Practical Python Programming

A course by @dabeaz

Contents | Previous (4.3 Special methods) | Next (5 Object Model)

4.4 Defining Exceptions

User defined exceptions are defined by classes.

```
class NetworkError(Exception):
    pass
```

Exceptions always inherit from `Exception`.

Usually they are empty classes. Use `pass` for the body.

You can also make a hierarchy of your exceptions.

```
class AuthenticationError(NetworkError):
     pass

class ProtocolError(NetworkError):
    pass
```

Exercises

Exercise 4.11: Defining a custom exception

It is often good practice for libraries to define their own exceptions.

This makes it easier to distinguish between Python exceptions raised in response to common programming errors versus exceptions intentionally raised by a library to a signal a specific usage problem.

Modify the `create_formatter()` function from the last exercise so that it raises a custom `FormatError` exception when the user provides a bad format name.

For example:

```
>>> from tableformat import create_formatter
>>> formatter = create_formatter('xls')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "tableformat.py", line 71, in create_formatter
    raise FormatError('Unknown table format %s' % name)
FormatError: Unknown table format xls
>>>
```

Contents | Previous (4.3 Special methods) | Next (5 Object Model)