



FIGURE 4.2 The result of invalid mathematical operations will be the special values NaN and Infinity.

TABLE 4.2 Operator Precedence

PRECEDENCE	OPERATOR	NOTE
1	. []	member operators
1	new	creates new objects
2	()	function call
3	++ --	increment and decrement
4	!	logical not
4	+ -	unary positive and negative
4	typeof void delete	
5	* / %	multiplication, division, and modulus
6	+ -	addition and subtraction
8	< <= > >=	comparison
9	== != === !==	equality
13	&&	logical and
14		logical or
15	?:	conditional operator
16	= += -= *= /= %= <<= >>= >>>= &= ^= =	assignment operators

The last thing to know about performing arithmetic in JavaScript is if the result of the arithmetic is invalid, JavaScript will return one of two special values:

- NaN, short for *Not a Number*
- Infinity

For example, you'll get these results if you attempt to perform arithmetic using strings or when you divide a number by zero, which surprisingly doesn't create an error (**Figure 4.2**). In Chapter 5, you'll learn how to use the `isNaN()` and `isFinite()` functions to verify that values are numbers safe to use as such.