# Contributing to Vimium

## Reporting a bug

File the issue [here](https://github.com/philc/vimium/issues).

## Contributing code

You'd like to fix a bug or implement a feature? Great! Before getting started, understand Vimium's design

principles and the goals of the maintainers.

### Vimium design principles

When people first start using Vimium, it provides an incredibly powerful workflow improvement and it makes

them feel awesome. Surprisingly, Vimium is applicable to a huge, broad population of people, not just users of

Vim.

In addition to power, a secondary goal of Vimium is approachability: minimizing the barriers which prevent a

new user from feeling awesome. Many of Vimium's users haven't used Vim before -- about 1 in 5 Chrome Store

reviews say this -- and most people have strong web browsing habits forged from years of browsing. Given that,

it's a great experience when Vimium feels like a natural addition to Chrome which augments, but doesn't break,

the user's current browsing habits.

\*\*Principles:\*\*

1. \*\*Easy to understand\*\*. Even if you're not very familiar with Vim. The Vimium video shows you all you need

to know to start using Vimium and feel awesome.

2. \*\*Reliable\*\*. The core feature set works on most sites on the web.

3. \*\*Immediately useful\*\*. Vimium doesn't require any configuration or doc-reading before it's useful. Just

watch the video or hit `?`. You can transition into using Vimium piecemeal; you don't need to jump in

whole-hog from the start.

4. \*\*Feels native\*\*. Vimium doesn't drastically change the way Chrome looks or behaves.

5. \*\*Simple\*\*. The core feature set isn't overwhelming. This principle is particularly vulnerable as we add to

Vimium, so it requires our active effort to maintain this simplicity.

6. \*\*Code simplicity\*\*. Developers find the Vimium codebase relatively simple and easy to jump into. This

provides us an active dev community.

### Which pull requests get merged?

\*\*Goals of the maintainers\*\*

The maintainers of Vimium are @smblott-github and @philc. We have limited bandwidth, which influences which

PRs we can review and merge.

Our goals are generally to keep Vimium small, maintainable, and really nail the broad appeal use cases. This

is in contrast to adding and maintaining an increasing number of complex or niche features. We recommend those

live in forked repos rather than the mainline Vimium repo.

PRs we'll likely merge:

\* Reflect all of the Vimium design principles.

\* Are useful for lots of Vimium users.

\* Have simple implementations (straightforward code, few lines of code).

PRs we likely won't:

\* Violate one or more of our design principles.

\* Are niche.

\* Have complex implementations -- more code than they're worth.

Tips for preparing a PR:

\* If you want to check with us first before implementing something big, open an issue proposing the idea.

You'll get feedback from the maintainers as to whether it's something we'll likely merge.

\* Try to keep PRs around 50 LOC or less. Bigger PRs create inertia for review.

### Installing From Source

Vimium is written in Javascript. To install Vimium from source:

\*\*On Chrome/Chromium:\*\*

1. Navigate to `chrome://extensions`

1. Toggle into Developer Mode

1. Click on "Load Unpacked Extension..."

1. Select the Vimium directory you've cloned from Github.

\*\*On Firefox:\*\*

For 'local storage' to work while using the temporary addon, you need to add an 'application' section to the

manifest with an arbitrary ID that is unique for you, for example:

"applications": {

"gecko": {

"id": "vimium@example.net"

}

},

After that:

1. Open Firefox

1. Enter "about:debugging" in the URL bar

1. Click "Load Temporary Add-on"

1. Open the Vimium directory you've cloned from Github, and select any file inside.

### Running the tests

Our tests use [shoulda.js](https://github.com/philc/shoulda.js) and

[Puppeteer](https://github.com/puppeteer/puppeteer). To run the tests:

1. `git submodule update --init --recursive` to fetch shoulda.js.

1. Install [nodejs](https://nodejs.org/) if you don't have it already.

1. `npm install puppeteer` to install [Puppeteer](https://github.com/puppeteer/puppeteer).

1. `./make.js test` to build the code and run the tests.

### Coding Style

\* We follow the recommendations from the [Airbnb Javascript style guide](https://github.com/airbnb/javascript).

\* When writing comments, uppercase the first letter of your sentence, and put a period at the end.

\* We follow two major differences from this style guide:

\* Wrap lines at 110 characters instead of 100, for historical reasons.

\* We use double-quoted strings by default, for historical reasons.

\* We allow short, simple if statements to be used without braces, like so:

if (string.length == 0)

return;

...

\* We're currently using Javascript language features from ES2018 or earlier. If we desire to use something

introduced in a later version of Javascript, we need to remember to update the minimum Chrome and Firefox

versions required.