Getting Started with pyflakes

WE'LL COVER THE FOLLOWING

The **pyflakes** project is a part of something known as the Divmod Project. Pyflakes doesn't actually execute the code it checks much like pylint doesn't execute the code it analyzes. You can install pyflakes using pip, easy_install or from source.

We will start by running pyflakes against the original version of the same piece of code that we used with pylint. Here it is again:

```
import sys

class CarClass:
    """"""

    def __init__(self, color, make, model, year):
        """Constructor"""
        self.color = color
        self.make = make
        self.model = model
        self.year = year

        if "Windows" in platform.platform():
            print("You're using Windows!")

        self.weight = self.getWeight(1, 2, 3)

    def getWeight(this):
        """"""
        return "2000 lbs"
```

As was noted in the previous section, this silly piece of code has 4 issues, 3 of which would stop the program from running. Let's see what pyflakes can find! Try running the following command and you'll see the following output:

```
c:\py101>c:\Python34\Scripts\pyflakes.exe crummy_code.py
crummy_code.py:1: 'sys' imported but unused
crummy_code.py:13: undefined name 'platform'
```

While pyflakes was super fast at returning this output, it didn’t find all the errors. The **getWeight** method call is passing too many arguments and the **getWeight** method itself is defined incorrectly as it doesn’t have a **self** argument. Well, you can actually call the first argument anything you want, but by convention it’s usually called **self**. If you fixed your code according to what pyflakes told you, your code still wouldn’t work.

Wrapping Up

The next step would be to try running pylint and pyflakes against some of your own code or against a Python package like SQLAlchemy and seeing what you get for output. You can learn a lot about your own code using these tools. pylint is integrated with many popular Python IDEs, such as Wingware, Editra, and PyDev. You may find some of the warnings from pylint to be annoying or not even really applicable. There are ways to suppress such things as deprecation warnings through command line options. Or you can use the **-generate-rcfile** to create an example config file that will help you control pylint. Note that pylint and pyflakes does not import your code, so you don’t have to worry about undesirable side effects.