contextlib.suppress(*exceptions)

Another handy little tool is the **suppress** class which was added in Python 3.4. The idea behind this context manager utility is that it can suppress any number of exceptions. Let's say we want to ignore the **FileNotFoundError** exception. If you were to write the following context manager, it wouldn't work:

```
with open('fauxfile.txt') as fobj:
    for line in fobj:
        print(line)

#Traceback (most recent call last):
#   File "/usercode/__ed_file.py", line 1, in <module>
# with open('fauxfile.txt') as fobj:
#FileNotFoundError: [Errno 2] No such file or directory: 'fauxfile.txt''
```

As you can see, this context manager doesn't handle this exception. If you want to ignore this error, then you can do the following:

```
from contextlib import suppress

with suppress(FileNotFoundError):
    with open('fauxfile.txt') as fobj:
        for line in fobj:
            print(line)
```

Here we import **suppress** and pass it the exception that we want to ignore, which in this case is the FileNotFoundError exception. If you run this code, you will note that nothing happens as the file does not exist, but an error is also not raised. It should be noted that this context manager is **reentrant**. This will be explained later on in this section.