

# Technical values and their priorities

The project uses these technical values to establish priorities and guide collaboration.

These are the shared values as of this writing and will evolve. We hope they are useful to people new to the project in order to better understand which contributions will be aligned with the current direction and as thinking points when trading off between conflicting goals.

The factors influencing every discussion/decision are different and priority 1 does not always trump priority 2 and so on.

## Values and priority level

- Priority 1 - Developer experience
- Priority 2 - Stability
- Priority 3 - Operational qualities
- Priority 4 - Node.js maintainer experience
- Priority 5 - Up to date technology and APIs

## Value descriptions

### 1 - Developer experience

We value ensuring that developers are productive and enjoy developing with Node.js. Some key elements of this include:

- Approachability (both technical and community)
- Great documentation
- Bundling friction-reducing APIs and components, even though they could be provided externally
- Compatibility and interoperability with browsers and other JavaScript environments so that as much code as possible runs as is both in Node.js and in the other environments
- Enabling/supporting external packages to ensure overall developer experience

### 2 - Stability

Whenever possible, we seek to ensure that working code continues to work. To keep the trust of developers and users, we value stability. Some key elements of this include:

- Backward compatibility
- Stable releases on a predictable schedule
- A strong safety net, including testing how changes in Node.js affect popular packages
- Careful consideration of what goes into Long Term Support (LTS) releases

### 3 - Operational qualities

We value keeping Node.js safe, performant, and lightweight. We value enabling the ability to investigate and debug problems in development and production. Some key elements of this include:

- High throughput (speed)
- Fast startup
- Small binary size
- Small memory footprint
- High-quality debugging tools
- Robust diagnostic tools (profilers, etc.)

- Responsible security practices

#### **4 - Node.js maintainer experience**

We value the productivity and happiness of the Node.js maintainers. Some key elements of this include:

- Approachability of the codebase
- Good internal documentation and guides
- Low-friction policies and processes
- Good CI and tooling to make maintainers productive

#### **5 - Up to date technology and APIs**

We value providing developers with modern APIs and technologies following existing standards whenever possible.

Some key elements of this include:

- Participating in standards work and organizations
- Web API compatibility
- Supporting and exposing new technologies and standards through early adoption