

## 8.20 通过字符串调用对象方法¶

### 问题¶

你有一个字符串形式的方法名称，想通过它调用某个对象的对应方法。

### 解决方案¶

最简单的情况，可以使用 `getattr()`：

```
import math
```

```
class Point:
```

```
    def __init__(self, x, y):
```

```
        self.x = x
```

```
        self.y = y
```

```
    def __repr__(self):
```

```
        return 'Point({!r},{!r})'.format(self.x, self.y)
```

```
    def distance(self, x, y):
```

```
        return math.hypot(self.x - x, self.y - y)
```

```
p = Point(2, 3)
```

```
d = getattr(p, 'distance')(0, 0) # Calls p.distance(0, 0)
```

另外一种方法是使用 `operator.methodcaller()`，例如：

```
import operator
```

```
operator.methodcaller('distance', 0, 0)(p)
```

当你需要通过相同的参数多次调用某个方法时，使用 `operator.methodcaller` 就很方便了。比如你需要排序一系列的点，就可以这样做：

```
points = [
```

```
    Point(1, 2),
```

```
    Point(3, 0),
```

```
    Point(10, -3),
```

```
    Point(-5, -7),
```

```
    Point(-1, 8),
```

```
    Point(3, 2)
```

```
]
```

```
# Sort by distance from origin (0, 0)
```

```
points.sort(key=operator.methodcaller('distance', 0, 0))
```

### 讨论¶

调用一个方法实际上是两部独立操作，第一步是查找属性，第二步是函数调用。因此，为了调用某个方法，你可以首先通过 `getattr()` 来查找到这个属性，然后再去以函数方式调用它即可。

`operator.methodcaller()` 创建一个可调对象，并同时提供所有必要参数，然后调用的时候只需要将实例对象传递给它即可，比如：

```
>>> p = Point(3, 4)
>>> d = operator.methodcaller('distance', 0, 0)
>>> d(p)
5.0
>>>
```

通过方法名称字符串来调用方法通常出现在需要模拟 `case` 语句或实现访问者模式的时候。参考下一小节获取更多高级例子。