Bitwise AND

The **AND** bitwise operation takes two sets of bits and for each pair of bits (the two bits at the same index in each set) returns `1` if **both** bits are `1`. Otherwise, it returns `0`.

```java
1 & 1 // 1
1 & 0 // 0
0 & 1 // 0
0 & 0 // 0
```

Think of it like a hose with two knobs. *Both* knobs must be set to on for water to come out.

When performing AND on two integers, only the digit columns shared by both integers remain:

```java
5 & 6
// gives 4


// at the bit level:
//     0101  (5)
//   & 0110  (6)
//   = 0100  (4)
```

See also:

- Binary Numbers (/concept/binary-numbers)
- Bitwise OR (/concept/or)
- Bitwise NOT (/concept/not)
- Bitwise XOR (eXclusive OR) (/concept/xor)
- Bit Shifting (/concept/bit-shift)

What's next?

If you're ready to start applying these concepts to some problems, check out our mock coding interview questions (/next).

They mimic a real interview by offering hints when you're stuck or you're missing an optimization.

Try some questions now ➜

Want more coding interview help?

Check out **interviewcake.com** for more advice, guides, and practice questions.