Frontendβ → JavaScript → Introduction to JavaScript → Variables

Theory: Variables

🕓 5 minutes 0 / 5 problems solved

Skip this topic Start practicing

1691 users solved this topic. Latest completion was about 2 hours ago.

You can already work in JavaScript with data types, the value of which does not need to be changed. However, when writing large scale programs it is usually necessary to store data and results of intermediate calculations somewhere. For this purpose **variables** are used.

§1. What is a variable

A **variable** is a named place in memory where you can store data to access it later. It's like a box with a gift inside: this is also a kind of variable!

Every variable has a **name** to distinguish it from other variables. It is possible to access a value by its variable's name.

Variables are one of the most often used elements of programs, so it is important to understand and use them well.

§2. Variables declaration

Before you start working with a variable, you need to **declare** (in other words, create) it. JavaScript uses two keywords to create variables:

- `let` defines a **mutable variable** the value of which can be changed as many times as needed;
- `const` declares a **constant** whose value you want to forbid overwriting.

When you declare a variable, you must give it a name after one of these keywords. It is good practice to assign a variable name that describes its contents. Always try to choose meaningful and readable names for variables to make your code easy to understand.

§3. Mutable variables

Let's try to create a mutable variable and understand why it is called that. Let's call it `month` and put data into it using the `=` symbol:

```
let month = "November"; // define the variable and assign it a value
```

In variables, it is possible to store any type of data. In this case, we saved the string `"November"`. You can refer to this line if you use a variable name:

```
let month = "November";

console.log(month); // outputs the content of the variable to the console
```

This code outputs the contents of the variable into the console. In our case, the result will be the string `"November"` – you can check it yourself.

The case of a name is important: `month` is not the same as `Month`.

Now let's try to change the variable and output a new value to the console:

```
let month = "November";
month = "December";

console.log(month); // December
```

As you can see, the value of the variable has really changed. Now it stores the string `December`.

§4. Other ways to declare variables

Current topic:

Variables ···

Topic depends on:

✕ Multi-line programs ···

✕ Comments ···

Topic is required for:

Naming variables ···

Boolean and logical operators ···

Introduction to Objects ···

Template literals ···

Arrays ···

Increment and decrement ···

Table of contents:

In older scripts, you may see that `var` keyword is used instead of `let`. It looks like this:

```
var age = 21;
```

This is an outdated way of declaring a variable. The `var` keyword is not bad per se, but there is no need to use `var` in new projects. You can also find code when variables are declared without keywords:

```
age = 21;
```

It's not good practice either. In the future we will consider why it is a bad idea to declare variables without keywords or with `var`, but for now it is more useful to get acquainted with the constants.

§5. Constants

Declare the immutable variable called `language` and add the string `"JavaScript"` as a value:

```
const language = "JavaScript";
```

If we wanted to change the value of a variable created with the `const` keyword, we would have failed and got an error:

```
const language = "JavaScript";
language = "PHP"; // error
```

Now that you know about variables, you will be able to create more complex and interesting programs.

Report a typo

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

😍 🙂 😐 🙁 😡

Start practicing

Comments (4) Hints (0) Useful links (0) Show discussion