Advanced Configuration

While Starship is a versatile shell, sometimes you need to do more than edit `starship.toml` to get it to do certain things. This page details some of the more advanced configuration techniques used in starship.

> **WARNING**
>
> The configurations in this section are subject to change in future releases of Starship.

TransientPrompt in PowerShell

It is possible to replace the previous-printed prompt with a custom string. This is useful in cases where all the prompt information is not always needed. To enable this, run `Enable-TransientPrompt` in the shell session. To make it permanent, put this statement in your `$PROFILE`. Transience can be disabled on-the-fly with `Disable-TransientPrompt`.

By default, the left side of input gets replaced with `>`. To customize this, define a new function called `Invoke-Starship-TransientFunction`. For example, to display Starship's `character` module here, you would do

```
function Invoke-Starship-TransientFunction {
  &starship module character
}
```

```
Enable-TransientPrompt
```

TransientPrompt and TransientRightPrompt in Cmd

Clink allows you to replace the previous-printed prompt with custom strings. This is useful in cases where all the prompt information is not always needed. To enable this, run `clink set prompt.transient <value>` where <value> can be one of:

- `always` : always replace the previous prompt
- `same_dir` : replace the previous prompt only if the working directory is same
- `off` : do not replace the prompt (i.e. turn off transience)

You need to do this only once. Make the following changes to your `starship.lua` to customize what gets displayed on the left and on the right:

- By default, the left side of input gets replaced with `> `. To customize this, define a new function called `starship_transient_prompt_func`. This function receives the current prompt as a string that you can utilize. For example, to display Starship's `character` module here, you would do

```
function starship_transient_prompt_func(prompt
  return io.popen("starship module character"
    .." --keymap="..rl.getvariable('keymap')
  ):read("*a")
end
```

- By default, the right side of input is empty. To customize this, define a new function called `starship_transient_rprompt_func`. This function receives the current prompt as a string that you can utilize. For example, to display the time at which the last command was started here, you would do

```
function starship_transient_rprompt_func(prom
  return io.popen("starship module time"):rea
end
load(io.popen('starship init cmd'):read("*a")
```

TransientPrompt and TransientRightPrompt in Fish

It is possible to replace the previous-printed prompt with a custom string. This is useful in cases where all the prompt information is not always needed. To enable this, run `enable_transience` in the shell session. To make it permanent, put this statement in your `~/.config/fish/config.fish`. Transience can be disabled on-the-fly with `disable_transience`.

Note that in case of Fish, the transient prompt is only printed if the commandline is non-empty, and syntactically correct.

- By default, the left side of input gets replaced with a bold-green `❯`. To customize this, define a new function called `starship_transient_prompt_func`. For example, to display Starship's `character` module here, you would do

```
end
starship init fish | source
enable_transience
```

- By default, the right side of input is empty. To customize this, define a new function called `starship_transient_rprompt_func`. For example, to display the time at which the last command was started here, you would do

```
function starship_transient_rprompt_func
  starship module time
end
starship init fish | source
enable_transience
```

Custom pre-prompt and pre-execution Commands in Cmd

Clink provides extremely flexible APIs to run pre-prompt and pre-exec commands in Cmd shell. It is fairly simple to use with Starship. Make the following changes to your `starship.lua` file as per your requirements:

- To run a custom function right before the prompt is drawn, define a new function called `starship_preprompt_user_func`. This function receives the current prompt as a string that you can utilize. For example, to draw a rocket before the prompt, you would do

```
end

load(io.popen('starship init cmd'):read("*a")
```

- To run a custom function right before a command is executed, define a new function called `starship_precmd_user_func`. This function receives the current commandline as a string that you can utilize. For example, to print the command that's about to be executed, you would do

```
function starship_precmd_user_func(line)
  print("Executing: "..line)
end

load(io.popen('starship init cmd'):read("*a")
```

Custom pre-prompt and pre-execution Commands in Bash

Bash does not have a formal preexec/precmd framework like most other shells. Because of this, it is difficult to provide fully customizable hooks in `bash`. However, Starship does give you limited ability to insert your own functions into the prompt-rendering procedure:

- To run a custom function right before the prompt is drawn, define a new function and then assign its name to `starship_precmd_user_func`. For example, to draw a rocket before the prompt, you would do

```
}
starship_precmd_user_func="blastoff"
```

- To run a custom function right before a command runs, you can use the `DEBUG` trap mechanism. However, you **must** trap the DEBUG signal *before* initializing Starship! Starship can preserve the value of the DEBUG trap, but if the trap is overwritten after starship starts up, some functionality will break.

sh

```
function blastoff(){
    echo "🚀"
}
trap blastoff DEBUG     # Trap DEBUG *before*
set -o functrace
eval $(starship init bash)
set +o functrace
```

Custom pre-prompt and pre-execution Commands in PowerShell

PowerShell does not have a formal preexec/precmd framework like most other shells. Because of this, it is difficult to provide fully customizable hooks in `powershell`. However, Starship does give you limited ability to insert your own functions into the prompt-rendering procedure:

Create a function named `Invoke-Starship-PreCommand`

Home

Guide

Advanced Installation

Configuration

Advanced Configuration

TransientPrompt in PowerShell

TransientPrompt and TransientRightPrompt in Cmd

TransientPrompt and TransientRightPrompt in Fish

Custom pre-prompt and pre-execution Commands in Cmd

Custom pre-prompt and pre-execution Commands in Bash

Custom pre-prompt and pre-execution Commands in PowerShell

Change Window Title

Enable Right Prompt

Continuation Prompt

Style Strings

Frequently Asked Questions

Presets

```
}
```

Change Window Title

Some shell prompts will automatically change the window title for you (e.g. to reflect your working directory). Fish even does it by default. Starship does not do this, but it's fairly straightforward to add this functionality to `bash`, `zsh`, `cmd` or `powershell`.

