**Lesson outline**

* Object-oriented programming syntax
  + Procedural vs. object-oriented programming
  + Classes, objects, methods and attributes
  + Coding a class
  + Magic methods
  + Inheritance
* Using object-oriented programming to make a Python package
  + Making a package
  + Tour of scikit-learn source code
  + Putting your package on PyPi

**Why object-oriented programming?**

Object-oriented programming has a few benefits over procedural programming, which is the programming style you most likely first learned. As you'll see in this lesson:

* Object-oriented programming allows you to create large, modular programs that can easily expand over time.
* Object-oriented programs hide the implementation from the end user.

Consider Python packages like [Scikit-learn](https://github.com/scikit-learn/scikit-learn), [pandas](https://pandas.pydata.org/), and [NumPy](http://www.numpy.org/). These are all Python packages built with object-oriented programming. Scikit-learn, for example, is a relatively large and complex package built with object-oriented programming. This package has expanded over the years with new functionality and new algorithms.

When you train a machine learning algorithm with Scikit-learn, you don't have to know anything about how the algorithms work or how they were coded. You can focus directly on the modeling.

Here's an example taken from the [Scikit-learn website](http://scikit-learn.org/stable/modules/svm.html):

from sklearn import svm

X = [[0, 0], [1, 1]]

y = [0, 1]

clf = svm.SVC()

clf.fit(X, y)

How does Scikit-learn train the SVM model? You don't need to know because the implementation is hidden with object-oriented programming. If the implementation changes, you (as a user of Scikit-learn) might not ever find out. Whether or not you *should* understand how SVM works is a different question.

In this lesson, you'll practice the fundamentals of object-oriented programming. By the end of the lesson, you'll have built a Python package using object-oriented programming.

**Lesson files**

This lesson uses classroom workspaces that contain all of the files and functionality you need. You can also find the files in the [data scientist nanodegree term 2 GitHub repo](https://github.com/udacity/DSND_Term2/tree/master/lessons/ObjectOrientedProgramming).