1、SimpleModel：這是模型的類名，代表了這個程式的主要功能——定義一個簡單的神經網絡模型。模型的架構相對較簡單，只包含兩個全連接層，並且適用於處理 MNIST 數據集的分類任務。

程式碼：

# 定義模型／模型架構修改

class SimpleModel(nn.Module):

def \_\_init\_\_(self):

super(SimpleModel, self).\_\_init\_\_()

self.fc1 = nn.Linear(784, 128)

self.fc2 = nn.Linear(128, 10)

def forward(self, x):

x = torch.flatten(x, start\_dim=1)

x = self.fc1(x)

x = torch.relu(x)

x = self.fc2(x)

return x

2、MNIST：這個程式主要是用來處理 MNIST 數據集的，包括下載和加載數據、定義模型、訓練和測試模型等。因此，將 "MNIST" 包含在程式名稱中可以清楚地表明這個程式的用途。

程式碼：

import torchvision

import torchvision.transforms as transforms

# 下載和加載　MNIST　數據集

transform = transforms.Compose([

transforms.ToTensor(),

transforms.Normalize((0.5,), (0.5,))

])

trainset = torchvision.datasets.MNIST(root='./data', train=True, download=True, transform=transform)

trainloader = torch.utils.data.DataLoader(trainset, batch\_size=64, shuffle=True)

testset = torchvision.datasets.MNIST(root='./data', train=False, download=True, transform=transform)

testloader = torch.utils.data.DataLoader(testset, batch\_size=64, shuffle=False)

3、Classifier：由於這個程式是用來建立一個分類器模型，它用於分類手寫數字圖像，所以將 "Classifier" 包含在程式名稱中可以明確地表明這個程式的功能。