Frontendβ → JavaScript → Data types and operations → Strings and numbers

Theory: Strings and numbers

5 minutes 0 / 5 problems solved

Skip this topic Start practicing

1690 users solved this topic. Latest completion was about 1 hour ago.

Any information you want to use in scripts has its own type. The **type** describes how to store this data in memory and what operations apply to it.

Let's draw an analogy: types can be represented by animal species or any other abstract attributes designed to group objects by some common feature. All the kittens and tomcats that you see are of the type ***cats***, but each of them is a separate object. Thinking of a cat as a type, you can assume some available operations, for example, a cat can purr.

In this topic, we will consider two simple data types which are often used in programming.

§1. Strings

When working with textual information in your program, you will have to use **strings**. This type of data is extremely common in JavaScript. Strings are written in single or double quotes.

- Examples of strings in double quotes:

```
console.log("");              // empty string
console.log("string");        // one word
console.log("Hello, world"); // a phrase
```

- Examples of strings in single quotes:

```
console.log('a');    // single character
console.log('1234'); // a sequence of digits
```

Note that any digits in quotes will also be considered as a string.

As you can see, strings are very easy to use!

§2. Numbers

Numbers are the most important thing for any programmer. You will hardly be able to write a serious program without numbers, so let's see how to output a number to the console:

```
console.log(12);
console.log(0);
console.log(-11);
console.log(-1.04);
```

You can use positive, negative numbers and zeros. There are no additional difficulties in recording floating point numbers.

Integer numbers can be used to count physical objects, while floating-point numbers are a good choice for statistical and scientific calculations.

§3. *typeof* operator

We can easily recognize the data type using the `typeof` operator. Let's look at two examples to see how to output the data type to the console.

There are two ways to write this operator.

- With parentheses:

```
console.log(typeof(9)); // number
```

- Without parentheses:

Current topic:

Strings and numbers ···

Topic depends on:

✕ Multi-line programs ···

✕ Comments ···

Topic is required for:

Arithmetic operators ···

Boolean and logical operators ···

Introduction to Objects ···

Table of contents:

```
console.log(typeof 9); // number
```

Feedback & Comments

The result of these two code samples is the same: we want to find out what type of data `9` is, and it turns out to be a `number`.

This operator is especially useful when you are working with many different data types, because JavaScript can automatically convert data types to each other, which is not always useful, but we will talk about this in the following topics.

Report a typo

148 users liked this theory. **1** didn't like it. **What about you?**

😍 🙂 😐 🙁 😡

Start practicing

Comments (5) Hints (0) Useful links (0) Show discussion