In [4]:

```
# Importing Packages
import numpy as np
from matplotlib import pyplot
import keras
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Flatten, Dense
import matplotlib.pyplot as plt
%matplotlib inline
```

In [5]:

```
(X_train, y_train), (X_test, y_test) = tf.keras.datasets.fashion_mnist.load_data()
X_train.shape, y_train.shape, "____" , X_test.shape, y_test.shape
((60000, 28, 28), (60000,), '____', (10000, 28, 28), (10000,))
X_train[0]
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/
train-labels-idx1-ubyte.gz
29515/29515 [==============================] - 0s 1us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/
train-images-idx3-ubyte.gz
26421880/26421880 [==============================] - 5s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/
t10k-labels-idx1-ubyte.gz
5148/5148 [==============================] - 0s 0s/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/
t10k-images-idx3-ubyte.gz
4422102/4422102 [==============================] - 1s 0us/step
```

```
        221, 220, 236, 225, 216, 199, 206, 186, 181, 177, 172, 181, 205,
        206, 115],
       [  0, 122, 219, 193, 179, 171, 183, 196, 204, 210, 213, 207, 211,
        210, 200, 196, 194, 191, 195, 191, 198, 192, 176, 156, 167, 177,
        210,  92],
       [  0,   0,  74, 189, 212, 191, 175, 172, 175, 181, 185, 188, 189,
        188, 193, 198, 204, 209, 210, 210, 211, 188, 188, 194, 192, 216,
        170,   0],
       [  2,   0,   0,   0,  66, 200, 222, 237, 239, 242, 246, 243, 244,
        221, 220, 193, 191, 179, 182, 182, 181, 176, 166, 168,  99,  58,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  40,  61,  44,  72,  41,  35,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0]], dtype=uint8)
```

In [6]:

```
class_labels = ["Trouser", "Pullover" , "Dress" , "Coat" , "Sandal" , "Shirt" , "Sr
class_labels
```

Out[6]:

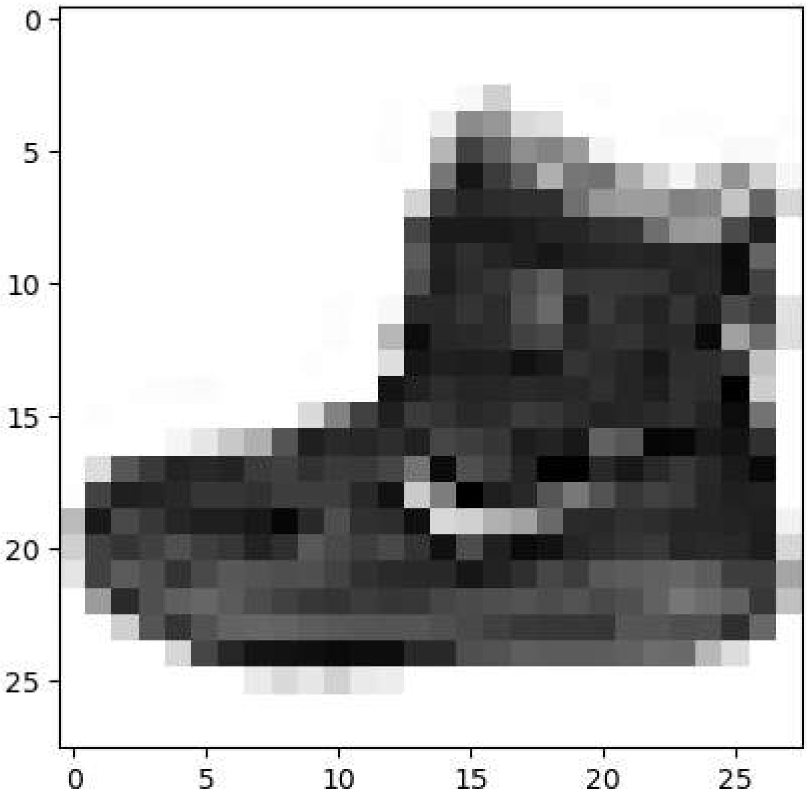
```
['Trouser', 'Pullover', 'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker']
```

In [7]:

```
plt.imshow(X_train[0] , cmap="Greys")
```

Out[7]:

```
<matplotlib.image.AxesImage at 0x18695b40990>
```



In [8]:

```
X_train.ndim
```

Out[8]:

```
3
```

In [9]:

```
X_train = np.expand_dims(X_train, -1)
X_test =np.expand_dims(X_test, -1)
```

