Table 4-1. JavaScript operators

Operator	Operation	A	N	Types
`++`	Pre- or post-increment	R	1	lval→num
`--`	Pre- or post-decrement	R	1	lval→num
`-`	Negate number	R	1	num→num
`+`	Convert to number	R	1	any→num
`~`	Invert bits	R	1	int→int
`!`	Invert boolean value	R	1	bool→bool
`delete`	Remove a property	R	1	lval→bool
`typeof`	Determine type of operand	R	1	any→str
`void`	Return undefined value	R	1	any→undef
`**`	Exponentiate	R	2	num,num→num
`*, /, %`	Multiply, divide, remainder	L	2	num,num→num
`+, -`	Add, subtract	L	2	num,num→num
`+`	Concatenate strings	L	2	str,str→str
`<<`	Shift left	L	2	int,int→int
`>>`	Shift right with sign extension	L	2	int,int→int
`>>>`	Shift right with zero extension	L	2	int,int→int
`<, <=,>, >=`	Compare in numeric order	L	2	num,num→bool
`<, <=,>, >=`	Compare in alphabetical order	L	2	str,str→bool
`instanceof`	Test object class	L	2	obj,func→bool
`in`	Test whether property exists	L	2	any,obj→bool
`==`	Test for non-strict equality	L	2	any,any→bool
`!=`	Test for non-strict inequality	L	2	any,any→bool
`===`	Test for strict equality	L	2	any,any→bool

Operator	Operation	A	N	Types
!==	Test for strict inequality	L	2	any,any→bool
&	Compute bitwise AND	L	2	int,int→int
^	Compute bitwise XOR	L	2	int,int→int
|	Compute bitwise OR	L	2	int,int→int
&&	Compute logical AND	L	2	any,any→any
||	Compute logical OR	L	2	any,any→any
??	Choose 1st defined operand	L	2	any,any→any
?:	Choose 2nd or 3rd operand	R	3	bool,any,any→any
=	Assign to a variable or property	R	2	lval,any→any
**=, *=, /=, %=, +=, -=, &=, ^=, |=, <<=, >>=, >>>=	Operate and assign	R	2	lval,any→any
,	Discard 1st operand, return 2nd	L	2	any,any→any