First, define a window title change function (identical in bash and zsh):

```sh
function set_win_title(){
    echo -ne "\033]0; YOUR_WINDOW_TITLE_HERE \
}
```

You can use variables to customize this title (`$USER`, `$HOSTNAME`, and `$PWD` are popular choices).

In `bash`, set this function to be the precmd starship function:

```sh
starship_precmd_user_func="set_win_title"
```

In `zsh`, add this to the `precmd_functions` array:

```sh
precmd_functions+=(set_win_title)
```

If you like the result, add these lines to your shell configuration file (`~/.bashrc` or `~/.zshrc`) to make

For example, if you want to display your current directory in your terminal tab title, add the following snippet to your `~/.bashrc` or `~/.zshrc` :

```sh
function set_win_title(){
    echo -ne "\033]0; $(basename "$PWD") \007"
}
starship_precmd_user_func="set_win_title"
```

For Cmd, you can change the window title using the `starship_preprompt_user_func` function.

```
function starship_preprompt_user_func(prompt)
  console.settitle(os.getenv('USERNAME').."@"
end

load(io.popen('starship init cmd'):read("*a")
```

You can also set a similar output with PowerShell by creating a function named `Invoke-Starship-PreCommand` .

```
# edit $PROFILE
function Invoke-Starship-PreCommand {
  $host.ui.RawUI.WindowTitle = "$env:USERNAME@
}

Invoke-Expression (&starship init powershell)
```

Enable Right Prompt

Some shells support a right prompt which renders on the same line as the input. Starship can set the content

Home
Guide
Advanced Installation
Configuration
Advanced Configuration
TransientPrompt in PowerShell
TransientPrompt and TransientRightPrompt in Cmd
TransientPrompt and TransientRightPrompt in Fish
Custom pre-prompt and pre-execution Commands in Cmd
Custom pre-prompt and pre-execution Commands in Bash
Custom pre-prompt and pre-execution Commands in PowerShell
Change Window Title
Enable Right Prompt
Continuation Prompt
Style Strings
Frequently Asked Questions
Presets

supported in `right_format`. The `$all` variable will only contain modules not explicitly used in either `format` or `right_format`.

Note: The right prompt is a single line following the input location. To right align modules above the input line in a multi-line prompt, see the `fill` module.

`right_format` is currently supported for the following shells: elvish, fish, zsh, xonsh, cmd, nushell.

Example

```
# ~/.config/starship.toml

# A minimal left prompt
format = """$character"""

# move the rest of the prompt to the right
right_format = """$all"""
```

Produces a prompt like the following:

```
▶                                                    starship
```

Continuation Prompt

Some shells support a continuation prompt along with the normal prompt. This prompt is rendered instead of the normal prompt when the user has entered an incomplete statement (such as a single left parenthesis or quote).

```
'[·](bright-black) ' .
```

Note: `continuation_prompt` should be set to a literal string without any variables.

Note: Continuation prompts are only available in the following shells:

- `bash`
- `zsh`
- `PowerShell`

Example

```
# ~/.config/starship.toml

# A continuation prompt that displays two fil
continuation_prompt = '▶▶ '
```

Style Strings

Style strings are a list of words, separated by whitespace. The words are not case sensitive (i.e. `bold` and `BoLd` are considered the same string). Each word can be one of the following:

- `bold`
- `italic`
- `underline`
- `dimmed`
- `inverted`
- `blink`
- `hidden`

- `fg:<color>`
- `<color>`
- `none`

where `<color>` is a color specifier (discussed below). `fg:<color>` and `<color>` currently do the same thing, though this may change in the future. `inverted` swaps the background and foreground colors. The order of words in the string does not matter.

The `none` token overrides all other tokens in a string if it is not part of a `bg:` specifier, so that e.g. `fg:red none fg:blue` will still create a string with no styling. `bg:none` sets the background to the default color so `fg:red bg:none` is equivalent to `red` or `fg:red` and `bg:green fg:red bg:none` is also equivalent to `fg:red` or `red`. It may become an error to use `none` in conjunction with other tokens in the future.

A color specifier can be one of the following:

- One of the standard terminal colors: `black`, `red`, `green`, `blue`, `yellow`, `purple`, `cyan`, `white`. You can optionally prefix these with `bright-` to get the bright version (e.g. `bright-white`).
- A `#` followed by a six-digit hexadecimal number. This specifies an RGB color hex code.
- A number between 0-255. This specifies an 8-bit ANSI Color Code.

If multiple colors are specified for foreground/background, the last one in the string will take priority.

