

10. Ma'lumot turini o'zgartirish va coercion

Reja:

▼ Type conversion



Type conversion (ma'lumot turini o'zgartirish) - bir turdagi qiymatni boshqa turga aylantirishga aytiladi.

Number



Number primitiv o'rab turuvchi obyektidan foydalanib, berilgan ma'lumot turini son ko'rinishiga o'tkazish imkoniyati mavjud. Undan tashqari `parseInt()`, yoki `parseFloat()` funksiyalaridan foydalanib ham son ko'rinishiga o'tkazish mumkin.

```
const age = "27";
const amount = "123.45";
console.log(Number(age)); // 27
console.log(parseInt(age)); // 123 (butun son)
console.log(parseFloat(amount)); // 123.45 (o'nlik son)
```



Agar berilgan ma'lumot son ko'rinishiga o'tkazishni imkoni bo'lmasa, **NaN** maxsus qiymati yuzaga keladi.

```
const name = "Ulugbek";
console.log(Number(name)); // NaN
```

```
console.log(parseInt(name)); // NaN
console.log(parseFloat(name)); // NaN
```

String



String primitiv o'rab turuvchi obyektidan foydalanib, berilgan ma'lumot turini matn ko'rinishiga o'tkazish imkoniyati mavjud. Undan tashqari `.toString()` metodidan foydalanib ham matn ko'rinishga o'tkazish mumkin.

```
const age = 27;
console.log(String(age)); // "27"
console.log(age.toString()); // "27"
```

Boolean



Boolean primitiv o'rab turuvchi obyektidan foydalanib, berilgan ma'lumot turini boolean ko'rinishiga o'tkazish imkoniyati mavjud.

```
console.log(Boolean(0)); // false
console.log(Boolean(1)); // true
console.log(Boolean("hello")); // true
console.log(Boolean("")); // false
```

▼ Type coercion



Type coercion — Javascriptda ma'lumot turini avtomatik o'zgartirish jarayoniga aytiladi. Agar operatsiya turli ma'lumot turlarini o'z ichiga olsa, Javascript ular bilan ishlashga mos keladigan turga o'zgartiradi.

Stringga o'zgartirish - raqam matn bilan qo'shilganda, raqam matnga aylanadi.

```
console.log(5 + "10"); // "510" (raqam stringga o'zgartiriladi)
```

Raqamga o'zgartirish - string boshqa arifmetik amalda ishlatilganda, raqamga o'zgaradi.

```
console.log("5" * 2); // 10 (string raqamga aylantiriladi)
console.log("10" - 5); // 5 (string raqamga aylantiriladi)
```

Raqamga o'zgartirish - boolean qiymatlar arifmetik amallarda ishlatilganda, ular quyidagicha raqamga o'zgaradi:

- `true` → `1`
- `false` → `0`

```
console.log(true + 1); // 2 (true 1 ga aylanadi)
console.log(false + 5); // 5 (false 0 ga aylanadi)
```

▼ Truthy va Falsy



Javascriptda barcha qiymatlar boolean kontekstida (masalan, `if` shartlari yoki mantiqiy operatorlar) ishlatilganda **truthy** yoki **falsy** sifatida baholanadi. Bu qiymatning `true` yoki `false` kabi ishlashini belgilaydi.



Falsy qiymatlar — bu boolean kontekstida `false` sifatida ishlaydigan qiymatlar.

Javascriptda faqat **8 falsy qiymat** mavjud:

1. `false` (boolean qiymat `false`)
2. `0` (nol)
3. `-0` (manfiy nol)
4. `""` (bo'sh string)
5. `null`
6. `undefined`
7. `NaN` (raqam emas)
8. `0n` (nol bigintda)

```
console.log(Boolean(false)); // false
console.log(Boolean(0)); // false
console.log(Boolean(-0)); // false
console.log(Boolean("")); // false (empty string)
console.log(Boolean(null)); // false
console.log(Boolean(undefined)); // false
console.log(Boolean(NaN)); // false
console.log(Boolean(0n)); // false
```



Truthy qiymatlar - bu **falsy** bo'lmagan barcha qiymatlar. Ular boolean kontekstida `true` sifatida ishlaydi.

1. Nol bo'lmagan har qanday raqam (`1`, `5`, `3.14`)
2. Bo'sh bo'lmagan stringlar (`"hello"`, `"0"`, `"false"`)
3. Obyektlar va massivlar (`{}`, `[]`)
4. Funktsiyalar
5. ...

```

console.log(Boolean(true));           // true
console.log(Boolean(1));               // true
console.log(Boolean(-1));              // true
console.log(Boolean("hello"));         // true (non-empty string)
console.log(Boolean(" "));             // true (string with a space)
console.log(Boolean([]));              // true (empty array)
console.log(Boolean({}));              // true (empty object)

```

▼ Suhbat savollari

▼ Javascriptda type conversion nima? Bu type coercion'dan qanday farq qiladi?

Type Conversion qiymatni bir ma'lumot turidan boshqasiga o'zgartirish jarayoni. Bu jarayon dasturchi tomonidan maxsus funksiyalar yordamida amalga oshiriladi (masalan, `String()`, `Number()` va boshqalar).

Type Coercion esa avtomatik tarzda Javascript tomonidan amalga oshiriladi. Masalan: `"5" * 2` holatida string avtomatik raqamga o'zgartiriladi.

▼ Qiymatni boshqa turlarga qanday o'tkazasiz?

- **String:** `String()` yoki `.toString()` ishlatiladi.
- **Number:** `Number()`, `parseInt()`, yoki `parseFloat()` ishlatiladi.
- **Boolean:** `Boolean()` ishlatiladi.

▼ `parseInt()`, `parseFloat()`, va `Number()` ning farqi nima? Qaysi holatlarda ishlatiladi?

- `parseInt()` — string ichidan faqat **butun sonlarni** ajratadi va keyingi belgilarni hisobga olmaydi.

Masalan: `parseInt("123abc")` → `123`

- `parseFloat()` — string ichidan **o'nlik sonlarni** oladi.

Masalan: `parseFloat("123.45abc")` → `123.45`

- `Number()` — faqat to'liq stringni raqamga o'zgartiradi. Agar string noto'g'ri bo'lsa, `NaN` qaytaradi.

Masalan: `Number("123abc")` → `NaN`

▼ `toString()` metodi qanday ishlaydi? Misol keltiring

`toString()` har qanday ma'lumot turini stringga aylantirish uchun ishlatiladi.

```
const num = 42;  
console.log(num.toString()); // "42"  
  
const bool = true;  
console.log(bool.toString()); // "true"
```

▼ String va raqamni `+` operatori yordamida birlashtirsangiz, nima bo'ladi?

String va raqam `+` operatori yordamida birlashtirilganda, raqam stringga o'zgartiriladi va birlashtiriladi.

```
console.log(5 + "10"); // "510"  
console.log("Hello " + 5); // "Hello 5"
```

▼ Qiymatni boolean kontekstida ishlatganingizda qanday natija chiqadi?

Javascriptda quyidagi qiymatlar **falsey** bo'lib, `false` ga o'zgaradi:

- `false`, `0`, `-0`, `""` (bo'sh string), `null`, `undefined`, `NaN`, `0n`.

Boshqa barcha qiymatlar **truthy** bo'lib, `true` ga o'zgaradi.