

Using math expressions in JS

 [Share](#)  [Tweet](#)  [Email](#)

In the JavaScript language (and most programming languages), we can use mathematical operators to calculate numbers and create expressions.

You've already seen examples of adding in JS, using the `+` operator. You can also use `-` for subtraction, `*` for multiplication, `/` for division, and `%` to take the remainder. Here are examples of those in action:

```
var x = 10;
var a = x + 5; // add 5, result is 15
var b = x - 5; // subtract 5, result is 5
var c = x * 2; // multiply by 2, result is 20
var d = x / 4; // divide by 4, result is 2.5
var e = x % 4; // divide by 4 & return remainder, result is 2
```

When you use multiple math operators in a single expression, the computer follows an "order of operations" to make sure it computes the result the same way each time. It's the same order of operations that you probably learned in math class, and is commonly remembered as "PEMDAS" - parentheses, exponents, multiplication, division, addition, subtraction. You can learn more about [order of operations](#) on Khan Academy.

For example, in the following code, it would first evaluate the two expressions in parentheses, then it would

multiply those results together, and finally, it would add 100 to that result.

```
var x = 10;  
var a = (x + 10) * (x / 2) + 100;
```

Just like in math expressions, you only need parentheses if you want your expression to be evaluated differently than it would be according to the order of operations. You can leave them out otherwise.

Next up, we'll go through an example using more complicated expressions with variables and math operators. If you want, play around with some of these math expressions first, to make sure you have a handle on how the computer evaluates them.