```
    1: import tensorflow as tf
    2: from linear import Linear
    3:
    4: ## Citation: https://colab.research.google.com/github/vsitzmann/siren/blob/master/explore_siren.ipynb#scrollTo=wt0akc6oiJgu
    5:
    6: class Siren(tf.Module):
    7:     def __init__(self, num_inputs, hidden_layer_width, num_outputs, num_hidden_layers):
    8:         ## Initlization for first layer:
    9:         self.firstLayer = Linear(
   10:             num_inputs, hidden_layer_width, first_layer=True, bias=True
   11:         )
   12:
   13:         ## initilization for the other layers:
   14:         self.hiddenLayers = [
   15:             Linear(hidden_layer_width, hidden_layer_width, first_layer=False)
   16:             for _ in range(num_hidden_layers)
   17:         ]
   18:
   19:         ## Final Layer
   20:         self.finalLayer = Linear(
   21:             hidden_layer_width, num_outputs, first_layer=False, bias=True
   22:         )
   23:
   24:     def __call__(self, X):
   25:         X = tf.constant(X)
   26:         X = tf.Variable(X, trainable=True)
   27:
   28:         layer = tf.math.sin(30 * self.firstLayer(X))
   29:         for hidden_layer in self.hiddenLayers:
   30:             layer = tf.math.sin(30 * hidden_layer(layer))
   31:         return (self.finalLayer(layer)), X
```