





# Frontender[1.0] JavaScript Старт, история, популярность, первый код

 YouTube	<a href="https://youtu.be/IIZhSuPuaMw">https://youtu.be/IIZhSuPuaMw</a>
 Telegram	<a href="https://t.me/Dmitry_Kolotilshikov">https://t.me/Dmitry_Kolotilshikov</a>
# Номер урока	1



**JavaScript (JS)** — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили.

Основные архитектурные черты:  
динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.



## Почему JavaScript самый популярный язык программирования?

- 1. Универсальность:** JavaScript используется как для **frontend**-разработки (веб-разработки) (создание интерактивных страниц), так и для **backend**-разработки (веб-серверы, API), а также для **мобильной разработки** (React Native), **разработки настольных приложений** (Electron), для создания **игр, 3D-графики и др.**
- 2. Простота изучения:** JavaScript имеет относительно простой синтаксис, что делает его доступным для начинающих программистов.
- 3. Востребованность:** На рынке труда существует огромный спрос на JavaScript-разработчиков.
- 4. Поддержка:** JavaScript поддерживается всеми современными браузерами, что делает его идеальным языком для создания веб-приложений.
- 5. Широкое сообщество:** JavaScript имеет большое и активное сообщество разработчиков, которое постоянно создает новые библиотеки и инструменты.
- 6. Инновации:** JavaScript постоянно развивается и обновляется, что делает его актуальным и востребованным.
- 7. Гибкость:** JavaScript позволяет создавать как простые, так и сложные приложения.
- 8. Производительность:** JavaScript может быть очень быстрым, если его правильно использовать.
- 9. Масштабируемость:** JavaScript-приложения могут быть легко масштабированы для поддержки большого количества пользователей.
- 10. Бесплатность:** JavaScript является бесплатным и открытым исходным кодом, что делает его доступным для всех.

### Вывод:

JavaScript является самым популярным языком программирования благодаря своей универсальности, простоте изучения, востребованности, поддержке, большому сообществу, постоянному развитию, гибкости, производительности, масштабируемости и бесплатности.

<https://www.tiobe.com/tiobe-index/>

<https://survey.stackoverflow.co/2023/#learning-to-code-learn-code-learn>

<https://www.orientsoftware.com/blog/what-can-you-do-with-javascript>

<https://ru.wikipedia.org/wiki/JavaScript>



## Что может JavaScript делать со страницей?

- DOM (Document Object Model) манипуляции. Изменение содержимого страницы
- Отправлять запросы на сервер (AJAX, Асинхронщина)
- Обработать события (клики мышью, нажатия клавиш, обработка полей ввода и тп)
- Хранение данных (local/session storage, cookie, indexedDB)
- Манипуляции с данными (map, filter, sort, add, drop, и тд)
- Создании и управление анимациями с помощью JS
- Работа с геолокацией и камерой
- Управление аудио и видео

