

6.1.4 JSON 数据

JSON 已经成为通过 HTTP 请求在 Web 浏览器和其他应用程序之间发送数据的标准格式之一。这是一种比表格型文本格式（如 CSV）灵活得多的数据格式。下面是一个例子：

```
obj = """
{"name": "Wes",
 "cities_lived": ["Akron", "Nashville", "New York", "San Francisco"],
 "pet": null,
 "siblings": [{"name": "Scott", "age": 34, "hobbies": ["guitars", "soccer"]},
              {"name": "Katie", "age": 42, "hobbies": ["diving", "art"]}
]
"""
```

除了空值 `null` 和一些其他的细微差别（如列表末尾不允许存在多余的逗号）之外，JSON 非常接近于正规的 Python 代码。JSON 的基本类型有对象（字典）、数组（列表）、字符串、数值、布尔值以及空值。对象中所有的键都必须是字符串。存在多个 Python 库可以读写 JSON 数据。这里我将使用 `json` 包，因为它内置于 Python 标准库。通过 `json.loads` 即可将 JSON 字符串转换成 Python 形式：

```
In [68]: import json

In [69]: result = json.loads(obj)

In [70]: result
Out[70]:
{'name': 'Wes',
 'cities_lived': ['Akron', 'Nashville', 'New York', 'San Francisco'],
 'pet': None,
 'siblings': [{'name': 'Scott',
                'age': 34,
                'hobbies': ['guitars', 'soccer']},
              {'name': 'Katie', 'age': 42, 'hobbies': ['diving', 'art']}]}
```

`json.dumps` 将 Python 对象转换成 JSON 格式：

```
In [71]: asjson = json.dumps(result)

In [72]: asjson
Out[72]: '{"name": "Wes", "cities_lived": ["Akron", "Nashville", "New York", "San Francisco"], "pet": null, "siblings": [{"name": "Scott", "age": 34, "hobbies": ["guitars", "soccer"]}, {"name": "Katie", "age": 42, "hobbies": ["diving", "art"]}]}'
```

用户可以自行决定如何将（一个或一组）JSON 对象转换为 `DataFrame` 或其他便于分析的数据结构。最简单方便的方式是向 `DataFrame` 构造器传入一个字典的列表（就是上面的 JSON 对象），并选取数据字段的子集：