

JAVASCRIPT VARIABLES

Abduvakhobova Robiyakhan Abdusamat kizi

3rd stage student of Andijan Mechanical Engineering Institute

E-mail: arobiyaxon274@gmail.com

*Scientific supervisor: **Abduvakhobov Abbasbek***

Intern teacher of the Department of Information Systems and Technologies of the Andijan Mechanical Engineering Institute

Annotation: This article provides information about variables in JavaScript. Often, a JavaScript application needs to work with data. Also, a chat application - data can contain users, messages, etc. To store this data, 'variables' are used. A variable is "named storage" for data. We can use variables to store data.

Key words: var, let, const, variable, user, function, block, types, expression, object.

Аннотация: В этой статье содержится информация о переменных в JavaScript. Часто приложению JavaScript необходимо работать с данными. Кроме того, приложение чата — данные могут содержать пользователей, сообщения и т. д. Для хранения этих данных используются «переменные». Переменная является «именованным хранилищем» для данных. Мы можем использовать переменные для хранения данных.

Ключевые слова: var, let, const, переменная, пользователь, функция, блок, типы, выражение, объект.

Annotatsiya: Ushbu maqola JavaScriptdagi o'zgaruvchilar haqida ma'lumot beradi. Ko'pincha JavaScript ilovasi ma'lumotlar bilan ishlashi kerak bo'ladi. Shuningdek u chat ilovasi - ma'lumotlar foydalanuvchilar, xabarlar va boshqalarni o'z ichiga olishi mumkin. Ushbu ma'lumotlarni saqlash uchun o'zgaruvchilar ishlatiladi. O'zgaruvchi ma'lumotlar uchun " nomli xotira " dir. Ma'lumotlarni saqlash uchun o'zgaruvchilardan foydalanishimiz mumkin.

Kalit so'zlar: var, let, const, o'zgaruvchi, foydalanuvchi, funktsiya, blok, tiplar, ifoda, obyekt.

Variables are not typed in JavaScript. After assigning a value of one type to a variable, there is no error and the variable can assume the new type. Because of this, JavaScript is sometimes said to be untyped. But this Java is mistyped and works on "dynamic typing".

To load from a variable, it must first be declared. JavaScript can do this in 3 different ways: using the var, let, or const keywords. Each of these for different reasons.

publish through var

Until ES2015, var was the only way to declare a variable.

```
var a = 0
```

If you forget to include the word `var`, you're assigning a value to an undeclared variable, and the result won't be what you expect. In modern environments or when accelerating strict mode, the above error occurs. In older environments (or if strict mode is disabled), initializes the variable and attaches it to the global object. For more information, initialization is the process of assigning an initial value to a variable.

If you don't initialize a variable when you declare it, it will set undefined as its security and keep it that way until you assign it a new value.

```
var a // typeof a === "undefined"
```

A variable can be declared more than once, provided that the prefix:

```
var a = 1
```

```
var a = 2
```

It is also possible to declare multiple variables in one line:

```
var a = 1, b = 2
```

There is a field of "scope" in programming, which can be translated into Uzbek as the field of view (IMHO). A viewport is a part of a program that can be seen and used independently.

A variable declared outside a function via the `var` field is attached to a global object and has global visibility. Such a variable can be used in the program view. If you declare a control inside a function via `var`, the variable is bound to the function and it only works inside the function, just like a function parameter.

If inside a function, a variable with the same name as a global variable is declared private, the private overrides the global variable.

It's important to note that when you create a block separated by `{ }` brackets, the scope is not created. A view scope is only created when you create a function, because a function view does not have a block view scope, but a `var`.

Help is available throughout the function of a variable declared inside a function. It's even possible to control the declaration at the end of the function after the function, because Java moves all the variables up, instead of compiling the JavaScript code. But to avoid confusion, it is better to declare the variable as a function.

to announce through.

As we said above, `var` block has no scope. `let` was added in ES2015 to solve this problem. The scope of the variable declared by `let` belongs only to the block in which it is declared and to the internal blocks in it.

Modern programmers are mostly moving away from `let` and abandoning `var` altogether.

Another difference is that when you declare a `let` on a function, unlike `var`, it doesn't make the variable a global variable.

Declare via `const`

Variables declared using var or let can improve the manual. A variable declared in const does not change after initialization, making it an immutable value, i.e. a constant.

```
const a = "test"
```

A help literal can be attached to a const. Another aspect: if an object is attached to the constant a, the values of the object can be changed through its functions.

const is a pointer modifier. has a block view area like let speed.

Modern programmers prefer to declare const variables on a per-program basis, as this avoids possible late-encounter errors.

