

# Classify uygulamak icin resimleri yassilastiriyorum (flatten image) tek coloumn yapiyorumda denilebilir
# Datayi matrixe cevirdim.
n_samples = len(digits.images)
data = digits.images.reshape((n_samples, -1))

# Classifiera atadim
classifier = svm.SVC(gamma=0.001)

# Digitleri train sete atadim
classifier.fit(data[:n_samples // 2], digits.target[:n_samples // 2])

# Ikinci asamada test datasi tahmin etti
expected = digits.target[n_samples // 2:]
predicted = classifier.predict(data[n_samples // 2:])

print("Classification report for classifier %s:\n%s\n"
      % (classifier, metrics.classification_report(expected, predicted)))
print("Confusion matrix:\n%s" % metrics.confusion_matrix(expected, predicted))

images_and_predictions = list(zip(digits.images[n_samples // 2:], predicted))
for index, (image, prediction) in enumerate(images_and_predictions[:4]):
    plt.subplot(2, 4, index + 5)
    plt.axis('off')
    plt.imshow(image, cmap=plt.cm.gray_r, interpolation='nearest')
    plt.title('Prediction: %i' % prediction)

plt.show()
```

```
Classification report for classifier SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma=0.001, kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False):
              precision    recall  f1-score   support

           0       1.00      0.99      0.99        88
           1       0.99      0.97      0.98        91
           2       0.99      0.99      0.99        86
           3       0.98      0.87      0.92        91
           4       0.99      0.96      0.97        92
           5       0.95      0.97      0.96        91
           6       0.99      0.99      0.99        91
           7       0.96      0.99      0.97        89
           8       0.94      1.00      0.97        88
           9       0.93      0.98      0.95        92

   micro avg       0.97      0.97      0.97       899
   macro avg       0.97      0.97      0.97       899
weighted avg       0.97      0.97      0.97       899


Confusion matrix:
[[87  0  0  0  1  0  0  0  0  0]
 [ 0 88  1  0  0  0  0  0  1  1]
 [ 0  0 85  1  0  0  0  0  0  0]
 [ 0  0  0 79  0  3  0  4  5  0]
 [ 0  0  0  0 88  0  0  0  0  4]
 [ 0  0  0  0  0 88  1  0  0  2]
 [ 0  1  0  0  0  0 90  0  0  0]
 [ 0  0  0  0  0  1  0 88  0  0]
 [ 0  0  0  0  0  0  0  0 88  0]
 [ 0  0  0  1  0  1  0  0  0 90]]
```



In [ ]: