



🏠 → [The JavaScript language](#) → [An introduction](#)

📅 2nd September 2019

Manuals and specifications

This book is a *tutorial*. It aims to help you gradually learn the language. But once you're familiar with the basics, you'll need other sources.

Specification

[The ECMA-262 specification](#) contains the most in-depth, detailed and formalized information about JavaScript. It defines the language.

But being that formalized, it's difficult to understand at first. So if you need the most trustworthy source of information about the language details, the specification is the right place. But it's not for everyday use.

A new specification version is released every year. In-between these releases, the latest specification draft is at <https://tc39.es/ecma262/>.

To read about new bleeding-edge features, including those that are “almost standard” (so-called “stage 3”), see proposals at <https://github.com/tc39/proposals>.

Also, if you're in developing for the browser, then there are other specs covered in the [second part](#) of the tutorial.

Manuals

- **MDN (Mozilla) JavaScript Reference** is a manual with examples and other information. It's great to get in-depth information about individual language functions, methods etc.

One can find it at <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference>.

Although, it's often best to use an internet search instead. Just use “MDN [term]” in the query, e.g. <https://google.com/search?q=MDN+parseInt> to search for `parseInt` function.

- **MSDN** – Microsoft manual with a lot of information, including JavaScript (often referred to as JScript). If one needs something specific to Internet Explorer, better go there: <http://msdn.microsoft.com/>.

Also, we can use an internet search with phrases such as “RegExp MSDN” or “RegExp MSDN jscript”.

Compatibility tables

JavaScript is a developing language, new features get added regularly.

To see their support among browser-based and other engines, see:

- <http://caniuse.com> – per-feature tables of support, e.g. to see which engines support modern cryptography functions: <http://caniuse.com/#feat=cryptography>.

- <https://kangax.github.io/compat-table> – a table with language features and engines that support those or don't support.

All these resources are useful in real-life development, as they contain valuable information about language details, their support etc.

Please remember them (or this page) for the cases when you need in-depth information about a particular feature.

[Previous lesson](#)[Next lesson](#)

Share

[Tutorial map](#)

Comments

- If you have suggestions what to improve - please [submit a GitHub issue](#) or a pull request instead of commenting.
- If you can't understand something in the article – please elaborate.
- To insert a few words of code, use the `<code>` tag, for several lines – use `<pre>` , for more than 10 lines – use a sandbox ([plnkr](#), [JSBin](#), [codepen](#)...)