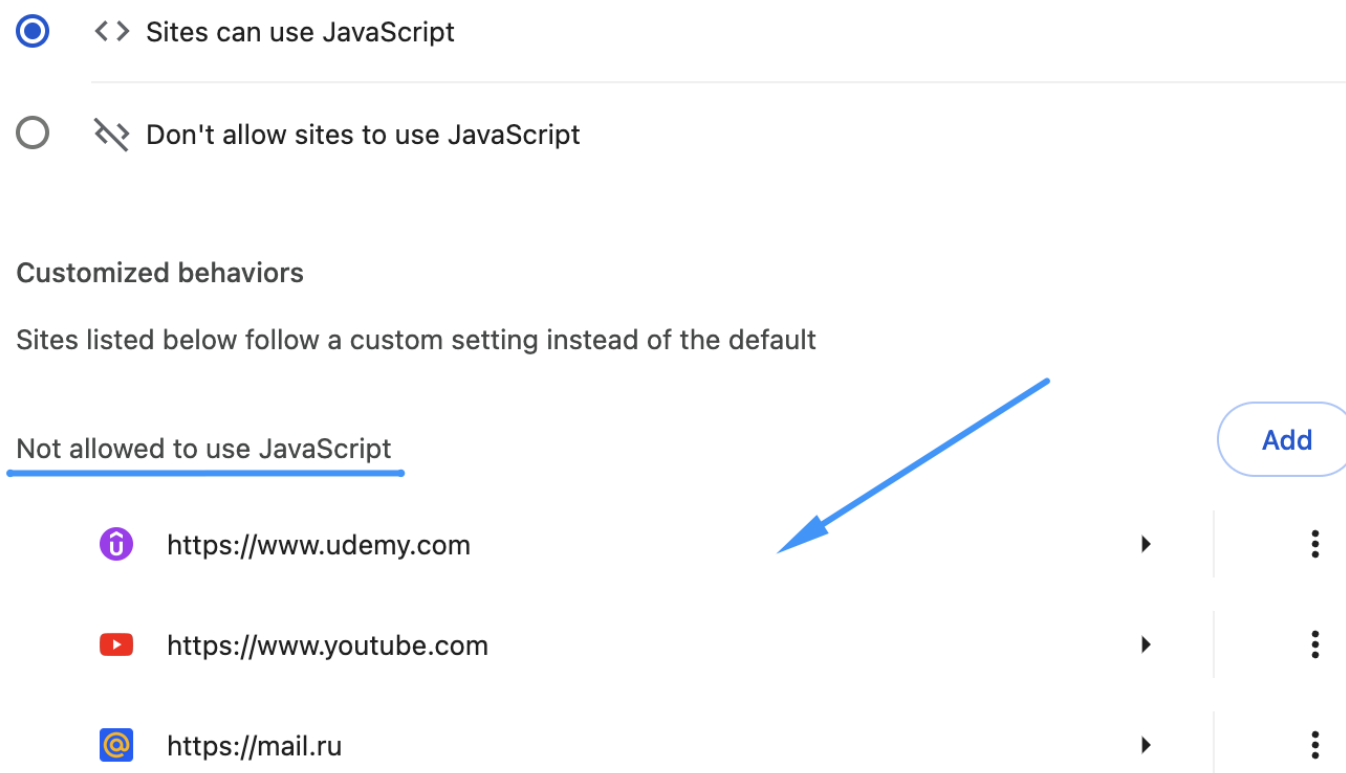


### Из возможных минусов/нюансов JavaScript

- Не имеет строгой типизации, что может привести к ошибкам во время выполнения.
- Некоторые функции JavaScript могут быть несовместимы с устаревшими браузерами.
- Большая вариативность действий.

## Если отключить JavaScript?

Путь: **chrome://settings/content/javascript**



JavaScript считается **динамически интерпретируемым языком** с применением компиляции **Just-In-Time (JIT)**.

Вот более подробное объяснение:

- **Интерпретация:** В основном JavaScript-код интерпретируется, то есть выполняется построчно непосредственно браузером во время выполнения.
- **JIT-компиляция:** Однако современные браузеры используют JIT-компиляцию для улучшения производительности. JIT-компилятор переводит часто используемые части JavaScript-кода в машинный код для более быстрого выполнения.

В итоге JavaScript можно назвать **смешанным языком**, так как он сочетает в себе элементы интерпретации и компиляции.



## Краткая история

-

**1995** Самая первая реализация JavaScript была создана Бренданом Эйхом (англ. *Brendan Eich*) в компании Netscape. Помимо Брендана Эйха, в разработке участвовали сооснователь Netscape Communications **Марк Андрессен** и сооснователь Sun Microsystems **Билл Джой**. Вначале язык назывался **Mocha**. Язык был реализован Бренданом Эйхом в течение десяти дней и впервые был включен в пре-альфу версию Netscape 2. Затем он был переименован в LiveScript и предназначался как для программирования на стороне клиента, так и для программирования на стороне сервера (там он должен был называться LiveWire)

### -конец 1995

На синтаксис оказали влияние языки **Си** и **Java**, и, поскольку Java в то время было модным словом, **4 декабря 1995** года LiveScript переименовали в **JavaScript**

- **1997** По инициативе компании Netscape была проведена стандартизация языка ассоциацией ECMA. Стандартизированная версия имеет название ECMAScript, описывается стандартом ECMA-262. Первой версии спецификации соответствовал **JavaScript версии 1.1**

### EcmaScript (или ECMAScript)

- это стандарт, который определяет язык программирования JavaScript (JS). Стандарт разрабатывается организацией **Ecma International** и определяет синтаксис, типы данных, операторы, объекты и другие элементы, необходимые для написания программ на JS.

Каждая новая версия **EcmaScript** включает в себя новые функциональные возможности, улучшения и исправления ошибок предыдущих версий.

JavaScript является реализацией стандарта **EcmaScript**, поэтому когда мы говорим о JavaScript, мы часто имеем в виду конкретную реализацию этого стандарта.

