

实训一：使用数组比较运算对超市牛奶价格进行对比

源程序：

```
import numpy as np

# A 超市的牛奶价格
milk_a = np.array([19.9, 25, 29.9, 45, 39.9])

# B 超市的牛奶价格
milk_b = np.array([18.9, 25, 24.9, 49, 35.9])

comparison = milk_a > milk_b

print("A 超市的牛奶价格是否大于 B 超市的牛奶价格：", comparison)
```

过程性结果：

实训一：使用数组比较运算对超市牛奶价格进行对比

```
[7]: import numpy as np

# A 超市的牛奶价格
milk_a = np.array([19.9, 25, 29.9, 45, 39.9])

# B 超市的牛奶价格
milk_b = np.array([18.9, 25, 24.9, 49, 35.9])

comparison = milk_a > milk_b
print("A 超市的牛奶价格是否大于 B 超市的牛奶价格：", comparison)

A 超市的牛奶价格是否大于 B 超市的牛奶价格： [ True False  True False  True]
```

结论：

第一个牛奶产品在 A 超市的价格(19.9)高于 B 超市的价格(18.9)，结果为 True。

第二个牛奶产品在 A 超市的价格(25)等于 B 超市的价格(25)，结果为 False。

第三个牛奶产品在 A 超市的价格(29.9)高于 B 超市的价格(24.9)，结果为 True。

第四个牛奶产品在 A 超市的价格(45)低于 B 超市的价格(49)，结果为 False。

第五个牛奶产品在 A 超市的价格(39.9)高于 B 超市的价格(35.9)，结果为 True。

实训二：创建 6x6 的简单数独游戏矩阵

源程序：

创建一个 6x6 的矩阵

```
sudoku = np.zeros((6,6), dtype=int)
```

填充矩阵，确保每行和每列的数字为 1-6 且不重复

```
sudoku[0] = [1, 2, 3, 4, 5, 6]
```

```
sudoku[1] = [2, 3, 4, 5, 6, 1]
```

```
sudoku[2] = [3, 4, 5, 6, 1, 2]
```

```
sudoku[3] = [4, 5, 6, 1, 2, 3]
```

```
sudoku[4] = [5, 6, 1, 2, 3, 4]
```

```
sudoku[5] = [6, 1, 2, 3, 4, 5]
```

```
print("6x6 数独游戏矩阵：")
```

```
print(sudoku)
```

过程性结果：

实训二：创建 6x6 的简单数独游戏矩阵

```
[8]: # 创建一个 6x6 的矩阵
sudoku = np.zeros((6,6), dtype=int)

# 填充矩阵，确保每行和每列的数字为 1-6 且不重复
sudoku[0] = [1, 2, 3, 4, 5, 6]
sudoku[1] = [2, 3, 4, 5, 6, 1]
sudoku[2] = [3, 4, 5, 6, 1, 2]
sudoku[3] = [4, 5, 6, 1, 2, 3]
sudoku[4] = [5, 6, 1, 2, 3, 4]
sudoku[5] = [6, 1, 2, 3, 4, 5]

print("6x6 数独游戏矩阵：")
print(sudoku)
```

6x6 数独游戏矩阵：

```
[[1 2 3 4 5 6]
 [2 3 4 5 6 1]
 [3 4 5 6 1 2]
 [4 5 6 1 2 3]
 [5 6 1 2 3 4]
 [6 1 2 3 4 5]]
```

结论：

矩阵的每一行包含 1 到 6 的数字、每一列也包含 1 到 6 的数字，并且不重复。

通过这个练习，我们巩固了矩阵创建和数组索引的使用方法，为更复杂的数据操作和分析打下了基础。