

13. Operatorlarning ustuvorligi

Reja:

▼ Operatorlar ustuvorligi



Javascriptda operatorlarning ustuvorligi (ya'ni, birinchi navbatda qaysi amallar bajarilishi) aniqlangan tartibga asoslanadi. Quyida, **arifmetik**, **solishtirish**, **tayinlash** va **boolean** operatorlarning ustuvorlik tartibi tushuntiriladi.

▼ Arifmetik operatorlar (Arithmetic Operators)

Arifmetik operatorlar quyidagi ustuvorlikka ega:

- **Ko'paytirish, bo'lish va qoldiq olish:** `*`, `/`, `%` — yuqori ustuvorlikda.
- **Qo'shish va ayirish:** `+`, `-` - ularning ustuvorligi pastroq.

```
const result = 10 + 5 * 2; // Natija: 20 (birinchi 5 * 2, keyin 10 + 10)
```

Dastlab, ko'paytirish va bo'lish operatorlari bajariladi, keyin esa qo'shish va ayirish.

▼ Solishtirish operatorlari (Comparison Operators)

Solishtirish operatorlari qiymatlarni taqqoslash uchun ishlatiladi:

- `==`, `!=` — tenglik va teng emaslik.
- `===`, `!==` — qat'iy tenglik va teng emaslik (turi ham hisobga olinadi).
- `<`, `>`, `<=`, `>=` — katta, kichik yoki tenglikni tekshirish.

Ustuvorlik tartibida bu operatorlar arifmetik operatorlardan keyin bajariladi.

```
const result = 5 > 3 && 10 > 2;  
// Natija: true (solishtirish birinchi, keyin AND bajar
```

```
iladi)
```

Solishtirish operatorlari har doim arifmetik hisob-kitoblardan keyin bajariladi.

▼ Tayinlash operatorlari (Assignment Operators)

Tayinlash operatorlari (`=`, `+=`, `-=`, `*=` va boshqalar) eng past ustuvorlikka ega. Bu shuni anglatadiki, boshqa barcha operatorlar bajarilgandan keyin tayinlash operatori bajariladi.

```
let x = 10;  
x += 5; // x = x + 5; Natija: 15
```

Tayinlash operatorlari qiymatlarni o'zgartirishda ishlatiladi va ular eng past ustuvorlikka ega.

▼ Boolean operatorlari (Logical Operators)

- `&&` (AND): Ikki qiymat ham `true` bo'lsa, natija `true`.
- `||` (OR): Har qanday bir qiymat `true` bo'lsa, natija `true`.
- `!` (NOT): Qiymatni teskarisiga o'zgartiradi.

Ustuvorlik:

1. `!` (NOT) — birinchi bajariladi.
2. `&&` (AND) — keyin bajariladi.
3. `||` (OR) — oxirgi bajariladi.

```
const result = true || false && !true;  
// Natija: true (birinchi NOT, keyin AND, oxirida OR bajariladi)
```

Boolean operatorlari quyidagi tartibda ishlaydi: `NOT` birinchi o'rinda, `AND` undan keyin, va oxirida `OR`.

Operatorlarning umumiy ustuvorlik tartibi

1. **Qavslar:** `()` — Har doim birinchi bajariladi.
2. **Arifmetik operatorlar:** `*`, `/`, `%` → `+`, `-`.
3. **Solishtirish:** `<`, `>`, `<=`, `>=`, `==`, `===`.
4. **Logical NOT:** `!`.
5. **Logical AND va OR:** `&&`, `||`.
6. **Tayinlash operatorlari:** `=`, `+=`, `--`, va boshqalar.

▼ Suhbat savollari

▼ Amaliy masala

```
const x = 5;  
const y = 10;  
const z = x + y > 15 && y - x < 10; // Natija nima bo'lad:
```

▼ Javob

Tushuntirish:

1. `x + y > 15`: Birinchi `x + y` hisoblanadi, natija: 15. Shundan keyin `15 > 15` ⇒ `false`.
2. `y - x < 10`: Keyin `y - x` hisoblanadi, natija: 5. Shundan keyin `5 < 10` ⇒ `true`.
3. `false && true`: Natija: `false`.

▼ Javascriptda qanday operatorlar birinchi bajariladi?

1. **Qavslar:** `()` har doim birinchi bajariladi.
2. **Arifmetik operatorlar:**
 - Ko'paytirish, bo'lish va qoldiq olish: `*`, `/`, `%`.
 - Qo'shish va ayirish: `+`, `-`.
3. **Solishtirish operatorlari:** `<`, `>`, `<=`, `>=`, `==`, `!=`.
4. **Boolean NOT:** `!`.
5. **Boolean AND va OR:** `&&`, `||`.

6. **Tayinlash operatorlari:** `=`, `+=`, `-=`, va boshqalar.

▼ **Boolean operatorlarining ustuvorlik tartibi qanday?**

1. `!` (**NOT**): Birinchi bajariladi.
2. `&&` (**AND**): Ikkinchi ustuvorlik.
3. `||` (**OR**): Eng past ustuvorlik.

▼ **Kod natijasi qanday?**

```
const result = 3 + 5 * 2 > 10 && 10 - 5 < 5 || false;
```

▼ **Javob**

Bajarilish tartibi va tushuntirish:

1. **Arifmetik amallar:**

- `5 * 2` → `10`.
- `3 + 10` → `13`.
- `10 - 5` → `5`.

2. **Solishtirish amallari:**

- `13 > 10` → `true`.
- `5 < 5` → `false`.

3. **Boolean operatorlari:**

- `true && false` → `false`.
- `false || false` → `false`.

Natija:

```
const result = false;
```