

signed bit

A **signed bit** is a specific bit (usually the highest or most significant bit in a binary number) used to indicate the sign of a number—whether it is positive or negative.

How Signed Bit Works in JavaScript Numbers

- **JavaScript numbers** follow the IEEE 754 double-precision 64-bit floating point standard.
 - Out of the 64 bits:
 - **1 bit** (bit 63) is the **sign bit**.
 - **11 bits** (bits 52–62) are for the exponent.
 - **52 bits** (bits 0–51) are for the fraction/mantissa^{[1] [2]}.
 - If the sign bit is **0**, the number is positive.
 - If the sign bit is **1**, the number is negative.

Sign bit (bit 63):

- 0 = positive
- 1 = negative^{[1] [2]}

Bitwise Operations

- While JavaScript numbers are 64 bits, bitwise operations convert numbers temporarily to **signed 32-bit integers**. In 32-bit signed representation:
 - The **leftmost bit** (bit 31) is the sign bit^{[3] [4]}.
 - 0 = positive; 1 = negative.

Examples:

- The binary for 5 (32 bits): 00000000000000000000000000000101 (sign bit is 0)
- The binary for -6 (32 bits): 1111111111111111111111111111010 (sign bit is 1)^[3]

General Concept (not just JavaScript)

- In **signed binary numbers** (used in most programming languages for integers), the sign bit is typically the **most significant bit (MSB)**.
 - 0 = positive value
 - 1 = negative value (often in two's complement form for integers)^[5]

Summary:

The **signed bit** is the special bit in a number's binary representation that determines its sign. In

JavaScript numbers (IEEE 754), it's the highest bit, and the same applies for 32-bit integers used in bitwise operations^{[2] [3] [1] [5]}.

*
**

1. https://www.w3schools.com/js/js_numbers.asp
2. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number
3. https://www.w3schools.com/js/js_bitwise.asp
4. https://exploringjs.com/js/book/ch_numbers.html
5. <https://www.geeksforgeeks.org/digital-logic/basics-of-signed-binary-numbers-of-ranges-of-different-datatypes/>