Home
Guide
Advanced Installation
Configuration
Advanced Configuration
TransientPrompt in PowerShell
TransientPrompt and TransientRightPrompt in Cmd
TransientPrompt and TransientRightPrompt in Fish
Custom pre-prompt and pre-execution Commands in Cmd
Custom pre-prompt and pre-execution Commands in Bash
Custom pre-prompt and pre-execution Commands in PowerShell
Change Window Title
Enable Right Prompt
Continuation Prompt
Style Strings
Frequently Asked Questions
Presets

quirks exist:

- Many terminals disable support for `blink` by default
- `hidden` is not supported on iTerm.
- `strikethrough` is not supported by the default macOS Terminal.app

Edit this page on GitHub

← Configuration Frequently Asked Questions →

Home

Guide

Advanced Installation

Chocolatey

termux

Funtoo Linux

Nix

Configuration

Advanced Configuration

Frequently Asked Questions

Presets

Advanced Installation

To install starship, you need to do two things:

1. Get the **starship** binary onto your computer
2. Tell your shell to use the starship binary as its prompt by modifying its init scripts

For most users, the instructions on the main page will work great. However, for some more specialized platforms, different instructions are needed.

There are so many platforms out there that they didn't fit into the main README.md file, so here are some installation instructions for other platforms from the community. Is yours not here? Please do add it here if you figure it out!

Chocolatey

Prerequisites

Head over to the Chocolatey installation page and follow the instructions to install Chocolatey.

Installation

```
choco install starship
```

termux

Prerequisites

```sh
pkg install getconf
```

Installation

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

Funtoo Linux

Installation

On Funtoo Linux, starship can be installed from core-kit via Portage:

```sh
emerge app-shells/starship
```

Nix

Getting the Binary

Imperatively

```sh
nix-env -iA nixos.starship
```

Declarative, single user, via home-manager

Enable the `programs.starship` module in your

```
{
  programs.starship = {
    enable = true;
    # Configuration written to ~/.config/star
    settings = {
      # add_newline = false;

      # character = {
      #   success_symbol = "[➜](bold green)";
      #   error_symbol = "[➜](bold red)";
      # };

      # package.disabled = true;
    };
  };
}
```

then run

```sh
home-manager switch
```

Declarative, system-wide, with NixOS

Add `pkgs.starship` to `environment.systemPackages` in your `configuration.nix`, then run

```sh
sudo nixos-rebuild switch
```

Edit this page on GitHub

← Guide Configuration →

Configuration

To get started configuring starship, create the following file: `~/.config/starship.toml` .

```sh
mkdir -p ~/.config && touch ~/.config/starshi
```

All configuration for starship is done in this TOML file:

```
# Get editor completions based on the config s
"$schema" = 'https://starship.rs/config-schema

# Inserts a blank line between shell prompts
add_newline = true

# Replace the '❯' symbol in the prompt with '·
[character] # The name of the module we are c
success_symbol = '[➜](bold green)' # The 'succ

# Disable the package module, hiding it from
[package]
disabled = true
```

Config File Location

You can change default configuration file location with `STARSHIP_CONFIG` environment variable:

```sh
export STARSHIP_CONFIG=~/example/non/default/
```

```
$ENV:STARSHIP_CONFIG = "$HOME\example\non\def
```

Or for Cmd (Windows) would be adding this line to your `starship.lua`:

```
os.setenv('STARSHIP_CONFIG', 'C:\\Users\\user
```

Logging

By default starship logs warnings and errors into a file named `~/.cache/starship/session_${STARSHIP_SESSION_KEY}.log`, where the session key is corresponding to an instance of your terminal. This, however can be changed using the `STARSHIP_CACHE` environment variable:

sh

```
export STARSHIP_CACHE=~/.starship/cache
```

Equivalently in PowerShell (Windows) would be adding this line to your `$PROFILE`:

```
$ENV:STARSHIP_CACHE = "$HOME\AppData\Local\Te
```

Or for Cmd (Windows) would be adding this line to your `starship.lua`:

```
os.setenv('STARSHIP_CACHE', 'C:\\Users\\user
```

Terminology

Module: A component in the prompt giving information based on contextual information from your OS. For example, the "nodejs" module shows the version of Node.js that is currently installed on your computer, if your current directory is a Node.js project.

Variable: Smaller sub-components that contain information provided by the module. For example, the "version" variable in the "nodejs" module contains the current version of Node.js.

By convention, most modules have a prefix of default terminal color (e.g. `via` in "nodejs") and an empty space as a suffix.

Strings

In TOML syntax, text values are declared with `'`, `"`, `'''`, or `"""`.

The following Starship syntax symbols have special usage in a format string and must be escaped to display as that character: `$ [ ] ( )`.

Symbol	Type	Notes
`'`	literal string	less escaping
`"`	string	more escaping
`'''`	multi-line literal string	less escaping

"""	multi-line string	more escaping, newlines in declarations can be ignored

For example:

```
# literal string
format = '☺\☻ '

# regular string
format = "☺\\☻ "

# escaping Starship symbols
format = '\[\$\] '
```

When using line breaks, multi-line declarations can be used. For example, if you want to print a `$` symbol on a new line, the following values for `format` are equivalent:

```
# with literal string
format = '''

\$'''

# with multiline basic string
format = """

\\$"""

# with basic string
format = "\n\\$"
```

In multiline basic strings, newlines can be used for formatting without being present in the value by

```
format = """
line1\
line1\
line1
line2\
line2\
line2
"""
```

Format Strings

Format strings are the format that a module prints all its variables with. Most modules have an entry called `format` that configures the display format of the module. You can use texts, variables and text groups in a format string.

