Context manager in Python

Python came out with a special new keyword several years ago in Python 2.5 that is known as the **with statement**. This new keyword allows a developer to create context managers. But wait! What's a context manager? They are handy constructs that allow you to set something up and tear something down automatically. For example, you might want to open a file, write a bunch of stuff to it and then close it. This is probably the classic example of a context manager. In fact, Python creates one automatically for you when you open a file using the **with statement**:

```
with open(path, 'w') as f_obj:
    f_obj.write(some_data)
```

Back in Python 2.4, you would have to do it the old fashioned way:

```
f_obj = open(path, 'w')
f_obj.write(some_data)
f_obj.close()
```

The way this works under the covers is by using some of Python's magic methods: **__enter__** and **__exit__**. Let's try creating our own context manager to demonstrate how this all works!