

History of Modules and the TC39

Introduced in ES6 (2015)

- <http://www.ecma-international.org/ecma-262/6.0/#sec-modules>
- Loader pulled out to separate specification prior to ratification
- Attempts to specify have been made in whatwg
- <https://github.com/whatwg/loader> (status: “good chunk is outdated”)
- Early loader draft discussed during harmony planning - [ECMAScript wiki archive: Module Loaders](#), which became <https://github.com/jorendorff/js-loaders/blob/e60d3651/specs/es6-modules-2013-12-02.pdf>, later moving into the WhatWG spec

Proposal for Pragma

- <https://github.com/tc39/proposal-modules-pragma>
- Rejected
- We could still implement, was implying that it could be revisited if it was found to exist in wild
- <https://github.com/tc39/tc39-notes/blob/master/es8/2017-07/jul-26.md#9ivb-modulescript-pragma-for-stage-2>

Unambiguous Grammar

- <https://github.com/rwaldron/tc39-notes/blob/6e244b58b8678d0fe104bb7b51f56d3a27aff5c/es7/2017-01/jan-25.md#13iia-proposed-grammar-change-to-es-modules>
- Rejected by different tc39 members for different reasons - some prefer a pragma, some prefer out of band, some prefer import options - punted
- <https://github.com/bmeck/UnambiguousJavaScriptGrammar>
- Prohibits “module by default” in the future

Dynamic Module Reform

- <https://github.com/tc39/proposal-dynamic-modules>
- Rejected
- <https://github.com/tc39/tc39-notes/blob/master/es7/2016-11/nov-30.md#12iia-proposal-to-reform-the-spec-to-solve-nodejs-ecosystem-compatibility-breaks-w-es-modules>

Nested imports

- <https://github.com/benjamn/reify/blob/master/PROPOSAL.md>
- was very interesting but it needs to be done in async contexts and means that `import` becomes even more prevalent

- Currently stalled because it seems to imply synchronous module evaluation

Module Import Options

- <https://github.com/tc39/tc39-notes/blob/master/es8/2017-05/may-23.md#16iib-module-import-options-discussion-potentially-for-stage-1>
- Discussion on in-band data on import statements
- Recommended to use query parameters for now, no need to advance

Import Ordering Clarification

- <https://github.com/rwaldron/tc39-notes/blob/b8da60318b564f136cbe8385f17f42abc0666cdd/es7/2016-09/sept-28.md#es-modules-lifecycle>
- No normative spec text accepted (a sticky point given caching), general consensus that ordering should make sense

Import.meta

- <https://github.com/tc39/proposal-import-meta>
- Module-local namespace for host-specific data
- Accepted
- WHATWG spec adds `import.meta.url`: <https://github.com/whatwg/html/pull/3141>

multi module grammar

- <https://www.keithcirkel.co.uk/proposal-multi-module-files/>
- hasn't been presented but has been scheduled
- Allows "bundles"

top level await

- <https://github.com/MylesBorins/proposal-top-level-await>
- Potentially specifies parts of the loader
- Removes notion of ESM as being synchronous
- Inline module alternative, may be similar to multi-modules?:
<https://github.com/MylesBorins/proposal-top-level-await/issues/5>

JS builtin modules

- Meeting later this week for TC39
- Temporal is stalled because of it