Variable

A variable contains a `$` symbol followed by the name of the variable. The name of a variable can only contain letters, numbers and `_` .

For example:

- `'$version'` is a format string with a variable named `version` .
- `'$git_branch$git_commit'` is a format string with two variables named `git_branch` and `git_commit` .
- `'$git_branch $git_commit'` has the two variables separated with a space.

Text Group

The first part, which is enclosed in a `[]`, is a format string. You can add texts, variables, or even nested text groups in it.

In the second part, which is enclosed in a `()`, is a style string. This can be used to style the first part.

For example:

- `'[on](red bold)'` will print a string `on` with bold text colored red.
- `'[⌘ $version](bold green)'` will print a symbol `⌘` followed by the content of variable `version`, with bold text colored green.
- `'[a [b](red) c](green)'` will print `a b c` with `b` red, and `a` and `c` green.

Style Strings

Most modules in starship allow you to configure their display styles. This is done with an entry (usually called `style`) which is a string specifying the configuration. Here are some examples of style strings along with what they do. For details on the full syntax, consult the advanced config guide.

- `'fg:green bg:blue'` sets green text on a blue background
- `'bg:blue fg:bright-green'` sets bright green text on a blue background
- `'bold fg:27'` sets bold text with ANSI color 27
- `'underline bg:#bf5700'` sets underlined text on a burnt orange background
- `'bold italic fg:purple'` sets bold italic purple text

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

Note that what styling looks like will be controlled by your terminal emulator. For example, some terminal emulators will brighten the colors instead of bolding text, and some color themes use the same values for the normal and bright colors. Also, to get italic text, your terminal must support italics.

Conditional Format Strings

A conditional format string wrapped in `(` and `)` will not render if all variables inside are empty.

For example:

- `'(@$region)'` will show nothing if the variable `region` is `None` or empty string, otherwise `@` followed by the value of region.
- `'(some text)'` will always show nothing since there are no variables wrapped in the braces.
- When `$combined` is a shortcut for `\[$a$b\]`, `'($combined)'` will show nothing only if `$a` and `$b` are both `None`. This works the same as `'(\[$a$b\] )'`.

Negative matching

Many modules have `detect_extensions`, `detect_files`, and `detect_folders` variables. These take lists of strings to match or not match. "Negative" options, those which should not be matched, are indicated with a leading '!' character. The presence of *any* negative indicator in the directory will result in the module not being matched.

Extensions are matched against both the characters

`foo.bar.tar.gz` will be matched against `bar.tar.gz` and `gz` in the `detect_extensions` variable. Files whose name begins with a dot are not considered to have extensions at all.

To see how this works in practice, you could match TypeScript but not MPEG Transport Stream files thus:

```
detect_extensions = ['ts', '!video.ts', '!aud
```

Prompt

This is the list of prompt-wide configuration options.

Options

Option	Default	Description
`format`	link	Configure the format of the prompt.
`right_format`	`''`	See Enable Right Prompt
`scan_timeout`	`30`	Timeout for starship to scan files (in milliseconds).
`command_timeout`	`500`	Timeout for commands executed by starship (in milliseconds).

`add_newline`	`true`	Inserts blank line between shell prompts.
`palette`	`''`	Sets which color palette from `palettes` to use.
`palettes`	`{}`	Collection of color palettes that assign colors to user-defined names. Note that color palettes cannot reference their own color definitions.
`follow_symlinks`	`true`	Follows symlinks to check if they're directories; used in modules such as git.

> **TIP**
>
> If you have symlinks to networked filesystems, consider setting `follow_symlinks` to `false`.

Example

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

```
# Use custom format
format = '''
[┌──────────────────────>](bold green)
[│](bold green)$directory$rust$package
[└─>](bold green) '''

# Wait 10 milliseconds for starship to check
scan_timeout = 10

# Disable the blank line at the start of the
add_newline = false

# Set 'foo' as custom color palette
palette = 'foo'

# Define custom colors
[palettes.foo]
# Overwrite existing color
blue = '21'
# Define new color
mustard = '#af8700'
```

Default Prompt Format

The default `format` is used to define the format of the prompt, if empty or no `format` is provided. The default is as shown:

```
format = '$all'

# Which is equivalent to
format = """
$username\
$hostname\
$localip\
$shlvl\
$singularity\
```

Home

Guide

Advanced Installation

Configuration

Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

```
$vcsh\
$fossil_branch\
$fossil_metrics\
$git_branch\
$git_commit\
$git_state\
$git_metrics\
$git_status\
$hg_branch\
$pijul_channel\
$docker_context\
$package\
$c\
$cmake\
$cobol\
$daml\
$dart\
$deno\
$dotnet\
$elixir\
$elm\
$erlang\
$fennel\
$golang\
$guix_shell\
$haskell\
$haxe\
$helm\
$java\
$julia\
$kotlin\
$gradle\
$lua\
$nim\
$nodejs\
$ocaml\
$opa\
$perl\
$php\
$pulumi\
```

Home

Guide

Advanced Installation

Configuration

Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

```
$rlang\
$red\
$ruby\
$rust\
$scala\
$solidity\
$swift\
$terraform\
$typst\
$vlang\
$vagrant\
$zig\
$buf\
$nix_shell\
$conda\
$meson\
$spack\
$memory_usage\
$aws\
$gcloud\
$openstack\
$azure\
$direnv\
$env_var\
$crystal\
$custom\
$sudo\
$cmd_duration\
$line_break\
$jobs\
$battery\
$time\
$status\
$os\
$container\
$shell\
$character"""
```

If you just want to extend the default format, you can

```
# Move the directory to the second line
format = '$all$directory$character'
```

AWS

The `aws` module shows the current AWS region and profile and an expiration timer when using temporary credentials. The output of the module uses the `AWS_REGION`, `AWS_DEFAULT_REGION`, and `AWS_PROFILE` env vars and the `~/.aws/config` and `~/.aws/credentials` files as required.