[https://www.w3schools.com/js/js\\_es6.asp](https://www.w3schools.com/js/js_es6.asp)

<https://ecma-international.org/publications-and-standards/standards/ecma-262/>

<https://tc39.es/ecma262/>

### - 2009 ECMAScript (ES5)

был выпущен с множеством фич.

### - 2015 ECMAScript 2015 (ES6)

Самое крупное обновление языка. Много изменений и нововведений в язык, например: стрелочные функции, классы, модули и т. д. Плюс ко всему было решено выпускать ежегодные обновления, чтобы избежать большого количества изменений в каждом обновлении.

-

**2016** ES2016, ES2017, ES2018, ES2019, ....

## КЛАССНО ТО, ЧТО В JS

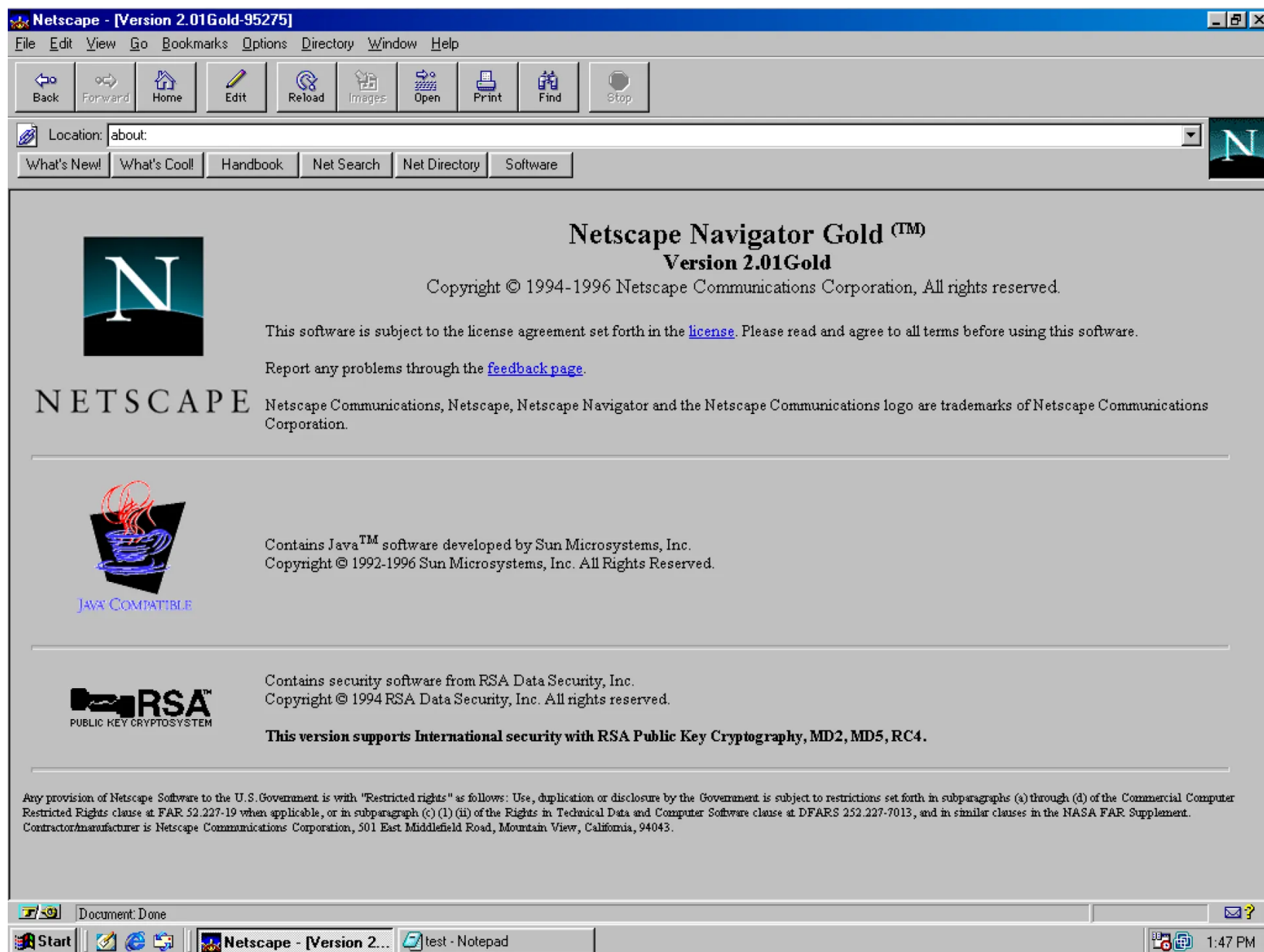
- старая функциональность никогда не удаляется;
- версии языка не полностью новые, а просто обновления/улучшения;
- сайты не ломаются и могут работать со старой функциональностью (сохраняется совместимость);

