Operator Precedence

This lesson focusses and emphasizes on operator precedence along with relevant examples.

WE'LL COVER THE FOLLOWING

- Operators precedence from **highest to lowest** precedence.
- Quiz time! :)

When you use compound expressions, it is very rare that operators are evaluated from left to right. For example, the following expression results in `7`, and not `9`:

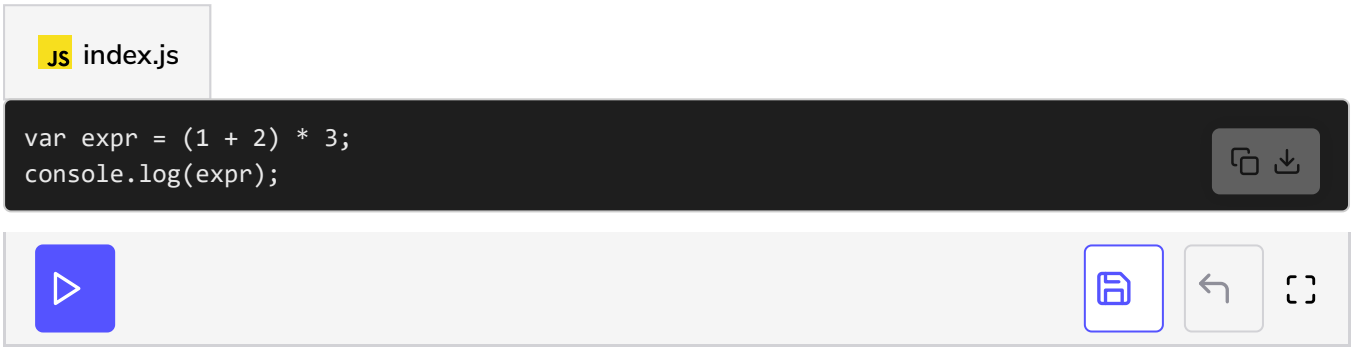
index.js

```
var expr = 1 + 2 * 3;
console.log(expr);
```

The precedence of the multiply operator is higher than the precedence of the

add operator, so first `2*3` will be evaluated to 6, and then `1 + 6` is calculated.

You can override the precedence of operators with parentheses. For example, this expression will result in 9:



Just as in every programming language, operator precedence is declared in JavaScript, too.

The table below summarizes the operators with their precedence, ordered from **highest to lowest** precedence.

Operators with the same precedence are evaluated from left to right.

Operators precedence from **highest to lowest** precedence.

Operator	Description
`. [] ()`	Field access, array indexing, function calls, and expression grouping
`+ - ++ -- ~ ! delete new typeof`	Unary operators, `delete` operator, object creation, `typeof` operator
`* / %`	Multiplicative operators
`+ -`	Addition (string concatenation), subtraction
`<< >> >>>`	Bitwise shift operators
`< <= > >= instanceof`	Relational operators, `instanceof` operator
`== != === !==`	Equality operators
`&`	Bitwise AND
`^`	Bitwise XOR
`|`	Bitwise OR
`&&`	Logical AND
`||`	Logical OR
`?:`	Conditional operator
`= += -= *= /= %= <<= >>= >>>=`	Assignment and compound assignment operators
`,`	Comma operator

Quiz time! :)

It’s time to test how much we’ve learned in this lesson with a short quiz!

Question 1

Q

What is the order of operations for the following piece of code?

```
let num_js = (19 + 10) * (10 - 10);
```

Check Answers

Question 2

Q

What is the value for the following set of operations?

```
10 * ((2 - 7) + 20) / 5
```

Check Answers

In the *next lesson*, we are going to discuss flow-control statements.

See you there!