Bitwise OR

The **OR** operation takes two bits and returns 1 if **either** of the bits are 1. Otherwise, it returns 0.

```
1 | 1  →  1
1 | 0  →  1
0 | 1  →  1
0 | 0  →  0
```

Think of it like a bucket with two holes in it. If *both* holes are closed, no water comes out. If *either* hole is open, *or if both* are open, water comes out.

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

Python 3.6

```
5 | 6  # gives 7

# At the bit level:
#     0101  (5)
#   | 0110  (6)
#   = 0111  (7)
```