Explanation

1. **Load the MNIST dataset and pre-process the data**:
    - Load the MNIST dataset using a library such as TensorFlow or PyTorch
    - Normalize the pixel values to be in the range [0, 1]
    - Split the data into training and testing sets

2. **Define the MLP model:**
    - Initialize an empty sequential model using a library such as TensorFlow's keras or PyTorch's nn module
    - Add an input layer to flatten the 28x28 images into 1D arrays
    - Add one or more dense hidden layers with a specified number of units and activation function (e.g. ReLU)
    - Add an output layer with 10 units and a softmax activation function

3. **Compile the model:**
    - Specify the optimizer (e.g. Adam or SGD) to use for training
    - Specify the loss function (e.g. cross-entropy loss for multi-class classification)
    - Specify any evaluation metrics to track during training (e.g. accuracy)

4. **Train the model:**
    - Train the model on the training data for a specified number of epochs
    - Optionally, use validation data to evaluate the model during training and early stop if the performance on the validation data doesn't improve

5. **Evaluate the model:**
    - Evaluate the model on the test data and report the final accuracy or other evaluation metrics

6. **Make predictions:**
    - Use the trained model to make predictions on new data and obtain the predicted values
    - Convert the predicted probabilities into predicted class labels by selecting the class with the highest predicted probability