



Приоритет операторов

Приоритет операторов определяет порядок, в котором операторы выполняются.

Операторы с более высоким приоритетом выполняются первыми.

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:

```
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

```
3 > 2 && 2 > 1
// вернёт true

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

Таблица

Данная таблица упорядочена с самого высокого приоритета (20) до самого низкого (1).

Приоритет	Тип оператора	Ассоциативность	Конкретные операторы
20	Группировка	не определено	(...)
19	Доступ к свойствам	слева направо
	Доступ к свойствам с возможностью вычисления	слева направо	... [...]
	new (со списком аргументов)	не определено	new ... (...)
	Вызов функции	слева направо	... (...)
18	new (без списка аргументов)	справа налево	new ...
17	Постфиксный инкремент	не определено	... ++
	Постфиксный декремент		... --
16	Логическое отрицание	справа налево	! ...

	Побитовое отрицание		~ ...
	Унарный плюс		+ ...
	Унарный минус		- ...
	Префиксный инкремент		++ ...
	Префиксный декремент		-- ...
	typeof		typeof ...
	void		void ...
	delete		delete ...
	await		await ...
15	Возведение в степень	справа налево	... ** ...
14	Умножение	слева направо	... * ...
	Деление		... / ...
	Остаток		... % ...
13	Сложение	слева направо	... + ...
	Вычитание		... - ...
12	Побитовый сдвиг влево	слева направо	... << ...
	Побитовый сдвиг вправо		... >> ...
	Сдвиг вправо с заполнением нулей		... >>> ...
11	Меньше	слева направо	... < ...
	Меньше или равно		... <= ...
	Больше		... > ...

	Больше или равно		... >= ...
	in		... in ...
	instanceof		... instanceof ...
10	Равно	слева направо	... == ...
	Не равно		... != ...
	Строго равно		... === ...
	Строго не равно		... !== ...
9	Побитовое «И»	слева направо	... & ...
8	Побитовое исключяющее «ИЛИ»	слева направо	... ^ ...
7	Побитовое «ИЛИ»	слева направо
6	Логическое «И»	слева направо	... && ...
5	Логическое «ИЛИ»	слева направо
4	Условный	справа налево	... ? ... : ...
3	Присваивание	справа налево	... = ...
			... += ...
			... -= ...
			... **= ...
			... *= ...
			... /= ...
			... %= ...

			... ~ ...
			... <<= ...
			... >>= ...
			... >>>= ...
			... &= ...
			... ^= ...
			... = ...
2	yield	справа налево	yield ...
	yield*		yield* ...
1	Запятая / Последовательность	слева направо	... , ...

Last modified: 21 июн. 2020 г., by MDN contributors