Lab 2. Dimensionality Reduction

We are going to learn dimensioanlity redcution using `scikit-learn` (https://scikit-learn.org/stable/) python package by comparing PCA with t-SNE. Our dataset is the handwritten digits provided by the same package.

In [1]:
```
import numpy as np
from sklearn.datasets import load_digits
from sklearn.decomposition import PCA
from sklearn.preprocessing import scale
from sklearn.manifold import TSNE
from utils import plot_images, plot_2D_samples
import time
```

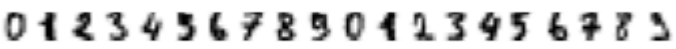
Step 1. Load Data

we will load and scale the data before applying the PCA decomposition. The handwritten digits dataset can be loaded by `load_digits` . More information about the dataset is available here (https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html).

In [2]:
```
X, y = load_digits(return_X_y=True)

print(np.shape(X))
plot_images(X)
print(y[0:20])
```

```
(1797, 64)
```



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

Preprocessing

For PCA, preprocessing is a must. Data should be normalized to have zero mean and unit variance. The `scale` function in `sklearn.preprocessing` provides the z-score normalization with that purpose.

In [3]:
```
X = scale(X)
print(np.shape(X))
plot_images(X)
```

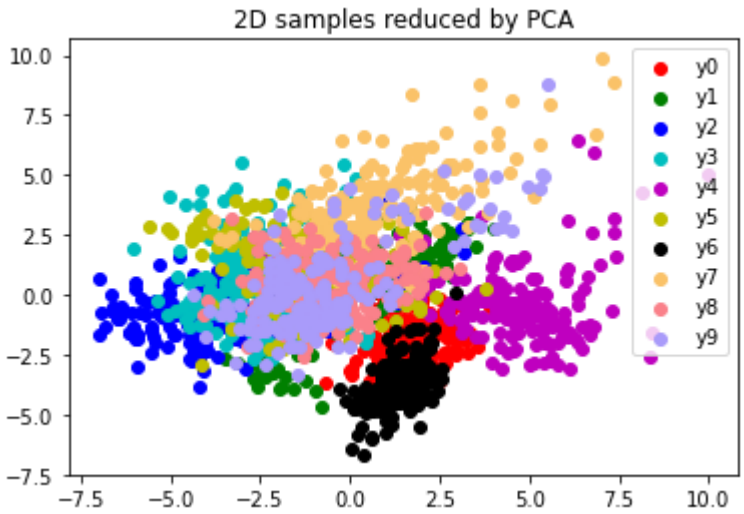
```
(1797, 64)
```

Step 2. Reducing Dimensions by PCA

In [4]:
```
tic = time.time()
X_reduced_pca = PCA(n_components=2).fit_transform(X)
tac = time.time()

print(np.shape(X_reduced_pca))
print("PCA processing time: %s sec." % str(tac - tic))
plot_2D_samples(X_reduced_pca, y, "2D samples reduced by PCA")
```

```
(1797, 2)
PCA processing time: 0.03932976722717285 sec.
```



Step 3. Reducing Dimension by t-SNE

In [5]:
```
tic = time.time()
X_reduced_tsne = TSNE(n_components=2).fit_transform(X)
tac = time.time()

print(np.shape(X_reduced_tsne))
print("t-SNE processing time: %s sec." % str(tac - tic))
plot_2D_samples(X_reduced_tsne, y, "2D samples reduced by t-SNE")
```

```
(1797, 2)
t-SNE processing time: 6.9511497020721436 sec.
```