The module will display a profile only if its credentials are present in `~/.aws/credentials` or if a `credential_process`, `sso_start_url`, or `sso_session` are defined in `~/.aws/config`. Alternatively, having any of the `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, or `AWS_SESSION_TOKEN` env vars defined will also suffice. If the option `force_display` is set to `true`, all available information will be displayed even if no credentials per the conditions above are detected.

When using aws-vault the profile is read from the `AWS_VAULT` env var and the credentials expiration date is read from the `AWS_SESSION_EXPIRATION` env var.

When using awsu the profile is read from the `AWSU_PROFILE` env var.

When using AWSume the profile is read from the `AWSUME_PROFILE` env var and the credentials

When using saml2aws the expiration information obtained from `~/.aws/credentials` falls back to the `x_security_token_expires` key.

When using aws-sso-cli the profile is read from the `AWS_SSO_PROFILE` env var.

Options

Option	Default	
`format`	`'on [$symbol($profile )(\($region\))(\[$duration\])]($style)'`	Th mo
`symbol`	`'☁️ '`	Th be cu
`region_aliases`	`{}`	Ta ali ad na
`profile_aliases`	`{}`	Ta ali ad na
`style`	`'bold yellow'`	Th mo
`expiration_symbol`	`'X'`	Th wh cr ex

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`disabled`	`false`	Dis mc
`force_display`	`false`	If ev c c s s ha

Variables

Variable	Example	Description
region	`ap-northeast-1`	The current AWS region
profile	`astronauts`	The current AWS profile
duration	`2h27m20s`	The temporary credentials validity duration
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Examples

Display everything

```
[aws]
format = 'on [$symbol($profile )(\($region\)
style = 'bold blue'
symbol = '🅰 '
[aws.region_aliases]
ap-southeast-2 = 'au'
us-east-1 = 'va'
[aws.profile_aliases]
CompanyGroupFrobozzOnCallAccess = 'Frobozz'
```

Display region

```
# ~/.config/starship.toml

[aws]
format = 'on [$symbol$region]($style) '
style = 'bold blue'
symbol = '🅰 '
[aws.region_aliases]
ap-southeast-2 = 'au'
us-east-1 = 'va'
```

Display profile

```
# ~/.config/starship.toml

[aws]
format = 'on [$symbol$profile]($style) '
style = 'bold blue'
symbol = '🅰 '
[aws.profile_aliases]
Enterprise_Naming_Scheme-voidstars = 'void**'
```

Azure

The `azure` module shows the current Azure Subscription. This is based on showing the name of the default subscription or the username, as defined in the `~/.azure/azureProfile.json` file.

Options

Variable	Default
`format`	`'on [$symbol($subscripti ](($style) '`
`symbol`	`'  '`
`style`	`'blue bold'`
`disabled`	`true`
`subscription_aliases`	`{}`

Examples

Display Subscription Name

```
# ~/.config/starship.toml

[azure]
disabled = false
format = 'on [$symbol($subscription)]($style)
symbol = ' '
style = 'blue bold'
```

Display Username

```
# ~/.config/starship.toml

[azure]
disabled = false
format = "on [$symbol($username)]($style) "
symbol = " "
style = "blue bold"
```

Display Subscription Name Alias

```
# ~/.config/starship.toml

[azure.subscription_aliases]
very-long-subscription-name = 'vlsn'
```

Battery

The `battery` module shows how charged the device's

Options

Option	Default
`full_symbol`	`'  '`
`charging_symbol`	`'  '`
`discharging_symbol`	`'  '`
`unknown_symbol`	`'  '`
`empty_symbol`	`'  '`
`format`	`'[$symbol$percentage](($style) '`
`display`	link

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`disabled`	`false`

Example

```
# ~/.config/starship.toml

[battery]
full_symbol = '🔋 '
charging_symbol = '⚡ '
discharging_symbol = '💀 '
```

Battery Display

The `display` configuration option is used to define when the battery indicator should be shown (threshold), which symbol would be used (symbol), and what it would like (style). If no `display` is provided. The default is as shown:

```
[[battery.display]]
threshold = 10
style = 'bold red'
```

The default value for the `charging_symbol` and `discharging_symbol` option is respectively the value of `battery` 's `charging_symbol` and `discharging_symbol` option.

Options

The `display` option is an array of the following table.

Option	Default	Descript
`threshold`	`10`	The upper bo the display op
`style`	`'red bold'`	The style use display optio use.
`charging_symbol`		Optional sym displayed if d option is in us defaults to ba `charging_sy` option.
`discharging_symbol`		Optional sym displayed if d option is in us defaults to ba `discharging l` option.

