

vncoder.vn

Bài 12: Toán tử nhị phân

3-4 phút

1. bitwise_and :

Phép toán bitwise AND trên các bit tương ứng của biểu diễn nhị phân của số nguyên trong mảng đầu vào được tính bằng hàm `np.bitwise_and()`.

Ví dụ :

```
import numpy as np
print 'Binary equivalents of 13 and 17:'
a,b = 13,17
print bin(a), bin(b)
print '\n'

print 'Bitwise AND of 13 and 17:'
print np.bitwise_and(13, 17)
```

Kết quả :

```
Binary equivalents of 13 and 17:
0b1101 0b10001
```

```
Bitwise AND of 13 and 17:
1
```

bảng bitwise AND sau đây.

2. bitwise_or :

Phép toán OR theo bit trên các bit tương ứng của biểu diễn nhị phân của số nguyên trong mảng đầu vào được tính bằng hàm `np.bitwise_or()`.

Ví dụ :

```
import numpy as np
a,b = 13,17
print 'Binary equivalents of 13 and 17:'
print bin(a), bin(b)

print 'Bitwise OR of 13 and 17:'
print np.bitwise_or(13, 17)
```

Kết quả :

```
Binary equivalents of 13 and 17:
0b1101 0b10001
```

```
Bitwise OR of 13 and 17:
29
```

3. invert :

Hàm này tính toán kết quả KHÔNG theo từng bit trên các số nguyên trong mảng đầu vào. Đối với các số nguyên có dấu, phần bù của hai được trả về.

Ví dụ :

```
import numpy as np

print 'Invert of 13 where dtype of ndarray is
uint8:'
print np.invert(np.array([13], dtype =
```

```
np.uint8))
print '\n'
# Comparing binary representation of 13 and
242, we find the inversion of bits

print 'Binary representation of 13:'
print np.binary_repr(13, width = 8)
print '\n'

print 'Binary representation of 242:'
print np.binary_repr(242, width = 8)
```

Kết quả :

```
Invert of 13 where dtype of ndarray is uint8:
[242]
```

```
Binary representation of 13:
00001101
```

```
Binary representation of 242:
11110010
```

Lưu ý rằng hàm `np.binary_repr ()` trả về biểu diễn nhị phân của số thập phân theo chiều rộng đã cho.

4. left_shift

Hàm `numpy.left_shift ()` dịch chuyển các bit trong biểu diễn nhị phân của một phần tử mảng sang trái theo các vị trí được chỉ định. Các số 0 bằng nhau được thêm vào từ bên phải.

Ví dụ :

```
import numpy as np
```

```
print 'Left shift of 10 by two positions:'
print np.left_shift(10,2)
print '\n'

print 'Binary representation of 10:'
print np.binary_repr(10, width = 8)
print '\n'

print 'Binary representation of 40:'
print np.binary_repr(40, width = 8)
# Two bits in '00001010' are shifted to left
and two 0s appended from right.
```

Kết quả :

Left shift of 10 by two positions:

40

Binary representation of 10:

00001010

Binary representation of 40:

00101000

5. right_shift :

Hàm `numpy.right_shift()` dịch chuyển các bit trong biểu diễn nhị phân của một phần tử mảng sang bên phải theo các vị trí được chỉ định và một số lượng 0 bằng nhau được thêm vào từ bên trái.

Ví dụ :

```
import numpy as np
```

```
print 'Right shift 40 by two positions:'
print np.right_shift(40,2)
print '\n'

print 'Binary representation of 40:'
print np.binary_repr(40, width = 8)
print '\n'

print 'Binary representation of 10'
print np.binary_repr(10, width = 8)
# Two bits in '00001010' are shifted to right
and two 0s appended from left.
```

Kết quả :

```
Right shift 40 by two positions:
10
```

```
Binary representation of 40:
00101000
```

```
Binary representation of 10
00001010
```