Starship

Configuration   Languages ▼   GitHub ⬏

Frequently Asked Questions

What is the configuration used in the demo GIF?

- **Terminal Emulator**: iTerm2⬏
  - **Theme**: Minimal
  - **Color Scheme**: Snazzy⬏
  - **Font**: FiraCode Nerd Font⬏
- **Shell**: Fish Shell⬏
  - **Configuration**: matchai's Dotfiles⬏
  - **Prompt**: Starship⬏

How do I get command completion as shown in the demo GIF?

Completion support, or autocomplete, is provided by your shell of choice. In the case of the demo, the demo was done with Fish Shell⬏ , which provides completions by default. If you use Z Shell (zsh), I'd suggest taking a look at zsh-autosuggestions⬏ .

Do top level `format` and `<module>.disabled` do the same thing?

🚀 **Starship**

Configuration   Languages ▼   GitHub ⬈

`<module>.disabled` is the preferred way to do so for these reasons:

- Disabling modules is more explicit than omitting them from the top level `format`
- Newly created modules will be added to the prompt as Starship is updated

The docs say Starship is cross-shell. Why isn't my preferred shell supported?

The way Starship is built, it should be possible to add support for virtually any shell. The starship binary is stateless and shell agnostic, so as long as your shell supports prompt customization and shell expansion, Starship can be used.

Here's a small example getting Starship working with bash:

```sh
# Get the status code from the last command ex
STATUS=$?

# Get the number of jobs running.
NUM_JOBS=$(jobs -p | wc -l)

# Set the prompt to the output of `starship pr
PS1="$(starship prompt --status=$STATUS --jobs
```

The Bash implementation⬈ built into Starship is slightly more complex to allow for advanced features like the Command Duration module⬈ and to ensure

Starship

Configuration   Languages ▼   GitHub ⬈

For a list of all flags accepted by `starship prompt`, use the following command:

```sh
starship prompt --help
```

The prompt will use as much context as is provided, but no flags are "required".

How do I run Starship on Linux distributions with older versions of glibc?

If you get an error like "*version 'GLIBC_2.18' not found (required by starship)*" when using the prebuilt binary (for example, on CentOS 6 or 7), you can use a binary compiled with `musl` instead of `glibc`:

```sh
curl -sS https://starship.rs/install.sh | sh
```

Why do I see `Executing command "..." timed out.` warnings?

Starship executes different commands to get information to display in the prompt, for example the version of a program or the current git status. To make sure starship doesn't hang while trying to execute these commands we set a time limit, if a command takes longer than this limit starship will stop the execution of

![Starship logo] **Starship**

using the `command_timeout` `key` so if you want you can increase the time limit. You can also follow the debugging steps below to see which command is being slow and see if you can optimise it. Finally you can set the `STARSHIP_LOG` env var to `error` to hide these warnings.

I see symbols I don't understand or expect, what do they mean?

If you see symbols that you don't recognise you can use `starship explain` to explain the currently showing modules.

Starship is doing something unexpected, how can I debug it?

You can enable the debug logs by using the `STARSHIP_LOG` env var. These logs can be very verbose so it is often useful to use the `module` command if you are trying to debug a particular module, for example, if you are trying to debug the `rust` module you could run the following command to get the trace logs and output from the module.

```sh
env STARSHIP_LOG=trace starship module rust
```

If starship is being slow you can try using the `timings` command to see if there is a particular module or command that to blame.

Starship

Configuration   Languages ▾   GitHub ⧉

This will output the trace log and a breakdown of all modules that either took more than 1ms to execute or produced some output.

Finally if you find a bug you can use the `bug-report` command to create a GitHub issue.

```sh
starship bug-report
```

Why don't I see a glyph symbol in my prompt?

The most common cause of this is system misconfiguration. Some Linux distros in particular do not come with font support out-of-the-box. You need to ensure that:

- Your locale is set to a UTF-8 value, like `de_DE.UTF-8` or `ja_JP.UTF-8` . If `LC_ALL` is not a UTF-8 value, you will need to change it⧉ .
- You have an emoji font installed. Most systems come with an emoji font by default, but some (notably Arch Linux) do not. You can usually install one through your system's package manager--noto emoji⧉ is a popular choice.
- You are using a Nerd Font⧉ .

To test your system, run the following commands in a terminal:

Starship

Configuration   Languages ▾   GitHub ⎋

The first line should produce a snake emoji ⎋ , while the second should produce a powerline branch symbol (e0a0)⎋ .

If either symbol fails to display correctly, your system is still misconfigured. Unfortunately, getting font configuration correct is sometimes difficult. Users on the Discord may be able to help. If both symbols display correctly, but you still don't see them in starship, file a bug report! ⎋

How do I uninstall Starship?

Starship is just as easy to uninstall as it is to install in the first place.

1. Remove any lines in your shell config (e.g. `~/.bashrc`) used to initialize Starship.
2. Delete the Starship binary.

If Starship was installed using a package manager, please refer to their docs for uninstallation instructions.

If Starship was installed using the install script, the following command will delete the binary:

```sh
# Locate and delete the starship binary
sh -c 'rm "$(command -v 'starship')"'
```

How do I install Starship without `sudo` ?

Starship

Home

Guide

Advanced Installation

Configuration

Advanced Configuration

Frequently Asked Questions

installation directory is not writable by the current user. The default installation diretory is the value of the `$BIN_DIR` environment variable or `/usr/local/bin` if `$BIN_DIR` is not set. If you instead set the installation directory to one that is writable by your user, you should be able to install starship without `sudo`. For example, `curl -sS https://starship.rs/install.sh | sh -s -- -b ~/.local/bin` uses the `-b` command line option of the install script to set the installation directory to `~/.local/bin`.

For a non-interactive installation of Starship, don't forget to add the `-y` option to skip the confirmation. Check the source of the installation script for a list of all supported installation options.

When using a package manager, see the documentation for your package manager about installing with or without `sudo`.

Edit this page on GitHub ⧉

← Advanced Configuration   Presets →