

# Строки и преобразование типов

**№ урока:** 4 **Курс:** JavaScript Starter

**Средства обучения:** Visual Studio Code  
Web Browser

## Обзор, цель и назначение урока

Научиться работать со строковым типом данных в JS, использовать конкатенацию. Понимать, что такое преобразование типов и правильно его использовать.

## Изучив материал данного занятия, учащийся сможет:

- Создавать строковые значения, используя разные подходы.
- Работать с конкатенацией.
- Работать с шаблонными строками.
- Преобразовывать строковые значения в числовые.
- Понимать принципы преобразования к типам Boolean и String.
- Использовать встроенную функцию prompt.

## Содержание урока

1. Что такое литерал, строковые литералы
2. Конкатенация
3. Шаблонные строки
4. Преобразование типов
5. Функция prompt

## Резюме

- **Литерал** – значение, явно указанное в коде.  
10 – числовой литерал;  
10n – числовой литерал (bigint);  
"hello" – строковой литерал;  
True – логический литерал;  
{ } – литерал объекта (подробнее будет рассмотрен в следующих уроках);  
[] – литерал массива (подробнее будет рассмотрен в следующих уроках).
- Для определения строковых переменных литерал может быть значением, взятым в одинарные или двойные кавычки.
- **Конкатенация** – операция сцепления строк. В JavaScript конкатенация выполняется с помощью оператора +.  
let fullName = firstName + " " + lastName; // fullName = "Ivan Ivanov"

- **Template literal (шаблонный строки)** – для создания строки на основе шаблона и значений, определенных в переменных или являющихся результатом выражений. Шаблонные строки создаются с использованием обратных кавычек (`).  

```
let fullName = `${firstName} ${lastName}`; // fullName = "Ivan Ivanov"
```

- **Преобразование типов** – преобразование значения одного типа в значение другого типа. Может происходить автоматически или с помощью специальных функций.  
`alert` – функция для отображения модального окна с сообщением. Данная функция принимает строковые значения.
- Преобразование типов может пригодиться в тех случаях, когда полученное значение не подходит для выполнения требуемой операции.  
Например, необходимо выполнить арифметическое сложение двух значений, полученных от пользователя. Значения, которые вводятся с клавиатуры, в сценарии будут доступны как значения типа `string` и их сложение будет выполняться как конкатенация. В такой ситуации может пригодиться использование преобразования типов.
- Самый простой способ преобразования типа — это вызов функции `String(value)`, `Number(value)` и `Boolean(value)`. При этом каждая функция вернет значение `value` соответствующего типа, если преобразование выполнилось успешно. Например, преобразовать значение «abc» в тип `number` не получится.
- **prompt** – встроенная функция, которая отображает диалоговое окно с полем ввода. Значение, введенное в поле ввода, будет возвращено функцией.  
Пример использования:  

```
let value = prompt("Введите значение", "Значение по умолчанию");
```

  
После выполнения такой функции введенное значение будет записано в переменную `value`.

### Закрепление материала

- Что такое литерал?
- Что такое конкатенация?
- Что такое шаблонные строки, в чем их преимущество перед конкатенацией?
- Что делает функция `prompt`?
- Какое назначение второго параметра функции `prompt`?
- Что такое преобразование типов, приведите пример, когда преобразование типов может пригодиться?

### Дополнительное задание

#### Задание

В файле `004 Strings and Data Conversion\08-simple-calc.html` доделайте логику калькулятора. Добавьте операции умножения, деления и вычитания.

Выведите результаты на экран двумя способами, через конкатенацию и через шаблонные строки.

### Самостоятельная деятельность учащегося

Выполните задания в директории Exercises\Tasks\04 Strings and Data Conversion в материалах к этому уроку.

### Рекомендуемые ресурсы

Форматирование текста в JavaScript

[https://developer.mozilla.org/ru/docs/Web/JavaScript/Guide/Text\\_formatting](https://developer.mozilla.org/ru/docs/Web/JavaScript/Guide/Text_formatting)