Sample TensorFlow code

While TensorFlow is written with fast custom C++ code under the hood, it has a high level Python interface that we will use throughout this course. As such, you'll need to be comfortable with basic coding in Python for this course.

For example we might create a custom Neural Network model using code that looks something like the below. Don't worry if you don't know what all of the words mean, we'll walk you through all of these topics later in the course. For now just make sure you know enough Python to understand the following high level takeaways:

1. This is a custom class that extends the `tf.keras.Model` base class
2. It defines four specific kinds of `tf.keras.layers`
3. It has a `call` function which will run an input through those layers (functions)

If that resonates with you then you probably know enough Python to do well in this course!

If you don't feel like you know enough Python yet, don't worry. Both Python.org and LearnPython.org have great resources you can use to get you up to speed. We're sure that with a little work, very soon you'll be ready to proceed with this course.

```python
# Load in the TensorFlow library
import tensorflow as tf

# Define my custom Neural Network model
class MyModel(tf.keras.Model):
    def __init__(self):
        super(MyModel, self).__init__()
        # define Neural Network layer types
        self.conv = tf.keras.layers.Conv2D(32, 3, activation='relu')
        self.flatten = tf.keras.layers.Flatten()
        self.dense1 = tf.keras.layers.Dense(128, activation='relu')
        self.dense2 = tf.keras.layers.Dense(10)

    # run my Neural Network model by evaluating
    # each layer on my input data
    def call(self, x):
        x = self.conv(x)
        x = self.flatten(x)
        x = self.dense1(x)
        x = self.dense2(x)
        return x

# Create an instance of the model
model = MyModel()
```