

Why Atom?

There are a lot of text editors out there; why should you spend your time learning about and using Atom?

Editors like Sublime and TextMate offer convenience but only limited extensibility. On the other end of the spectrum, Emacs and Vim offer extreme flexibility, but they aren't very approachable and can only be customized with special-purpose scripting languages.

We think we can do better. Our goal is a zero-compromise combination of hackability and usability: an editor that will be welcoming to an elementary school student on their first day learning to code, but also a tool they won't outgrow as they develop into seasoned hackers.

As we've used Atom to build Atom, what began as an experiment has gradually matured into a tool we can't live without. On the surface, Atom is the modern desktop text editor you've come to expect. Pop the hood, however, and you'll discover a system begging to be hacked on.

The Nucleus of Atom

The web is not without its faults, but two decades of development has forged it into an incredibly malleable and powerful platform. So when we set out to write a text editor that we ourselves would want to extend, web technology was the obvious choice. But first, we had to free it from its chains.

The Native Web

Web browsers are great for browsing web pages, but writing code is a specialized activity that warrants dedicated tools. More importantly, the browser severely restricts access to the local system for security reasons, and for us, a text editor that couldn't write files or run local subprocesses was a non-starter.

For this reason, we didn't build Atom as a traditional web application. Instead, Atom is a specialized variant of Chromium designed to be a text editor rather than a web browser. Every Atom window is essentially a locally-rendered web page.

All the APIs available to a typical Node.js application are also available to the code running in each window's JavaScript context. This hybrid provides a really unique client-side development experience.

Since everything is local, you don't have to worry about asset pipelines, script concatenation, and asynchronous module definitions. If you want to load some code, just require it at the top of your file. Node's module system makes it easy to break the system down into lots of small, focused packages.

JavaScript, Meet C++

Interacting with native code is also really simple. For example, we wrote a wrapper around the Oniguruma regular expression engine for our TextMate grammar support. In a browser, that would have required adventures with NaCl or Esprima. Node integration made it easy.

In addition to the Node APIs, we also expose APIs for native dialogs, adding application and context menu items, manipulating the window dimensions, etc.

Web Tech: The Fun Parts

Another great thing about writing code for Atom is the guarantee that it's running on the newest version of Chromium. That means we can ignore issues like browser compatibility and polyfills. We can use all the web's shiny features of tomorrow, today.

For example, the layout of our workspace and panes is based on flexbox. It's an emerging standard and has gone through a lot of change since we started using it, but none of that mattered as long as it worked.

With the entire industry pushing web technology forward, we're confident that we're building Atom on fertile ground. Native UI technologies come and go, but the web is a standard that becomes more capable and ubiquitous with every passing year. We're excited to dig deeper into its toolbox.

An Open Source Text Editor

We see Atom as a perfect complement to GitHub's primary mission of building better software by working together. Atom is a long-term investment, and GitHub will continue to support its development with a dedicated team going forward. But we also know that we can't achieve our vision for Atom alone. As Emacs and Vim have demonstrated over the past three decades, if you want to build a thriving, long-lasting community around a text editor, it has to be open source.

The entire Atom editor is free and open source and is available under the <https://github.com/atom> organization.

Resource

<http://flight-manual.atom.io/getting-started/sections/why-atom/>