Example

```
[[battery.display]] # 'bold red' style and di
threshold = 10
style = 'bold red'

[[battery.display]] # 'bold yellow' style and
threshold = 30
style = 'bold yellow'
discharging_symbol = '💦'

# when capacity is over 30%, the battery indi
```

Buf

The `buf` module shows the currently installed version of Buf. By default, the module is shown if all of the following conditions are met:

- The `buf` CLI is installed.
- The current directory contains a `buf.yaml`, `buf.gen.yaml`, or `buf.work.yaml` configuration file.

Options

Option	Default	De
`format`	`'with [$symbol($version )]($style)'`	Th for bu mo
`version_format`	`'v${raw}'`	Th for
`symbol`	`'🐃 '`	Th us dis the of
`detect_extensions`	`[]`	Wh ext sh tri mo
`detect_files`	`['buf.yaml', 'buf.gen.yaml', 'buf.work.yaml']`	W file sh tri

`detect_folders`	`[]`	Wh fol sh tri mo
`style`	`'bold blue'`	Th for mo
`disabled`	`false`	Dis the e mo

Variables

Variable	Example	Description
`version`	`v1.0.0`	The version of `buf`
`symbol`		Mirrors the value of option `symbol`
`style *`		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[buf]
symbol = '🦬 '
```

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

Bun

The `bun` module shows the currently installed version of the bun JavaScript runtime. By default the module will be shown if any of the following conditions are met:

- The current directory contains a `bun.lockb` file
- The current directory contains a `bunfig.toml` file

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for Av va ra ma mi pa
`symbol`	`'🍞 '`	A f str rep the of
`detect_extensions`	`[]`	Wh ext sh tri m

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`detect_files`	`['bun.lockb', 'bunfig.toml']`	Vvi file sh tri mc
`detect_folders`	`[]`	Wl fol sh tri mc
`style`	`'bold red'`	Th the
`disabled`	`false`	Dis bu mc

Variables

Variable	Example	Description
version	`v0.1.4`	The version of `bun`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[bun]
```

C

The `c` module shows some information about your C compiler. By default the module will be shown if the current directory contains a `.c` or `.h` file.

Options

Option	Default
`format`	`'via [$symbol($version(-$nam ) )]($style)'`
`version_format`	`'v${raw}'`
`symbol`	`'C '`
`detect_extensions`	`['c', 'h']`
`detect_files`	`[]`

`detect_folders`	`[]`
`commands`	`[ [ 'cc', '--version' ], [ 'gcc', '--version' ], [ 'clang', '--version' ] ]`
`style`	`'bold 149'`
`disabled`	`false`

Variables

Variable	Example	Description
name	clang	The name of the compiler
version	13.0.0	The version of the compiler
symbol		Mirrors the value of option `symbol`
style		Mirrors the value of option `style`

NB that `version` is not in the default format.

Commands

Each command is represented as a list of the executable name, followed by its arguments, usually something like `['mycc', '--version']`. Starship will try executing each command until it gets a result on STDOUT.

If a C compiler is not supported by this module, you can request it by raising an issue on GitHub.

Example

```
# ~/.config/starship.toml

[c]
format = 'via [$name $version]($style)'
```

Character

The `character` module shows a character (usually an arrow) beside where the text is entered in your terminal.

The character will tell you whether the last command was successful or not. It can do this in two ways:

- changing color (`red` / `green`)
- changing shape (`❯` / `✖`)

By default it only changes color. If you also want to change its shape take a look at this example.

> **WARNING**

`vimcmd_replace_symbol` , and `vimcmd_visual_symbol` are only supported in fish due to upstream issues with mode detection in zsh .

Options

Option	Default	I
`format`	`'$symbol '`	T st b te
`success_symbol`	`'[❯](bold green)'`	T st b te th co su
`error_symbol`	`'[❯](bold red)'`	T st b te th co fa
`vimcmd_symbol`	`'[❮](bold green)'`	T st b te th vi m

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`vimcmd_replace_one_symbol`	`'[❮](bold purple)'`	T st b te th vi e
`vimcmd_replace_symbol`	`'[❮](bold purple)'`	T st b te th vi m
`vimcmd_visual_symbol`	`'[❮](bold yellow)'`	T st b te th vi m
`disabled`	`false`	D m

Variables

Variable	Example	Description
symbol		A mirror of either `success_symbol`, `error_symbol`, `vimcmd_symbol` or `vimcmd_replace_one_symbol` etc.

Examples

With custom error shape

```
# ~/.config/starship.toml

[character]
success_symbol = '[→](bold green) '
error_symbol = '[✗](bold red) '
```

Without custom error shape

```
# ~/.config/starship.toml

[character]
success_symbol = '[→](bold green) '
error_symbol = '[→](bold red) '
```

With custom vim shape

```
# ~/.config/starship.toml

[character]
vimcmd_symbol = '[V](bold green) '
```

CMake

The `cmake` module shows the currently installed version of CMake. By default the module will be activated if any of the following conditions are met:

- The current directory contains a `CMakeLists.txt` file
- The current directory contains a `CMakeCache.txt`

Options

Option	Default	
`format`	`'via [$symbol($version )]($style)'`	
`version_format`	`'v${raw}'`	
`symbol`	`'△ '`	
`detect_extensions`	`[]`	
`detect_files`	`['CMakeLists.txt', 'CMakeCache.txt']`	
`detect_folders`	`[]`	
`style`	`'bold blue'`	

