







Frontender[1.0] JavaScript - Преобразование и приведение типов (type conversion & casting)

| | |
|--|---|
|  YouTube | https://youtu.be/YI-AUBPNxms |
|  Telegram | https://t.me/Dmitry_Kolotilshikov |
|  Github | https://github.com/DmitryKolotilshikov/ |
|  Boosty | https://boosty.to/dmitry_ko |
| # Номер урока | 6 |



Задачи к этому уроку тут https://boosty.to/dmitry_ko



== (двойное равно используется для сравнения двух значений на равенство) (сравнивается с приведением к одному типу)

| | true | false | 1 | 0 | -1 | "true" | "false" | "1" | "0" | "-1" | " " | null | undefined | Infinity | -Infinity | [] | {} | [[]] | [0] | [1] | NaN |
|-----------|------|-------|---|---|----|--------|---------|-----|-----|------|-----|------|-----------|----------|-----------|----|----|------|-----|-----|-----|
| true | | | | | | | | | | | | | | | | | | | | | |
| false | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | | | | |
| -1 | | | | | | | | | | | | | | | | | | | | | |
| "true" | | | | | | | | | | | | | | | | | | | | | |
| "false" | | | | | | | | | | | | | | | | | | | | | |
| "1" | | | | | | | | | | | | | | | | | | | | | |
| "0" | | | | | | | | | | | | | | | | | | | | | |
| "-1" | | | | | | | | | | | | | | | | | | | | | |
| " " | | | | | | | | | | | | | | | | | | | | | |
| null | | | | | | | | | | | | | | | | | | | | | |
| undefined | | | | | | | | | | | | | | | | | | | | | |
| Infinity | | | | | | | | | | | | | | | | | | | | | |
| -Infinity | | | | | | | | | | | | | | | | | | | | | |
| [] | | | | | | | | | | | | | | | | | | | | | |
| {} | | | | | | | | | | | | | | | | | | | | | |
| [[]] | | | | | | | | | | | | | | | | | | | | | |
| [0] | | | | | | | | | | | | | | | | | | | | | |
| [1] | | | | | | | | | | | | | | | | | | | | | |
| NaN | | | | | | | | | | | | | | | | | | | | | |

💡 **===** (тройное равно используется для сравнения двух значений на равенство) (**сравнивается по типу и по значению, без приведения типов**)

| | true | false | 1 | 0 | -1 | "true" | "false" | "1" | "0" | "-1" | " " | null | undefined | Infinity | -Infinity | [] | {} | [[]] | [0] | [1] | NaN |
|-----------|------|-------|---|---|----|--------|---------|-----|-----|------|-----|------|-----------|----------|-----------|----|----|-------|-----|-----|-----|
| true | | | | | | | | | | | | | | | | | | | | | |
| false | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | | | | |
| -1 | | | | | | | | | | | | | | | | | | | | | |
| "true" | | | | | | | | | | | | | | | | | | | | | |
| "false" | | | | | | | | | | | | | | | | | | | | | |
| "1" | | | | | | | | | | | | | | | | | | | | | |
| "0" | | | | | | | | | | | | | | | | | | | | | |
| "-1" | | | | | | | | | | | | | | | | | | | | | |
| " " | | | | | | | | | | | | | | | | | | | | | |
| null | | | | | | | | | | | | | | | | | | | | | |
| undefined | | | | | | | | | | | | | | | | | | | | | |
| Infinity | | | | | | | | | | | | | | | | | | | | | |
| -Infinity | | | | | | | | | | | | | | | | | | | | | |
| [] | | | | | | | | | | | | | | | | | | | | | |
| {} | | | | | | | | | | | | | | | | | | | | | |
| [[]] | | | | | | | | | | | | | | | | | | | | | |
| [0] | | | | | | | | | | | | | | | | | | | | | |
| [1] | | | | | | | | | | | | | | | | | | | | | |
| NaN | | | | | | | | | | | | | | | | | | | | | |

💡 **Использование === предпочтительнее**, так как оно предсказуемо и помогает избежать ошибок в программе, связанных с неочевидным преобразованием типов.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="script.js"></script>
  <title>Type conversion</title>
</head>
<body>
```

```
</body>
</html>
```

```
// Type conversion

let value = 10;

// число в строку
value = value + '';
value = `${value}`;
value = value.toString();
value = String(value);

// строку в число
value = Number(value);
value = +value;
value = parseInt(value);

value = '10.22';
value = parseFloat(value);

value = 10.29442313.toFixed(3);

// console.log(value);
// -----
// Casting

let val = 10;
val = 10 + '10'; // ('1010') string
val = 10 * '10'; // (100) number
val = 10 / '10'; // (1) number
val = 10 - '10'; // (0) number
val = 10 % '10'; // (0) number

val = 10 + true; // (11) true приводится в 1
val = 10 + false // (10) false приводтся к 0, так как это falsy значение
val = 10 + null // (10) null приводтся к 0, так как это falsy значение

val = 10 + undefined // (NaN)
val = 10 + NaN // (NaN)
val = 10 + [] // ('10') string
val = 10 + {} // (10[object Object]) string

val = true + true; // (2) number
val = true + false; // (1) number
val = false + false; // (0) number

// console.log(val);
// -----

// Операторы сравнения == и ===
// == сравнивается с приведением к одному типу
// === сравнивается по типу и по значению, без приведения типов
// Результат сравнения всегда true или false

let res;
res = true == true; // true
```

```
res = true === true; // true
res = true == 1; // true
res = true === 1; // false

res = false == 0; // true
res = false === 0; // false

res = 1 == '1'; // true
res = 1 === '1'; // false

res = null == undefined; // true
res = null === undefined; // false

res = '' == []; // true
res = '' === []; // false

// и тд, есть много комбинаций, смотри таблицы приведения типов в pdf к уроку

console.log(res);
```