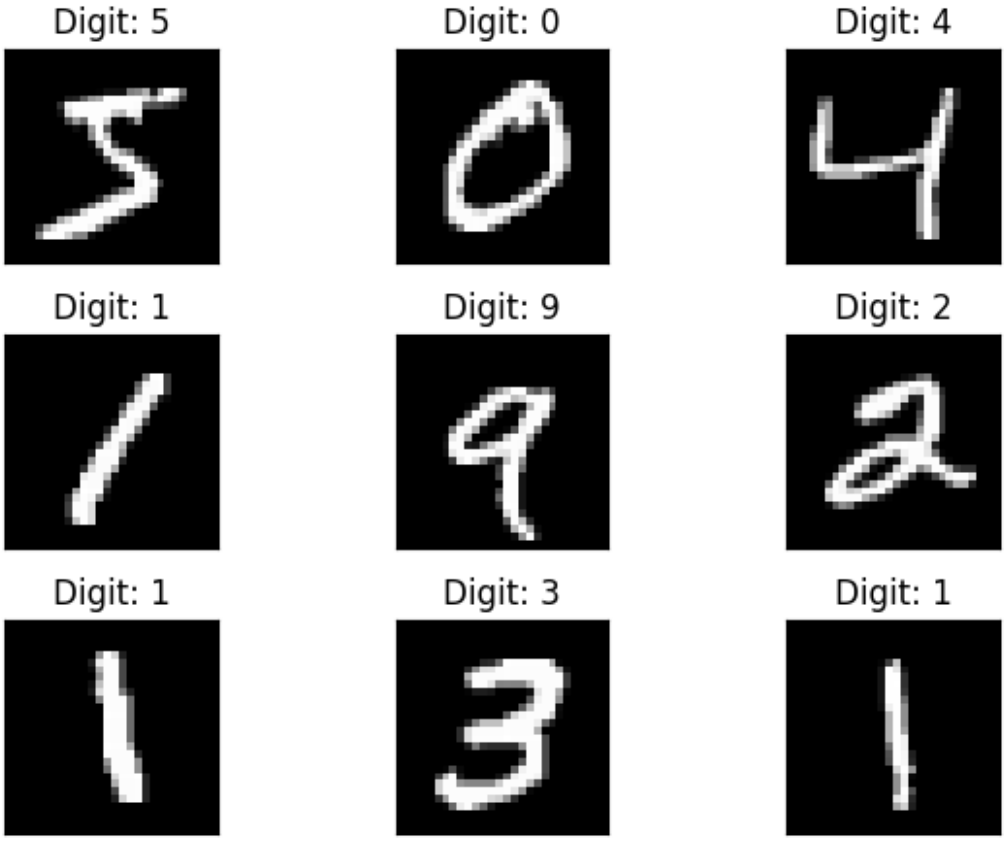
```
# 1. Preparing the data
import keras
from keras.datasets import mnist
import matplotlib.pyplot as plt
from keras import backend as k
import os

# Load mnist dataset
(X_train, y_train), (X_test, y_test) = mnist.load_data()


print("--- Visualizing Data (Close plot window to continue) ---")
fig = plt.figure()
for i in range(9):
    plt.subplot(3, 3, i + 1)
    plt.tight_layout()
    plt.imshow(X_train[i], cmap='gray', interpolation='none')
    plt.title("Digit: {}".format(y_train[i]))
    plt.xticks([])
    plt.yticks([])

plt.show()
print("------------------------------------------------------\n")

img_rows, img_cols = 28, 28


if k.image_data_format() == 'channels_first':
    X_train = X_train.reshape(X_train.shape[0], 1, img_rows, img_cols)
    X_test = X_test.reshape(X_test.shape[0], 1, img_rows, img_cols)
    input_shape = (1, img_rows, img_cols)
else:
    X_train = X_train.reshape(X_train.shape[0], img_rows, img_cols, 1)
    X_test = X_test.reshape(X_test.shape[0], img_rows, img_cols, 1)
    input_shape = (img_rows, img_cols, 1)

# more reshaping
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
X_train /= 255
X_test /= 255
print('--- Data Preprocessing ---')
print('X_train shape:', X_train.shape) #X_train shape: (60000, 28, 28,
1)
print(X_train.shape[0], 'train samples')
print(X_test.shape[0], 'test samples')
print("--------------------------\n")

num_category = 10
```

```
--- Visualizing Data (Close plot window to continue) ---
```




```
------------------------------------------------------------

--- Data Preprocessing ---
X_train shape: (60000, 28, 28, 1)
60000 train samples
10000 test samples
---------------------------
```