



# JavaScript Standard Library aka Built-in Modules

for stage 2

Michael Saboff  
Mattijs Hoitink  
Mark Miller

# Agenda

- Proposal Status
- Responses to June 2019 Advancement Objections
  - Adding ability to import from scripts
  - Unified Namespace
- Stage 2

# Current Status

Current Proposal:

<https://github.com/tc39/proposal-javascript-standard-library>

# Request to add import from Scripts

- This proposal builds upon the currently specified module semantics.
  - ▶ We didn't want to add any new surface area.
  - ▶ Imports are currently not allowed in Scripts.
  - ▶ Importing from scripts is orthogonal to JS Standard Libraries.
  - ▶ Champions have no desire to expand this proposal to importing from scripts as it opens up a whole new set of issues.
  - ▶ Believe that importing from scripts should be a separate proposal.
  - ▶ Have no desire to tie this proposal to a Import from Script proposal.

# Unified Standard Namespace

- Strong objection to move forward unless there was one unified namespace for browsers.
  - ▶ There will likely be several namespaces. (Hopefully not too many.)
  - ▶ Standards bodies will likely want their functionality in a dedicated namespace, e.g. TC-39, TC-53, W3C, ...
  - ▶ Other namespaces will likely have multiple stakeholders, e.g web components agreed upon by both W3C and Node Foundation

# Two Namespace Classes

## Standard Managed

Governed by one or more stakeholders

Openly communicated governance

Identifiers widely communicated and reserved in central repository

Name correspond to functional areas

Modules are standardized

Included modules for wide adoption

Intended for large development community

## Vendor / adhoc Managed

Informal or localized governance

Private / adhoc governance

Identifiers chosen from unused but preferably reserved in central repository

Name based on vendor criteria

Module inclusion based on adhoc criteria

Included module at vendors discretion

For vendor or product specific developers

# Our Namespace Proposal

- TC-39 is responsible for the content of the **js:** namespace.
  - ▶ Outside parties are free to propose additions.
  - ▶ TC-39 uses stage process to approve modules for the **js:** namespace
- We, TC-39, set up a repository for additional namespaces and share that repository with other stakeholders
  - ▶ We initially reserve generic and specific namespaces, then open up to other interested parties
  - ▶ The initial reserved set could include names like **std:**, **lib:**, **web:**, **css:**, etc

# Unified Standard Namespace for Browsers

- Chrome or any other browser or other hosts are free to use their own namespace and include modules they choose.



Questions?

Stage 2?

Thank you!