

Expressions

What are expressions and why are they used in JavaScript? Let's find out in this lesson!

An *expression* is a piece of code that produces a value. An expression is created by combining variables, values and operators. Every expression has a value and thus a type. Calculating an expression's value is called *evaluation*. During evaluation, variables are replaced by their values.

```
// 3 is an expression whose value is 3
const c = 3;
// c is an expression whose value is the value of c (3 here)
let d = c;
// (d + 1) is an expression whose value is d's + 1 (4 here)
d = d + 1; // d now contains the value 4
console.log(d); // Show 4
```



Operator priority inside an expression is the same as in math. However, an expression can integrate *parenthesis* that modify these priorities.

```
let e = 3 + 2 * 4; // e contains 11 (3 + 8)
console.log(e);
e = (3 + 2) * 4; // e contains 20 (5 * 4)
console.log(e);
```



It is possible to include expressions in a string by using *backticks* (```) to delimit the string. Such a string is called a *template literal*. Inside a template literal, expressions are identified by the `${expression}` syntax.

This is often used to create strings containing the values of some variables.

```
const country = "France";
```



```
console.log(`I live in ${country}`); // Show "I live in France"  
const x = 3;
```

```
const y = 7;  
console.log(`${x} + ${y} = ${x + y}`); // Show "3 + 7 = 10"
```

