

关于 Python 类型特性和变量管理的一点说明

《用 PYTHON 玩转数据》

BY DAZHUANG@NJU

1. Python 类型特性

首先请回答 Python 是()态()类型语言。

- A. 静; 弱
- B. 动; 弱
- C. 静; 强
- D. 动; 强

答案应该 D, Python 是动态强类型语言。简单来说, 动态是指 Python 中变量可以指向不同类型的对象, 例如 `x = 3`, `x = 1.5`, 而在一般程序设计语言如 C 语言中在使用变量前需要先声明变量类型例如 `int x`, 表明变量 `x` 的类型为 `int`, 那此时 `x` 的类型就确定了, 如果执行赋值运算 “`x = 1.5`”, `x` 获得的值是 1 而不是 1.5, 这种方式称为静态的; 强类型同样可对比理解, 在 Python 中 `1 + 'A'` 是不允许的, 两者类型必须一致, 而在 C 语言中这个表达式是将 1 和字符 A 的 ASCII 值 65 相加, 结果是 66, 因此 Python 是强类型。

2. Python 变量管理

每个对象被创建时都会获得一个身份 id, 可用 `id()` 函数查看, 同时会伴随一个引用计数器。例如:

```
>>> PI = 3.14159
```

```
>>> pi = 3.14159
```

```
>>> PI is pi
```

```
False
```

分析: 因为 `PI` 和 `pi` 指向了两个不同的 3.14159, 它们有各自的内存空间, 可通过 `id()` 函数查看

```
>>> id(PI)

2376033150440

>>> id(pi)

2376033150416

>>> pi = PI

>>> PI is pi

True
```

分析：此时 `pi` 指向了 `PI` 指向的内存区域，可通过 `id()` 函数查看

```
>>> id(PI)

2376033150440

>>> id(pi)

2376033150440

>>> p = 3

>>> q = 3

>>> p is q

True
```

我们会发现上面的整型变量 `p` 和 `q` 与前面的浮点型变量 `PI` 和 `pi` 的情况不一样，在 `Python` 中确实存在特殊的地方，小整数（并无确切范围）和字符串是不可变的，`Python` 会采用高效的方式去存储，大家可以试试字符串是否确实是这样。