mdn web docs

> Эта страница была переведена с английского языка силами сообщества. Вы тоже можете внести свой вклад, присоединившись к русскоязычному сообществу MDN Web Docs.

Приоритет операторов

Приоритет операторов определяет порядок, в котором операторы выполняются. Операторы с более высоким приоритетом выполняются первыми.

Интерактивный пример

JavaScript Demo: Expressions - Operator precedence

```
1  console.log(3 + 4 * 5); // 3 + 20
2  // Expected output: 23
3
4  console.log(4 * 3 ** 2); // 4 * 9
5  // Expected output: 36
6
7  let a;
8  let b;
9
10 console.log((a = b = 5));
11 // Expected output: 5
12
```

Run ›

Reset

Ассоциативность

Ассоциативность определяет порядок, в котором обрабатываются операторы с одинаковым приоритетом. Например, рассмотрим выражение:

```
a OP b OP c
```

Левая ассоциативность (слева направо) означает, что оно обрабатывается как `(a OP b) OP c`, в то время как правая ассоциативность (справа налево) означает, что они интерпретируются как `a OP (b OP c)`. Операторы присваивания являются право-ассоциативными, так что вы можете написать:

JS

```
a = b = 5;
```

с ожидаемым результатом, что `a` и `b` будут равны 5. Это происходит, потому что оператор присваивания возвращает тот результат, который присваивает. Сначала `b` становится равным 5, затем `a` принимает значение `b`.

Примеры

JS

```
3 > 2 && 2 > 1;
// вернёт true

3 > 2 > 1;
// вернёт false, потому что 3 > 2 возвращает true, в свою очередь true > 1 вернёт
false
// Добавление скобок значительно повышает читаемость выражения: (3 > 2) > 1
```

Таблица

Операторы упорядочены с самого высокого (18) до самого низкого (1) приоритета.

Обратите внимание, что spread-оператор (`...`) намеренно не включен в таблицу, потому что он вообще не является оператором и правильно

говорить `spread-синтаксис` . Подробнее можно почитать в ответе на Stack Overflow (en) .

Приоритет	Тип оператора	Ассоциативность	Конкретные операторы
18	Группировка	не определено	`( … )`
17	Доступ к свойствам	слева направо	`… . …`
	Доступ к свойствам с возможностью вычисления		`… [ … ]`
	new (со списком аргументов)	не определено	`new … ( … )`
	Вызов функции	слева направо	`… ( … )`
	Оператор опциональной последовательности (?.)		`?.`
16	new (без списка аргументов)	справа налево	`new …`
15	Постфиксный инкремент	не определено	`… ++`
	Постфиксный декремент		`… --`
14	Логическое отрицание (!) (англ.)	справа налево	`! …`
	Побитовое отрицание (~) (англ.)		`~ …`
	Унарный плюс (англ.)		`+ …`
	Унарный минус (англ.)		`- …`
	Префиксный инкремент (англ.)		`++ …`
	Префиксный декремент (англ.)		`-- …`
	typeof		`typeof …`
	void		`void …`
	delete		`delete …`

	await		`await …`
13	Возведение в степень (**) (англ.)	справа налево	`… ** …`
12	Умножение (*) (англ.)	слева направо	`… * …`
	Деление (/) (англ.)		`… / …`
	Остаток от деления (%) (англ.)		`… % …`
11	Сложение (+) (англ.)	слева направо	`… + …`
	Вычитание (-) (англ.)		`… - …`
10	Побитовый сдвиг влево (<<) (англ.)	слева направо	`… << …`
	Побитовый сдвиг вправо (>>) (англ.)		`… >> …`
	Сдвиг вправо с заполнением нулей (>>>) (англ.)		`… >>> …`
9	Меньше (<) (англ.)	слева направо	`… < …`
	Меньше или равно (<=) (англ.)		`… <= …`
	Больше (>) (англ.)		`… > …`
	Больше или равно (>=) (англ.)		`… >= …`
	in		`… in …`
	instanceof		`… instanceof …`
8	Равенство (==) (англ.)	слева направо	`… == …`
	Неравенство (!=) (англ.)		`… != …`
	Строгое равенство (===) (англ.)		`… === …`
	Строгое неравенство (!==) (англ.)		`… !== …`
7	Побитовое «И» (&) (англ.)	слева направо	`… & …`

6	Побитовое исключающее «ИЛИ» (^) (англ.)	слева направо	… ^ …
5	Побитовое «ИЛИ» (|) (англ.)	слева направо	… | …
4	Логическое «И» (&&) (англ.)	слева направо	… && …
3	Логическое «ИЛИ» (||) (англ.)	слева направо	… || …
	Оператор нулевого слияния (??)		… ?? …
2	Присваивание (англ.)	справа налево	… = …
			… += …
			… -= …
			… **= …
			… *= …
			… /= …
			… %= …
			… <<= …
			… >>= …
			… >>>= …
			… &= …
			… ^= …
			… |= …
			… &&= …
			… ||= …
			… ??= …
	Условный (тернарный) оператор	справа налево	… ? … : …

	yield	справа налево	`yield …`
	yield*		`yield* …`
1	Запятая / Последовательность	слева направо	`… , …`

Help improve MDN

Was this page helpful to you?

Yes No

Learn how to contribute.

This page was last modified on 11 дек. 2023 г. by MDN contributors.