tc39-2014-015.docTC-39 Process

The ECMA TC-39 committee is responsible for evolving the ECMAScript programming language and authoring the specification. The committee operates by consensus and has discretion to alter the specification as it sees fit. However, the general process for making changes to the specification is as follows.

Development

Changes to the language are developed by way of a process which provides guidelines for evolving an addition from an idea to a fully specified feature, complete with acceptance tests and multiple implementations. There are four "maturity" stages. The TC-39 committee must approve acceptance for each stage.

Maturity Stages

	Stage	Purpose	Criteria	Acceptance signifies	Spec quality	Post-acceptance changes expected	Implementation types expected
0	***Strawman***	● *Allow input into the specification.*	● *None*	*N/A*	*N/A*	*N/A*	*N/A*
1	**Proposal**	● Make the case for the addition ● Describe the shape of a solution ● Identify potential challenges	● Identified "champion" who will advance the addition ● Prose outlining the problem or need and the general shape of a solution. ● Illustrative examples of usage ● High-level API ● Discussion of key algorithms, abstractions and semantics ● Identification of potential "cross-cutting" concerns and implementation challenges/complexity.	The committee expects to devote time to examining the problem space, solutions and cross-cutting concerns	None	Major	Polyfills / demos
2	**Draft**	● Precisely describe the syntax and semantics using formal spec language	● Above ● Initial spec text	The committee expects the feature to be developed and eventually included in the standard	Draft: all *major* semantics, syntax and API are covered, but TODOs, placeholders and editorial issues are expected	Incremental	Experimental
3	**Candidate**	● Indicate that further refinement will require feedback from implementations	● Above ● Complete spec text ● Designated reviewers have signed off on the current spec text. ● The ECMAScript editor has signed off on the current spec text.	The solution is complete and no further work is possible without implementation experience, significant usage and external feedback.	Complete: all semantics, syntax and API are completed described	Limited: only those deemed critical based on implementation experience	Spec compliant
4	**Finished**	● Indicate that the addition is ready for inclusion in the formal ECMAScript standard	● Above ● Test 262 acceptance tests have been written for mainline usage scenarios. ● Two compatible implementations which pass the acceptance tests. ● The ECMAScript editor has signed off on the current spec text.	The addition will be included in the soonest practical standard revision	Final: All changes as a result of implementation experience are integrated.	None	Shipping

Input into the process

Ideas for evolving the ECMAScript language are accepted in any form. Any discussion, idea or proposal for a change or addition which has not been submitted as a formal proposal is considered to be a "strawman" (stage 0) and has no acceptance requirements. Such submissions must either come from members of TC-39 or from non-members who have registered via ECMA International.

Spec revisions and scheduling

TC-39 *may* submit for approval to the Ecma General Assembly a new Edition of the ECMAScript language in March and September of every year. Additions which have been accepted by the committee as "finished" (stage 4) *may* be included in a new Edition.

Status of in-process additions

TC-39 will maintain a list of in-process additions, along with the current maturity stage of each, on its website.

Spec Text

At stages "draft" (stage 2) and later, the semantics, API and syntax of an addition must be described as edits to the latest published ECMAScript standard, using the same language and conventions. The quality of the spec text expected at each stage is described above.

Calls for implementation and feedback

When an addition is accepted at the “candidate” (stage 3) maturity level, the committee is signifying that it believes design work is complete and further refinement will require implementation experience, significant usage and external feedback.

Reviewers

Anyone can be a reviewer and submit feedback on an in-process addition. The committee may identify designated reviewers for acceptance at the “candidate” maturity stage. Designated reviewers should not be authors of the spec text for the addition and should have expertise applicable to the subject matter.

Eliding the process

The committee may elide the process based on the scope of a change under consideration as it sees fit.

Role of the editor

In-process additions will likely have spec text which is authored by a champion or a committee member other than the editor although in some case the editor may also be a champion with responsibility for specific features. The editor is responsible for the overall structure and coherence of the ECMAScript specification. It is also the role of the editor to provide guidance and feedback to spec text authors so that as an addition matures, the quality and completeness of its specification improves. It is also the role of the editor to integrate additions which have been accepted as “finished” (stage 4) into the a new revision of the specification.