

The jQuery Core team receives many requests to add features to the library. Here are some of the criteria we use in our decisions on whether to add a feature:

- **Does it require access to jQuery internals?** If the feature needs information that only jQuery has, for example the list of events attached to an element, it is a candidate for feature addition.
- **Can jQuery itself benefit from the feature?** Features adding useful APIs that also improve the file size or performance inside jQuery itself are good candidates.
- **Is the feature related to DOM operations?** jQuery's primary mission is to be a DOM library. Features beyond that are often better served by libraries such as [lodash](#) for utilities or [Raphaël](#) for SVG.
- **Is the need already served by a plugin?** If there is a jQuery plugin that already does the job well, there is little reason for us to add duplicate functionality inside the library. Simply use the plugin.
- **Does it change an existing API?** Adding new flags or signatures to existing APIs can break code that duck-punches the API. New arguments also complicate documentation and can create [confusing Boolean traps](#).
- **Does it support best practice?** jQuery wants to make it easy to write good and fast code, so new features should follow that pattern. jQuery 1.9 actually [removed many features](#) that lead to brittle or slow code.