## JavaScript и Java [ [править](#) | [править код](#) ]

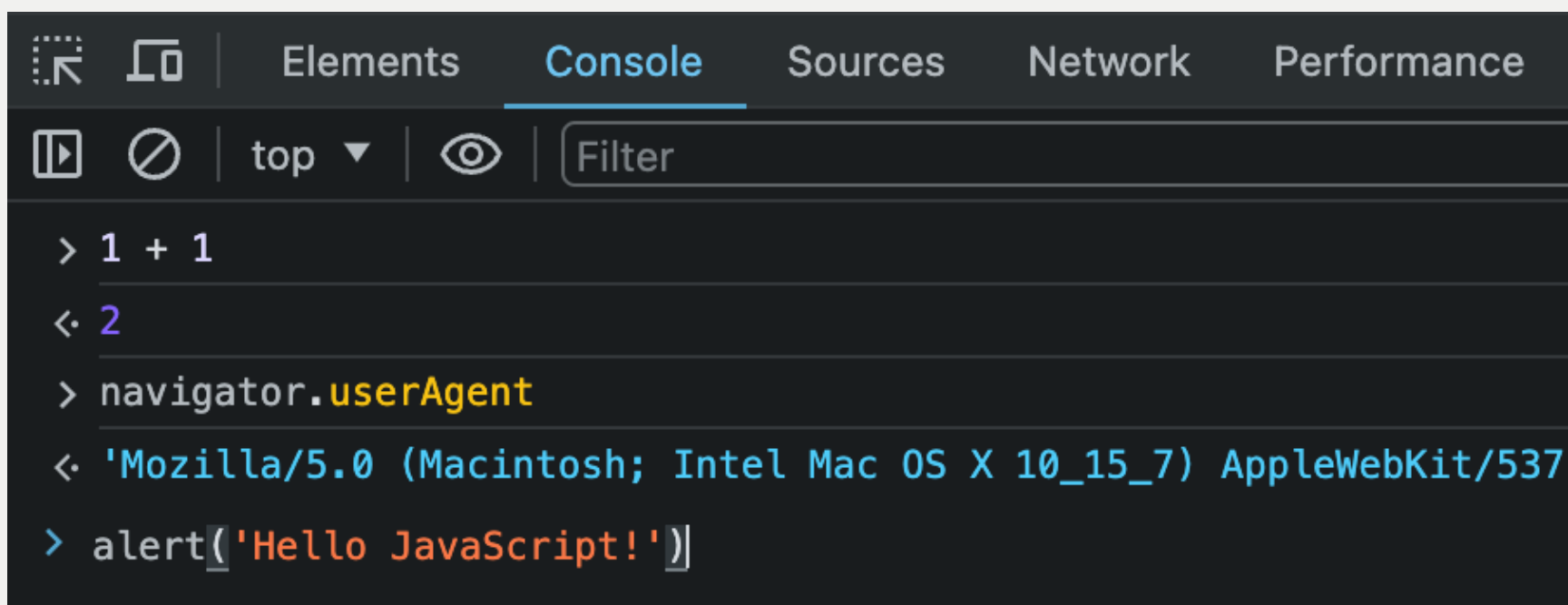
Общим заблуждением является то, что JavaScript аналогичен или тесно связан с [Java](#), это не так<sup>[30]</sup>. Оба языка имеют C-подобный синтаксис, являются объектно-ориентированными и, как правило, широко используются в клиентских веб-приложениях. Из важных различий можно отметить:

- Java реализует ООП подход, основанный на [классах](#), JavaScript — на [прототипах](#);
- Java имеет [статическую типизацию](#), JavaScript — [динамическую типизацию](#);
- Java загружается из скомпилированного байт-кода; JavaScript интерпретируется напрямую из файла (но часто с незаметной [JIT-компиляцией](#)).

