# Contributing

We can use help in a bunch of areas and any help is appreciated. Our [GitHub issues](https://github.com/romefrontend/rome/issues) serve as a place for any discussion, whether it's bug reports, questions, project direction etc. As the project grows this policy may change.

Our [Discord server](https://discord.gg/9WxHa5d) is open for help and more adhoc discussion. All activity on the Discord is still moderated and will be strictly enforced under the project's [Code of Conduct](./CODE\_OF\_CONDUCT.md).

## Getting Started

Getting started with developing Rome takes only three commands. You will only need Node v12 or above.

```bash

git clone https://github.com/romefrontend/rome

cd rome

./rome --help

```

## Developing on Windows

You need to use the backslash (`\`) to run any `rome` command on Windows instead of the slash (`/`); Windows uses backslashes for file paths.

For example, to run help:

```bash

.\rome --help

```

or you can directly use `rome` without any path referencing like below:

```bash

rome --help

```

No dependency installation step is required as we check in our `node\_modules` folder that contains only a copy of TypeScript and some definitions.

### User files

If files specific to your local development environment should be ignored, please add these files to a global git ignore file rather than to a git ignore file within Rome.

You can find more information on this process [here](https://help.github.com/en/github/using-git/ignoring-files#configuring-ignored-files-for-all-repositories-on-your-computer).

## Website

The [Rome website](https://romefrontend.dev/) is built with [Eleventy](https://www.11ty.dev/). To start a development server you can run the following commands:

```bash

cd website

npm install

npm start

```

## Checks

When working on Rome you will want to run the tests and linter to validate your changes. You can do both of these with a single command:

```bash

./rome ci

```

This is the main command we run when you submit a PR, so running it locally and making sure it passes will make it easier to review and merge your changes.

To automatically update test snapshots, apply formatting and autofixes, add the `--fix` flag.

```bash

./rome ci --fix

```

You can alternatively run more specific commands if you need to, but they shouldn't be necessary.

### Linting

To run just the linter use:

```bash

./rome check

```

And to automatically apply formatting and autofixes:

```bash

./rome check --apply

```

### Testing

If you would like to run only the test runner:

```bash

./rome test

```

And to update snapshots:

```bash

./rome test --update-snapshots

```

### Generated files

If you are adding a new lint rule, or modifying some core code, you might need to regenerate some files. We have generated files to avoid having to write a lot of boilerplate and automate common tasks.

```bash

./rome run scripts/generate-all-files

```

## Scripts

Here are some other scripts that you might find useful.

### `lint-create-rule`

This is used to generate new lint rules and boilperlate.

```bash

./rome run scripts/lint-create-rule [category]/[ruleName]

```

The `category` is one of the lint category folders defined in [`internal/compiler/lint/rules`](https://github.com/romefrontend/rome/tree/main/internal/compiler/lint/rules). Some of these represent specific languages, or general themes.

For example, to create a rule in the `js` category called `camelCase` run:

```bash

./rome run scripts/lint-create-rule js/camelCase

```

The created files will be listed in the console output. See those files for inline comments on what to insert. Use other lint rules as a reference.

### `ast-create-node`

This is used to generate new ast nodes and boilerplate.

```bash

./rome run scripts/ast-create-node [language] [nodeType] [category]

```

The `language` is one of the language folders defined in [`https://github.com/romefrontend/rome/tree/main/internal/ast/`]

The `category` is one of the category folders inside the `language` folders.

```bash

./rome run scripts/ast-create-node js JSArrayType typescript

```

The created files will be listed in the console output.