

Theory: Naming variables

🕒 4 minutes

0 / 5 problems solved

Skip this topic

Start practicing

1497 users solved this topic. Latest completion was 19 minutes ago.

As you know, each **variable** has a unique **name** that identifies it among other variables. Setting a good variable name may not be as easy as it seems. Experienced developers pay a lot of attention to names to make their programs easy to understand. It is important because programmers spend a lot of time reading and understanding code written by other developers. If the variables aren't named well, then even your own code may seem unclear to you after a few months. This is not good, because programming is not only about writing programs, but also about their further support and development.

In this topic, we will look more closely at how to select nice names for your variables according to the agreements and practices established in the JavaScript community.

§1. Restrictions

There are no specific requirements for characters that can be used in JavaScript to create names. However, there are two restrictions that you can easily remember: a name can only consist of letters, numbers and symbols `_`, `$`; and the name shouldn't start with a digit.

Look at the examples below:

```
1 let $testName; // right
2 let testName;  // right
3 let test_name; // right
4 let test18;    // right
5 let test;      // right
6
7 let 18test;    // wrong
8 let test-name; // wrong
```

A hyphen `-` is *not* a valid character for variable names in JS.

In JS, the case of the characters matters, so `firstName` and `firstname` will be different variables.

§2. Reserved names

Reserved names cannot be used when selecting a variable name. These are words that already perform specific functions in Javascript. These words include: *let*, *const*, *return*, and *function*. The complete list of words that are reserved can be found [on MDN](#).

The code below returns a syntactical error:

```
1 let const = 5;           // error
2 let return = "apple";    // error
```

§3. Naming conventions for variables

In addition, there are the following conventions for naming variables:

- The variable name must be readable and descriptive, and should explain to the reader what sort of values are stored. For example:

```
1 let count; // good name
2 let n;     // bad name
```

Coming up with such names — short and precise — comes with experience, but only if you put some work in it.

Current topic:

[Naming variables](#) ...

Topic depends on:

✗ [Variables](#) ...

Table of contents:

[1 Naming variables](#)

[§1. Restrictions](#)

[§2. Reserved names](#)

[§3. Naming conventions for variables](#)

[Feedback & Comments](#)

- Despite the validity of the underscore `_` symbol, most style guides recommend opting for camelCase to write a name consisting of several words. **camelCase** is a way of writing phrases where each new word in the middle of the phrase begins with a capital letter, without intermediate spaces or punctuation marks:

```
1 |
let serverMessage; // the variable name is written according to camelCase practice
```

These conventions are optional, but it is strongly recommended to follow them. As we mentioned at the beginning of this lesson, they make your code more readable for you and other JavaScript programmers.

 Report a typo

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



Start practicing

[Comments \(9\)](#)[Hints \(0\)](#)[Useful links \(0\)](#)[Show discussion](#)