Types

You may have heard that JavaScript is typeless. As I mentioned before, this is completely wrong. JavaScript has typing, and it works in the "dynamic typing" feature. This means you can assign any type to a variable. There are two kinds of types in JavaScript: simple and complex types.

Normal types

Common types include:

- Thigs
- Rows
- Boolean types

There are also two special types:

- null
- undefined

Let's look at each of them separately.

Thigs

There is only one type of number in JavaScript, and that is real numbers.

Numeric literals are written in the program as numbers and can be integer or real number literals depending on how they are written.

Integers

20

546984621654984

0xCC // 16-digit number

Actual numbers:

3.14

.1234

5.2e4

Rows

A string is a sequence of characters. In the program code, it is represented by a string literal and closed with quotation marks - "" or quotation marks ' '.

```
'series'
```

```
"text"
```

In JavaScript, the `\` character is used to represent strings on multiple lines.

```
"All is one \
series"
```

The `\` character can also be used to use ``` or `'''` between lines. This separates the above characters from line-opening begs.

```
"Uzbekistan"
```

Strings can be concatenated using the `+` operator.

Row template

Added in ES2015, it provides new possibilities for working with strings.

You can create a template by putting an expression between `${}` in the line. For example, without templates:

```
var a = 5
var b = 10
console.log ('The product is ' + (a * b) + '.')
// The product is 50
```

We write the same using the template:

```
var a = 5
var b = 10
console.log ('The product is ${a * b}.')
// The product is 50
```

Boolean type

JavaScript uses the keywords `true` or `false` to represent boolean types. The comparison operators (`===`, `==`, `<`, `>`) return one of the above two as the result.

Control operators like `if`, `while` use logical types in the program process.

A Boolean value determines an expression not only by whether it is exactly `true` or `false`, but also by whether it is `true` or close to `false`. For example, the value `false` is obtained because all of the following values are close to `false`:

```
0
-0
NaN
undefined
null
" // an empty string
```

In all other cases, `true` is obtained.

null

`null` is a special value indicating no value.

This value is found in almost all programming languages. For example, in Python it is found as `None`.

undefined

`undefined` means that the variable is not initialized and the value is `undefined`.

Returns undefined if the function has no return value. Also, if a function parameter exists, but no value is passed to the function when it is called, the function passes the parameters to undefined

The value is checked for undefined as follows:

```
typeof var === 'undefined'
```

Complex types

All types except those listed above are complex types. These are functions, arrays, and objects. Each of them has its own characteristics, and they also possess the properties of the object.

Expressions

The part of the code that creates a new value is called an expression.

There are several types of expressions in JavaScript.

Arithmetic expressions

Expressions used to calculate numbers include:

```
1 / 2
```

```
i++
```

```
i -= 2
```

```
i * 2
```

String expressions

Expressions executed on rows

```
'this' + 'line'
```

```
s += 'row'
```

Primary expressions

This type includes pointers, literals, and constants:

```
6
```

```
2.06
```

```
'this is something'
```

```
true
```

```
this //current object
```

```
undefined
```

```
i
```

Also some keywords

```
function
```

```
class
```

```
function* //function generator
```

```
yield
```

```
yield* //redirects to another generator or iterator
```

```
async function* //asynchronous function expression
```

```
wait
```

```
() //grouping
```

Array and object initialization expressions

```
[] //array literal
```

```
{}
```

```
[1, 2, 3]
```

{a: 1, b: 2}

{a: {b: 1}}

Logical expressions

Boolean expressions create a logical value using operators

a && b

a || b

a

Left-hand expressions

new // creates a new model from the constructor

super // calls the parent constructor

...obj

Property reference expressions

object.property

object[property]

object['property']

Object creation expressions

new object ()

new a (1)

Function definition expressions

Function () {}

Function (a, b) {return a * b}

(a, b) => a * b

a => a * 5

() => {return 5}

Function call expressions

a.x(5)

window.resize()

Referenses:

1. M. Aripov, M. Muhammadiyev. Informatics, information technologies. Textbook. T: TDYuI, 2004.
2. Hoshimov O.O, Tulyaganov M.M. Computer and digital technologies Tashkent new century generation-2009. Page 103.
3. Jeffrey Brandebau.JAVA SCRIPT. "Collection of recipes for professionals". St. Petersburg.2010.
4. SH.A.Nazirov, G.Divald .Fundamentals programming. "Tashkent"-2007
5. M. ARIPOV, S. DOTTOYEV, M. FAYZIYEVA. "Web technologies".T: Publishing House of the National Society of Philosophers of Uzbekistan-2013