Приведение типов в JavaScript

При сложении числа и строки JavaScript будет воспринимать число как строку.

JavaScript вычисляет выражения слева направо
Таким образом, разные последовательности могут привести к разным результатам:

Преобразование к числу toNumber()

В Javascript преобразование чисел, строк, объектов к числу (не обязательно целому, может быть и дробное) можно сделать с помощью функции `Number()`:

```
Number(myVar);
```

```
Number(5); //5
Number('0.25'); //0.25
Number('q5'); //NaN
Number('abc'); //NaN
Number(false); //0
Number(true); //1
```

Преобразование строки к целочисленному типу parseInt(string, radix);

parseInt(string, radix);

string - строковое представление числа

radix - основание системы счисления

Функция parseInt преобразует первый аргумент в число по указанному основанию, а если это невозможно - возвращает NaN.

Например, radix=10 даст десятичное число, 16 - шестнадцатиричное и т.п. Для radix>10 цифры после девяти представлены буквами латинского алфавита.

Если в процессе преобразования parseInt обнаруживает цифру, которая не является цифрой в системе счисления с основанием radix, например G вв 16-ричной системе или A в десятичной, то процесс преобразования тут же завершается и возвращается значение, полученное из строки на данный момент.

parseInt округляет дробные числа, т.к останавливается на десятичной точке.

Если radix не указан или равен 0, то javascript предполагает следующее:

- Если входная строка начинается с "0x", то radix = 16
- Если входная строка начинается с "0", то radix = 8. Этот пункт зависит от реализации и в некоторых браузерах (Google Chrome) отсутствует.
- В любом другом случае **radix=10**

Если преобразовать в число не удается, parseInt возвращает NaN

Преобразование к целочисленному типу parseInt(string, radix);

Важный факт, что функция позволяет использовать пробелы в начале и конце входной строки.

Кроме того `parseInt()` округляет дробные числа, потому что в работе останавливается на дробной десятичной точке.

```
parseInt(5); //5
parseInt(5.33); //5
parseInt(' 5.64'); //5
parseInt(true); //NaN
parseInt(false); //NaN
parseInt('abc'); //NaN
parseInt('5abc'); //5
parseInt('x5abc'); //NaN
parseInt('123bc',10) // 123
```

Булевый (логический) тип

Булевый тип (`boolean`) может принимать только два значения: `true` (истина) и `false` (ложь).

Такой тип, как правило, используется для хранения значений да/нет: `true` значит «да, правильно», а `false` значит «нет, не правильно».

```
let nameFieldChecked = true; // да, поле отмечено

let ageFieldChecked = false; // нет, поле не отмечено
```

Булевые значения также могут быть результатом сравнений:

```
let isGreater = 4 > 1;


alert( isGreater ); // true (результатом сравнения будет "да")
```

toBoolean

В Javascript преобразование типа к boolean можно сделать следующими способами:

!!myVar
Boolean(myVar)

Оба предложенных варианта **toBoolean** ("к булевскому") преобразуют свой аргумент в значение типа Boolean согласно следующей таблице:

Входной тип	**Результат**
Undefined	false
Null	false
Boolean	Результат совпадает с входным аргументом (преобразование не производится).
Number	Результат равен false, если аргумент равен 0 или NaN, иначе результат равен true.
String	Результат равен false, если аргумент является пустой строкой (его длина равна нулю), иначе результат равен true.
Object	true

Использование методов `prompt` и `alert` для создания игры «Угадай число»?

```
const number = 7;
let result = false;

while (!result) {
  const answer = prompt('Угадай число от 1 до 10?');
  if (answer === null) {
    break;
  }
  switch (+answer) {
    case number - 2:
    case number + 2:
      alert('Уже теплее!');
      break;
    case number - 1:
    case number + 1:
      alert('Горячо!');
      break;
    case number:
      alert('Ты угадал! Это число {$number}.');
      result = true;
      break;
    default:
      alert('Холодно!');
  }
  }
```