CHAPTER 4

The Power of Introspection

This chapter covers one of Python's strengths: introspection. As you know, everything in Python is an object, and introspection is code looking at other modules and functions in memory as objects, getting information about them, and manipulating them. Along the way, you'll define functions with no name, call functions with arguments out of order, and reference functions whose names you don't even know ahead of time.

Diving In

Listing 4-1 is a complete, working Python program. You should understand a good deal about it just by looking at it. The numbered lines illustrate concepts covered in the previous chapters. Don't worry if the rest of the code looks intimidating; you'll learn all about it in this chapter.

NOTE You can download all of the examples presented in this book from the Downloads section of the Apress web site (http://www.apress.com).

Listing 4-1. apihelper.py

def info(object, spacing=10, collapse=1):
    '''Print methods and doc strings."
    Takes module, class, list, dictionary, or string.""
    methodList = [method for method in dir(object) if callable(getattr(object, method))]
    processFunc = collapse and (lambda s: " ".join(s.split())) or (lambda s: s)
    print "\n".join(["%s %s" %
                (method.ljust(spacing),
                 processFunc(str(getattr(object, method).__doc__)))
               for method in methodList])

if __name__ == "__main__":
    print info.__doc__

M. Pilgrim, Dive Into Python
© Mark Pilgrim 2004
This module has one function, `info`. According to its function declaration, it takes three parameters: `object`, `spacing`, and `collapse`. The last two are actually optional parameters, as you'll see shortly.

The `info` function has a multiline doc string that succinctly describes the function's purpose. Note that no return value is mentioned; this function will be used solely for its effects, rather than its value.

Code within the function is indented.

The `if __name__` trick allows this program do something useful when run by itself, without interfering with its use as a module for other programs. In this case, the program simply prints the doc string of the `info` function.

The `info` function is designed to be used by you, the programmer, while working in the Python IDE. It takes any object that has functions or methods (like a module, which has functions, or a list, which has methods) and prints the functions and their doc strings. Listing 4-2 shows an example of using the `info` function.

### Listing 4-2. Sample Use of apihelper.py

```python
>>> from apihelper import info
>>> li = []
>>> info(li)
append  L.append(object) -- append object to end
count   L.count(value) -> integer -- return number of occurrences of value
extend  L.extend(list) -- extend list by appending list elements
index   L.index(value) -> integer -- return index of first occurrence of value
insert  L.insert(index, object) -- insert object before index
pop     L.pop([index]) -> item -- remove and return item at index (default last)
remove  L.remove(value) -- remove first occurrence of value
reverse L.reverse() -- reverse *IN PLACE*
sort    L.sort([cmpfunc]) -- sort *IN PLACE*; if given, cmpfunc(x, y) -> -1, 0, 1
```

By default, the output is formatted to be easy to read. Multiline doc strings are collapsed into a single long line, but this option can be changed by specifying 0 for the `collapse` argument. If the function names are longer than ten characters, you can specify a larger value for the `spacing` argument to make the output easier to read, as demonstrated in Listing 4-3.

### Listing 4-3. Advanced Use of apihelper.py

```python
>>> import odbchelper
>>> info(odbchelper)
buildConnectionString Build a connection string from a dictionary Returns string.
```

By default, the output is formatted to be easy to read. Multiline doc strings are collapsed into a single long line, but this option can be changed by specifying 0 for the `collapse` argument. If the function names are longer than ten characters, you can specify a larger value for the `spacing` argument to make the output easier to read, as demonstrated in Listing 4-3.