# Contributing

First of all, thank you for contributing. Any contribution is highly appreciated and welcome.

## Philosophy

There's a simple philosophy behind the Spaceship:

\* \*\*Show only what's needed.\*\* The prompt should not be overloaded, it shows only what user needs at the moment (current directory, git branch, etc).

\* \*\*Work out of the box.\*\* The prompt should work right after installation without any additional configuration. Install it and use it.

\* \*\*Be configurable.\*\* It works without configuration, but if a configuration is needed, it provides an easy-to-use interface for customization.

Please, keep this simple rules in mind while you're contributing to Spaceship.

## Setup

1. \*\*Fork\*\* this repo (click the \_fork\_ button)

2. \*\*Clone\*\* your fork to your working machine (via `git clone`)

3. \*\*Update submodules\*\* in your fork (via `git submodule update --init --recursive`)

4. \*\*Make your changes\*\*. Check our [API](./docs/API.md) for more information (we suggest you to check out a new branch for changes).

5. \*\*Test\*\* your code (via `npm test`)

6. \*\*Add and commit\*\* your contributions

7. \*\*Push\*\* your changes to your remote fork

8. \*\*Open a pull-request\*\* to our primary repo

## Sections

Spaceship supports most of the popular programming languages, runtimes, version managers, etc. If it doesn't support something that you need, feel free to open a pull request, but answer these questions for yourself before:

### Will it clutter the prompt?

Having too much in prompt looks ugly. your much space or be shown too often.

\* \*\*Good:\*\* `? v1.2.3`

\* \*\*Bad:\*\* `? spasheship#c3BhY2VzaGlw`

### Is it worth to be aware of it?

Is value changes quite often so it needs to be shown in prompt? Would it be useful for other users? Maybe there's a reason to execute a command instead of cluttering prompt.

\* \*\*Good:\*\* git status/branch, runtime version via version manager, etc

\* \*\*Bad:\*\* version of language-specific framework, settled projects versions, etc

### Will it slow down the prompt?

Every additional section will slow down the prompt a little bit. If your section performs any heavy checkings, find a way to make it faster or consider using an alias instead of creating a new section.

\* \*\*Good:\*\* check if command exists, check the value of environment variable

\* \*\*Bad:\*\* network requests, reading large files, etc

### What is the naming convention for options?

All options of prompt follow a specific pattern so that it is easy to remember: `SPACESHIP\_SECTION\_<OPTION>[\_PROPERTY]`. The rule is simple: when naming new properties, keep unique parts of the name to the end.

\* \*\*Good:\*\*

```

SPACESHIP\_GIT\_STATUS\_COLOR\_BEHIND

SPACESHIP\_GIT\_STATUS\_COLOR\_DIVERGED

```

\* \*\*Bad:\*\*

```

SPACESHIP\_GIT\_STATUS\_BEHIND\_COLOR

SPACESHIP\_GIT\_STATUS\_DIVERGED\_COLOR

```

Here, `GIT\_STATUS` is \*section\*, `COLOR` is \*option\* and `BEHIND` or `DIVERGED` is \*property\*.

### Documentation

When updating documentation for your section, make sure the markdown document is being properly rendered by Github. Specifically, the following common pitfalls have already been discovered:

\* Empty inline code block ` ` will only be rendered if you put at least one non-breaking whitespace "&nbsp;" inside, like so: `` ` ` `` ? ` `

\* Leading and trailing whitespaces in inline code blocks will be stripped, to indicate that a whitespace is present, use the middot symbol `·`, like so: `` `?·` `` ? `?·`

\*\*Thanks for reading this contribution guide! Happy hacking!\*\*