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`disabled`	`false`	

Variables

Variable	Example	Description
version	`v3.17.3`	The version of cmake
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

COBOL / GNUCOBOL

The `cobol` module shows the currently installed version of COBOL. By default, the module will be shown if any of the following conditions are met:

- The current directory contains any files ending in `.cob` or `.COB`
- The current directory contains any files ending in `.cbl` or `.CBL`

Options

Option	Default	D
`symbol`	'⚙ '	Th us

		the of
`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for Av var ra ma mi pa
`style`	`'bold blue'`	Th for mo
`detect_extensions`	`['cbl', 'cob', 'CBL', 'COB']`	Wh ext sh tri mo
`detect_files`	`[]`	Wh file sh tri mo
`detect_folders`	`[]`	Wh fol sh tri mo
`disabled`	`false`	Dis the m

Variables

Variable	Example	Description
version	`v3.1.2.0`	The version of `cobol`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Command Duration

The `cmd_duration` module shows how long the last command took to execute. The module will be shown only if the command took longer than two seconds, or the `min_time` config value, if it exists.

> **Do not hook the DEBUG trap in Bash**
>
> If you are running Starship in `bash`, do not hook the `DEBUG` trap after running `eval $(starship init $0)`, or this module **will** break.

Bash users who need preexec-like functionality can use rcaloras's bash_preexec framework. Simply define the arrays `preexec_functions` and `precmd_functions` before running `eval`

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

Options

Option	Default	Desc
`min_time`	`2_000`	Shorte durati show (in millise
`show_milliseconds`	`false`	Show millise additi secon the du
`format`	`'took [$duration ]($style) '`	The fo the m
`style`	`'bold yellow'`	The st the m
`disabled`	`false`	Disabl `cmd_ n` mo
`show_notifications`	`false`	Show notific when comm compl
`min_time_to_notify`	`45_000`	Shorte durati notific millise
`notification_timeout`		Dura' show notific (in

		If unse notific timeo deterr daem all not daem honor optior

Variables

Variable	Example	Description
duration	`16m40s`	The time it took to execute the command
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[cmd_duration]
min_time = 500
format = 'underwent [$duration](bold yellow)'
```

Conda

The `conda` module shows the current Conda environment, if `$CONDA_DEFAULT_ENV` is set.

modifier, you may want to run `conda config --set changeps1 False`.

Options

Option	Default
`truncation_length`	`1`
`symbol`	`'🅒 '`
`style`	`'bold green'`
`format`	`'via [$symbol$environment](($style) '`
`ignore_base`	`true`

`disabled`	`false`

Variables

Variable	Example	Description
environment	`astronauts`	The current conda environment
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[conda]
format = '[$symbol$environment](dimmed green)
```

Container

The `container` module displays a symbol and

Options

Option	Default	Description
`symbol`	`'⬢'`	The symbol shown, when inside a container
`style`	`'bold red dimmed'`	The style for the module.
`format`	`'[$symbol \[$name \]]($style) '`	The format for the module.
`disabled`	`false`	Disables the `container` module.

Variables

Variable	Example	Description
name	`fedora-toolbox:35`	The name of the container
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
[container]
format = '[$symbol \[$name\]]($style) '
```

Crystal

The `crystal` module shows the currently installed version of Crystal. By default the module will be shown if any of the following conditions are met:

- The current directory contains a `shard.yml` file
- The current directory contains a `.cr` file

Options

Option	Default	De
`symbol`	`'🔮 '`	Th us dis the of
`format`	`'via [$symbol($versio n )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for Av va ra ma mi pa

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`style`	`'bold red'`	Th for mo
`detect_extensions`	`['cr']`	Wh ext sh tri mo
`detect_files`	`['shard.yml']`	Wh file sh tri mo
`detect_folders`	`[]`	Wh fol sh tri mo
`disabled`	`false`	Dis the c mo

Variables

Variable	Example	Description
version	`v0.32.1`	The version of `crystal`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[crystal]
format = 'via [✨ $version](bold blue) '
```

Daml

The `daml` module shows the currently used Daml SDK version when you are in the root directory of your Daml project. The `sdk-version` in the `daml.yaml` file will be used, unless it's overridden by the `DAML_SDK_VERSION` environment variable. By default the module will be shown if any of the following conditions are met:

- The current directory contains a `daml.yaml` file

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for Av var m m pa

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`symbol`	`'Λ '`	A f str rep the of
`style`	`'bold cyan'`	Th the
`detect_extensions`	`[]`	Wh ext sh tri mo
`detect_files`	`['daml.yaml']`	Wh file sh tri mo
`detect_folders`	`[]`	Wh fol sh tri mo
`disabled`	`false`	Dis da mo

Variables

Variable	Example	Description
version	`v2.2.0`	The version of `daml`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

Example

```
# ~/.config/starship.toml

[daml]
format = 'via [D $version](bold bright-green)
```

Dart

The `dart` module shows the currently installed version of Dart. By default the module will be shown if any of the following conditions are met:

- The current directory contains a file with `.dart` extension
- The current directory contains a `.dart_tool` directory
- The current directory contains a `pubspec.yaml`, `pubspec.yml` or `pubspec.lock` file