In [10]:
```
X_train.ndim
```

Out[10]:
```
4
```

In [11]:
```
X_train =X_train/255
X_test = X_test/255
```

In [12]:
```
from sklearn.model_selection import train_test_split
X_train, X_validation, y_train, y_validation = train_test_split(X_train, y_train)
```

In [13]:
```
X_train.shape, y_train.shape, X_validation.shape, y_validation.shape
```

Out[13]:
```
((45000, 28, 28, 1), (45000,), (15000, 28, 28, 1), (15000,))
```

In [14]:
```
cnn = keras.models.Sequential([

    tf.keras.layers.Conv2D(filters=32, kernel_size=3,strides=(1,1),padding='valid',
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Conv2D(filters=64, kernel_size=3,strides=(2,2),padding='same',
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.25),
    tf.keras.layers.Dense(256, activation='relu'),
    tf.keras.layers.Dropout(0.25),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(10, activation='softmax')

])
```

In [15]:
```
cnn.compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=['accur
```

In [16]:
```
cnn.fit(X_train, y_train, epochs=20 , batch_size=16 , verbose=1 , validation_data=(
313/313 [==============================] - 3s 9ms/step
```

In [18]:

```
cnn.evaluate(X_test, y_test)
```

```
313/313 [==============================] - 3s 8ms/step - loss: 0.3499 - accuracy:
0.8961
```

Out[18]:

```
[0.3499019145965576, 0.8960999846458435]
```

In [ ]:

```
plt.figure(figs
```

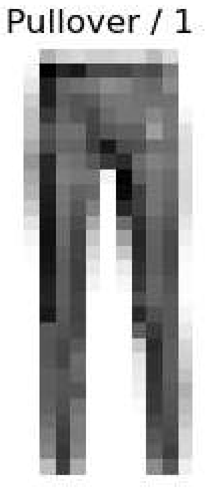
In [ ]:

In [20]:

```
ize=(16,16))
j=1
for i in np.random.randint(0,1000,25):
 plt.subplot(5,5,j)
j+=1
plt.imshow(X_train[i],cmap="Greys")
plt.axis('off')
plt.title('{} / {}'.format(class_labels[y_train[i]], y_train[i]))
```

Out[20]:

```
Text(0.5, 1.0, 'Pullover / 1')
```

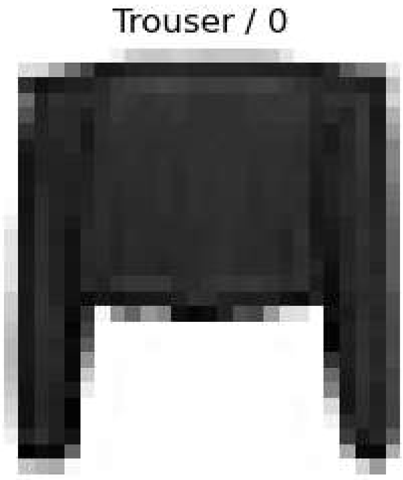


In [21]:

```
plt.figure(figsize=(16,16))
j=1
for i in np.random.randint(0,1000,25):
 plt.subplot(5,5,j)
j+=1
plt.imshow(X_train[i],cmap="Greys")
plt.axis('off')
plt.title('{} / {}'.format(class_labels[y_train[i]], y_train[i]))
```

Out[21]:

```
Text(0.5, 1.0, 'Trouser / 0')
```

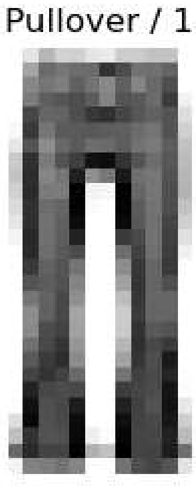


In [32]:

```
plt.figure(figsize=(16,16))
j=1
for i in np.random.randint(0,1000,25):
 plt.subplot(5,5,j)
j+=1
plt.imshow(X_train[i],cmap="Greys")
plt.axis('off')
plt.title('{} / {}'.format(class_labels[y_train[i]], y_train[i]))
```

Out[32]:

```
Text(0.5, 1.0, 'Pullover / 1')
```



In [ ]: