*// Arrays and objects can hold other arrays and objects:*

*let* points = [ *// An array with 2 elements.*

{x: 0, y: 0}, *// Each element is an object.*

{x: 1, y: 1}

];

*// Operators act on values (the operands) to produce a new value.*

*// Arithmetic operators are some of the simplest:*

3 + 2 *// => 5: addition*

3 - 2 *// => 1: subtraction*

3 \* 2 *// => 6: multiplication*

3 / 2 *// => 1.5: division*

points[1].x - points[0].x *// => 1: more complicated operands also work*

"3" + "2" *// => "32": + adds numbers, concatenates strings*

*// JavaScript defines some shorthand arithmetic operators*

*let* count = 0; *// Define a variable*

count++; *// Increment the variable*

count--; *// Decrement the variable*

count += 2; *// Add 2: same as count = count + 2;*

count \*= 3; *// Multiply by 3: same as count = count \* 3;*

count *// => 6: variable names are expressions, too.*

*// Equality and relational operators test whether two values are equal,*

*// unequal, less than, greater than, and so on. They evaluate to true or false.*

*let* x = 2, y = 3; *// These = signs are assignment, not equality tests*

x === y *// => false: equality*

x !== y *// => true: inequality*

x < y *// => true: less-than*

x <= y *// => true: less-than or equal*

x > y *// => false: greater-than*

x >= y *// => false: greater-than or equal*

"two" === "three" *// => false: the two strings are different*

"two" > "three" *// => true: "tw" is alphabetically greater than "th"*

false === (x > y) *// => true: false is equal to*

false

*// Logical operators combine or invert boolean values*

(x === 2) && (y === 3) *// => true: both comparisons are true. && is AND*

(x > 3) || (y < 3) *// => false: neither comparison is true. || is OR*

!(x === y) *// => true: ! inverts a boolean value*