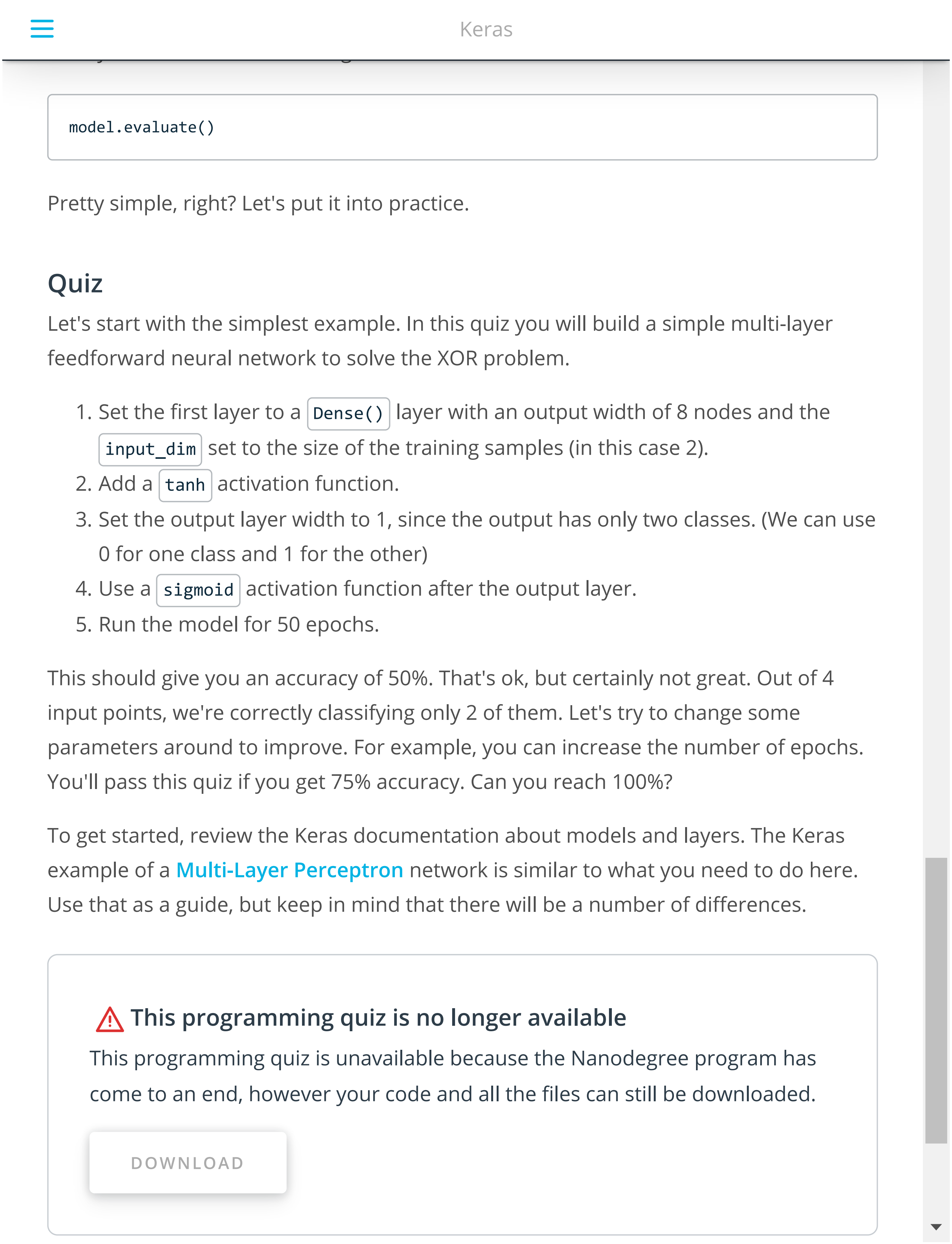

```
model.evaluate()
```

Pretty simple, right? Let's put it into practice.

Quiz

Let's start with the simplest example. In this quiz you will build a simple multi-layer feedforward neural network to solve the XOR problem.

1. Set the first layer to a `Dense()` layer with an output width of 8 nodes and the `input_dim` set to the size of the training samples (in this case 2).
2. Add a `tanh` activation function.
3. Set the output layer width to 1, since the output has only two classes. (We can use 0 for one class and 1 for the other)
4. Use a `sigmoid` activation function after the output layer.
5. Run the model for 50 epochs.

This should give you an accuracy of 50%. That's ok, but certainly not great. Out of 4 input points, we're correctly classifying only 2 of them. Let's try to change some parameters around to improve. For example, you can increase the number of epochs. You'll pass this quiz if you get 75% accuracy. Can you reach 100%?

To get started, review the Keras documentation about models and layers. The Keras example of a Multi-Layer Perceptron network is similar to what you need to do here. Use that as a guide, but keep in mind that there will be a number of differences.

⚠ This programming quiz is no longer available

This programming quiz is unavailable because the Nanodegree program has come to an end, however your code and all the files can still be downloaded.