

[← course home \(/table-of-contents\)](#)

Bitwise XOR (eXclusive OR)

The **XOR** operation (or **exclusive or**) takes two bits and returns 1 if **exactly one** of the bits is 1. Otherwise, it returns 0.

```
1 ^ 1 → 0
1 ^ 0 → 1
0 ^ 1 → 1
0 ^ 0 → 0
```

Think of it like a bag of chips where only one hand can fit in at a time. If no one reaches for chips, no one gets chips, and if both people reach for chips, they can't fit and no one gets chips either!

When performing XOR on two integers, the XOR operation is calculated on each pair of bits (the two bits at the same index in each number).

```
5 ^ 6 # gives 3
```

Python 3.6 ▼

```
# At the bit level:
#   0101 (5)
# ^ 0110 (6)
# = 0011 (3)
```

[← course home \(/table-of-contents\)](#)

Next up: Bitwise NOT ➔ [\(/concept/not?course=fc1§ion=bit-manipulation\)](/concept/not?course=fc1§ion=bit-manipulation)