Первая публичная версия JavaScript была встроена в



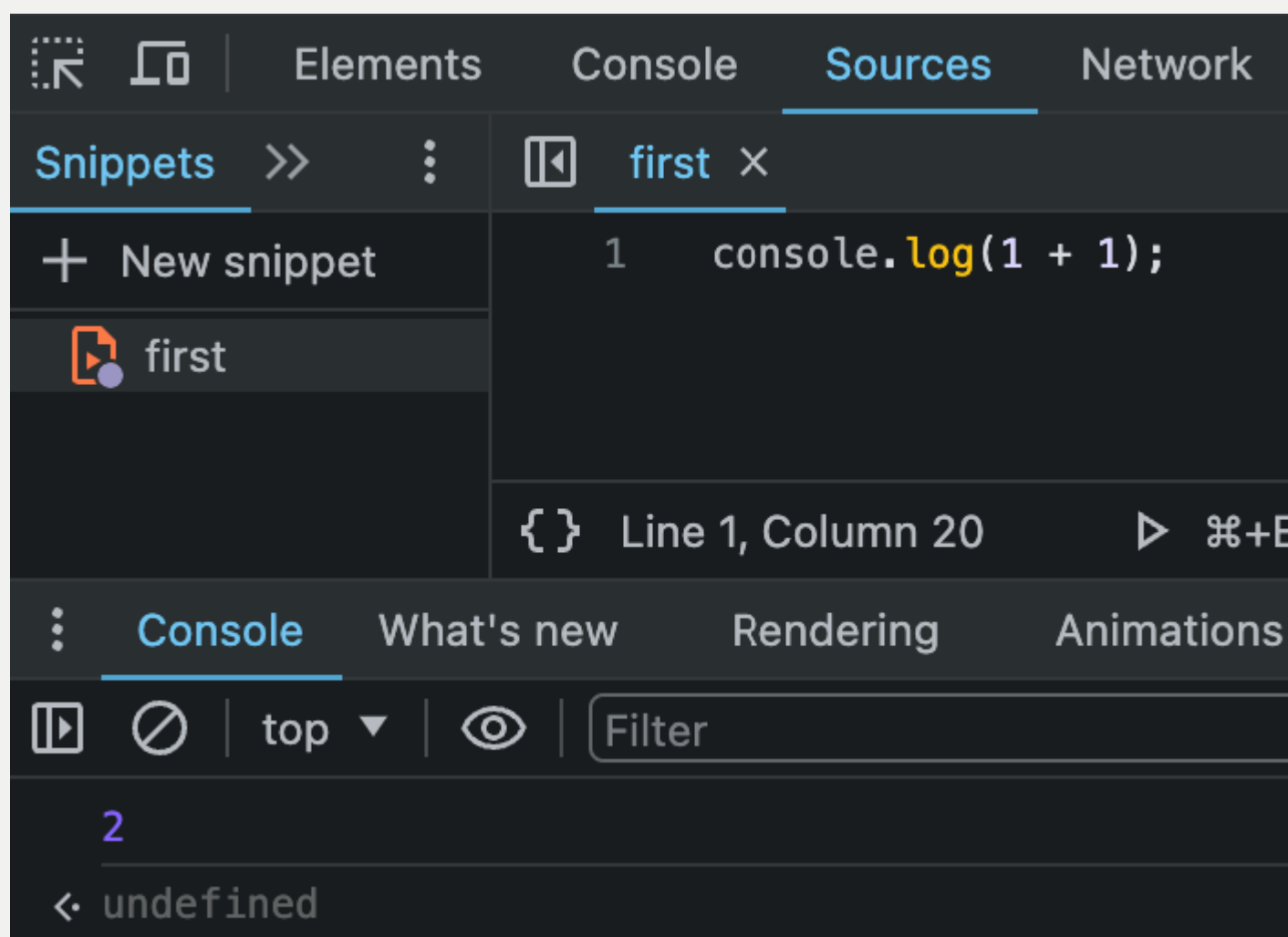
💡 ПЕРВЫЙ КОД (в Console, DevTools)



The screenshot shows the Chrome DevTools Console with the 'Console' tab selected. The interface includes a toolbar with icons for opening the console, disabling it, a dropdown menu set to 'top', a visibility icon, and a 'Filter' input field. The console log shows the following sequence of commands and results:

```
> 1 + 1  
< 2  
> navigator.userAgent  
< 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537...'  
> alert('Hello JavaScript!')
```

💡 ПЕРВЫЙ КОД (в Sources - Snippets, DevTools)



The screenshot shows the Chrome DevTools 'Sources' panel with the 'Snippets' sub-panel selected. The 'first' snippet is open, showing a single line of code: `console.log(1 + 1);`. The snippet is named 'first' and has a small icon next to it. The console at the bottom shows the result of the execution: `2`. The console also shows `undefined` for the next line, indicating the end of the execution.