

Приведение типов в JavaScript

Junior Frontend Developer

Цель урока:

Вы узнаете принцип работы не строго типизированных языков программирования (как JavaScript) и научитесь конвертировать один тип данных в другой (например, строку в число).

Содержание урока:

1. Явное и неявное приведение типов к строке.
2. Явное и неявное приведение типов к числу.
3. Явное и неявное приведение типов к булевому значению.

Дополнительные материалы:

Неявное преобразование типов в JavaScript: <https://habr.com/ru/company/ru...>

Задание #1

Мы с вами уже обсуждали, что JavaScript имеет динамическую типизацию. В данном языке программирования мы можем преобразовывать данные только к 3-м типам: к number, boolean и string.

Для начала, создайте новые файлы: index.html и index.js. Не забудьте подключить index.js в index.html при помощи тега script.

Сейчас ваша задача состоит в том, чтобы создать 8 переменных с помощью let, которые будут представлять 8 разных типов данных. Каждую из этих переменных требуется явно преобразовать к 3-м типам: к number, boolean и string. Конечные значения выведите в консоль.

Пример:

```
let age = 18;  
console.log(Number(age), Boolean(age), String(age));
```

Обязательно после каждого console.log посмотрите, что у вас выводится в консоль и подумайте, почему получилось именно такое значение, которое было выведено.

Примечание: Symbol вы не сможете преобразовать к типу данных number. Если вы попытаетесь это сделать, то у вас будет ошибка.

Задание #2

В JavaScript существует 2 вида преобразований типов: явное и неявное.

Пример:

```
console.log(Number('18')); // явное
console.log(+ '18'); // неявное
```

Из них стоит использовать явное, так как оно уменьшает количество случайных багов.

Для начала, создайте новые файлы: task-2.html и task-2.js. Не забудьте подключить task-2.js в task-2.html при помощи тега script.

Ваша задача состоит в следующем. Ниже вам даны 10 примеров явного и неявного преобразования типов. Необходимо определить, какое преобразование явное, а какое неявное.

Если преобразование является явным, то напишите напротив примера комментарий “явное”, если неявным, то исправьте его на явное по шаблону ниже.

Шаблон ответа на пример console.log(+ '1'):

```
console.log(+ '1'); //неявное
console.log(Number('1'));
```

Также для каждого значения выведете его typeof.

Пример:

```
console.log(Number('10'), typeof Number('10')); // явное
//1
console.log(+ '50');
//2
console.log(Number('100'));
```

```
//3
console.log('' + 1);
//4
console.log(String(1));
//5
console.log(Boolean(0));
//6
console.log(+ '001');
//7
console.log(1 + '');
//8
console.log(Boolean(1));
//9
console.log(String(001));
//10
console.log(Number('Hello World'));
```

Задание #3

Когда мы пытаемся преобразовать к другим типам данных не совсем стандартные сущности, то у нас получаются иногда необычные значения.

Например, если мы преобразуем к строке функцию alert, то мы увидим следующее:

```
console.log(String(alert));  
// Возвращает function alert() { [native code] }
```

Сейчас ваша задача состоит в том, чтобы преобразовать явно к string, number и boolean следующие сущности, посмотреть на результат и проанализировать его:

1. console.log
 2. { name: 'Maxim' }
 3. Symbol('key')
 4. Number
 5. "
 6. 0
 7. -10
 8. '-105'
-

Задание #4

Вам даны 10 примеров явного преобразования типов.

Ваша задача понять, что будет выведено в консоль в каждом примере. Сначала запишите свои ответы в заметки, а только потом уже запускайте код и сверьтесь с ответами.

```
// 1  
console.log(Number(' 1 2 3 4 5'));  
// 2  
console.log(Number('1234 5'));  
// 3  
console.log(Number('12345'));  
// 4  
console.log(String(false));  
// 5  
console.log(Boolean(000000));  
// 6  
console.log(Boolean(0.0000001));  
// 7  
console.log(String(undefined));  
// 8  
console.log(Number('100f'));  
// 9  
console.log(Number('100'));  
// 10  
console.log(Number('000001'));
```