

## contextlib.closing(thing)

The contextlib module comes with some other handy utilities. The first one is the **closing** class which will close the thing upon the completion of code block. The Python documentation gives an example that's similar to the following one:

```
from contextlib import contextmanager

@contextmanager
def closing(db):
    try:
        yield db.conn()
    finally:
        db.close()
```



Basically what we're doing is creating a closing function that's wrapped in a contextmanager. This is the equivalent of what the closing class does. The difference is that instead of a decorator, we can use the **closing** class itself in our with statement. Let's take a look:

```
from contextlib import closing
from urllib.request import urlopen

with closing(urlopen('http://www.google.com')) as webpage:
    for line in webpage:
        # process the line
        pass
```



In this example, we open a url page but wrap it with our closing class. This will cause the handle to the web page to be closed once we fall out of the with statement's code block.