# how is 0 represented

Great question! The representation of **0** in computers (specifically in JavaScript and most programming languages) depends on the data type and the underlying binary format. Let's break it down for both **integers** and **floating-point numbers**, as used in JavaScript:

### 1. Integer Representation (Two's Complement)

For integer representations (most languages use **two's complement** for signed integers):

- **0** is simply represented as all bits set to zero.
- Example for an 8-bit integer:
  `00000000` = 0
- The **sign bit** (the leftmost bit) is `0`, indicating positive, but all other bits are also `0`, giving a value of zero.
- In 32 bits:
  `00000000 00000000 00000000 00000000` = 0

### 2. IEEE 754 Floating-Point Representation (JavaScript Number Type)

JavaScript numbers use the 64-bit **IEEE 754 double-precision floating-point** format. Here's how **0** is encoded:

- **Sign bit**: 0 for positive, 1 for negative
- **Exponent**: All 0s (11 bits)
- **Fraction/Mantissa**: All 0s (52 bits)

**Positive zero (default 0):**

```
sign      exponent         mantissa
0         000 0000 0000    000...000   (all zeros)
```

All bits are zero:
`0000000000000000000000000000000000000000000000000000000000000000`
(the actual binary string is 64 bits, with each section as above)

**Negative zero (-0):**

- Yes, in IEEE 754 and JavaScript, there is a -0!
- It's the same as above, except the **sign bit** is 1:

```
sign      exponent         mantissa
```

```
1         000 0000 0000    000...000
```

This distinction only matters in special mathematical situations, like dividing by zero.

### 3. In JavaScript:

- `0` and `-0` are both considered equal (`0 === -0` is `true`), but can be distinguished with `Object.is(0, -0)` (returns `false`).

### Summary Table

| Type | Binary Bits | Value |
|---|---|---|
| 8-bit Integer Zero | 00000000 | 0 |
| 32-bit Integer Zero | 00000000 00000000 00000000 00000000 | 0 |
| Double 0 (JavaScript) | All 64 bits zero | +0 |
| Double -0 | Only sign bit 1, rest zeros | -0 |

### Key Point

- **Zero** is represented in binary as “all zeros” for both integers and floating-point numbers, except for negative zero in floating-point where only the sign bit differs.

If you’d like, I can show you the exact hexadecimal or binary representation for zero in memory for various number types—just ask!