

globalThis

Published 16 July 2019 · tagged with ECMAScript

If you've written JavaScript for use in a web browser before, you may have used `window` to access the global `this`. In Node.js, you may have used `global`. If you've written code that must work in either environment, you may have detected which of these is available, and then used that — but the list of identifiers to check grows with the number of environments and use cases you want to support. It gets out of hand quickly:

```
// A naive attempt at getting the global `this`. Don't use this!
const getGlobalThis = () => {
  if (typeof globalThis !== 'undefined') return globalThis;
  if (typeof self !== 'undefined') return self;
  if (typeof window !== 'undefined') return window;
  if (typeof global !== 'undefined') return global;
  // Note: this might still return the wrong result!
  if (typeof this !== 'undefined') return this;
  throw new Error('Unable to locate global `this`');
};
const theGlobalThis = getGlobalThis();
```

For more details on why the above approach is insufficient (as well as an even more complicated technique), read [a horrifying globalThis polyfill in universal JavaScript](#).

[The globalThis proposal](#) introduces a *unified* mechanism to access the global `this` in any JavaScript environment (browser, Node.js, or something else?), regardless of the script goal (classic script or module?).

```
const theGlobalThis = globalThis;
```

Note that modern code might not need access to the global `this` at all. With JavaScript modules, you can declaratively `import` and `export` functionality instead of messing with global state. `globalThis` is still useful for polyfills and other libraries that need global access.

globalThis support

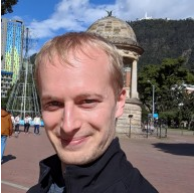
71+

65+

12.1+

12+

[about this feature support listing](#)



Posted by Mathias Bynens ([@mathias](#)).

Retweet this article!

[Branding](#) · [Terms](#) · [Privacy](#) · [Twitter](#) · [Edit this page on GitHub](#)

Except as otherwise noted, any code samples from the V8 project are licensed under [V8's BSD-style license](#). Other content on this page is licensed under [the Creative Commons Attribution 3.0 License](#). For details, see [our site policies](#).