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for A va ra ma

		pa
`symbol`	`'🎯 '`	A f str rep the of
`detect_extensions`	`['dart']`	Wh ext sh tri mo
`detect_files`	`['pubspec.yaml', 'pubspec.yml', 'pubspec.lock']`	Wh file sh tri mo
`detect_folders`	`['.dart_tool']`	Wh fol sh tri mo
`style`	`'bold blue'`	Th the
`disabled`	`false`	Dis da mo

Variables

Variable	Example	Description
version	`v2.8.4`	The version of `dart`
symbol		Mirrors the value of option `symbol`

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[dart]
format = 'via [🔰 $version](bold red) '
```

Deno

The `deno` module shows you your currently installed version of Deno. By default the module will be shown if any of the following conditions are met:

- The current directory contains a `deno.json`, `deno.jsonc`, `mod.ts`, `mod.js`, `deps.ts` or `deps.js` file

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	T' fo Av va

		m m p
`symbol`	`'🦕 '`	A f str rep the of
`detect_extensions`	`[]`	Wh ext sh tri mo
`detect_files`	`['deno.json', 'deno.jsonc', 'mod.ts', 'mod.js', 'deps.ts', 'deps.js']`	Wh file sh tri mo
`detect_folders`	`[]`	Wh fol sh tri mo
`style`	`'green bold'`	Th the
`disabled`	`false`	Dis d mo

Variables

Variable	Example	Description
version	`v1.8.3`	The version of `deno`

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

Example

```
# ~/.config/starship.toml

[deno]
format = 'via [🦕 $version](green bold) '
```

Directory

The `directory` module shows the path to your current directory, truncated to three parent folders. Your directory will also be truncated to the root of the git repo that you're currently in.

When using the `fish_style_pwd_dir_length` option, instead of hiding the path that is truncated, you will see a shortened name of each directory based on the number you enable for the option.

For example, given `~/Dev/Nix/nixpkgs/pkgs` where `nixpkgs` is the repo root, and the option set to `1`. You will now see `~/D/N/nixpkgs/pkgs`, whereas before it would have been `nixpkgs/pkgs`.

Options

Option	Default

`truncation_length`	`3`
`truncate_to_repo`	`true`
`format`	`'[$path]($style)[$read_only]($read_only_style)'`
`style`	`'bold cyan'`
`disabled`	`false`
`read_only`	`'🔒'`
`read_only_style`	`'red'`
`truncation_symbol`	`''`
`before_repo_root_style`	
`repo_root_style`	
`repo_root_format`	`'[$before_root_pa($before_repo_roo)[$repo_root]($repo_root_style)[$path]($style)`

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

	`($read_only_style)`
`home_symbol`	`'~'`
`use_os_path_sep`	`true`

▸ This module has a few advanced configuration options that control how the directory is displayed.

Variables

Variable	Example	Description
path	`'D:/Projects'`	The current directory path
style*	`'black bold dimmed'`	Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

▸ The git repos have additional variables.

Example

```
# ~/.config/starship.toml

[directory]
truncation_length = 8
truncation_symbol = '…/'
```

Direnv

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

rc file, whether it is loaded, and whether it has been allowed by `direnv`.

Options

Option	Default
`format`	`'[$symbol$loaded/$allo ](($style) '`
`symbol`	`'direnv '`
`style`	`'bold orange'`
`disabled`	`true`
`detect_extensions`	`[]`
`detect_files`	`['.envrc']`
`detect_folders`	`[]`

`allowed_msg`	`'allowed'`
`denied_msg`	`'denied'`
`loaded_msg`	`'loaded'`
`unloaded_msg`	`'not loaded'`

Variables

Variable	Example	Description
loaded	`loaded`	Whether the current rc file is loaded.
allowed	`denied`	Whether the current rc file is allowed.

rc_path	`/home/test/.envrc`	The current rc file path.
symbol		Mirrors the value of option `symbol`.
style*	`red bold`	Mirrors the value of option `style`.

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[direnv]
disabled = false
```

Docker Context

The `docker_context` module shows the currently active Docker context if it's not set to `default` or if the `DOCKER_MACHINE_NAME`, `DOCKER_HOST` or `DOCKER_CONTEXT` environment variables are set (as they are meant to override the context in use).

Options

Option	Default	
`format`	`'via [$symbol$context]($style) '`	Th mo

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`symbol`	`'🐳 '`	Th be the co
`only_with_files`	`true`	Or the
`detect_extensions`	`[]`	Wh sh mo os
`detect_files`	`['docker-compose.yml', 'docker-compose.yaml', 'Dockerfile']`	Wh sh mo os
`detect_folders`	`[]`	Wh sh mo os
`style`	`'blue bold'`	Th mo
`disabled`	`false`	Dis d mo

Variables

Variable	Example	Description
context	`test_context`	The current docker context

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[docker_context]
format = 'via [🐋 $context](blue bold)'
```

Dotnet

The `dotnet` module shows the relevant version of the .NET Core SDK for the current directory. If the SDK has been pinned in the current directory, the pinned version is shown. Otherwise the module shows the latest installed version of the SDK.

By default this module will only be shown in your prompt when one or more of the following files are present in the current directory:

- `global.json`
- `project.json`
- `Directory.Build.props`
- `Directory.Build.targets`

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

- `*.fsproj`
- `*.xproj`

You'll also need the .NET Core SDK installed in order to use it correctly.

Internally, this module uses its own mechanism for version detection. Typically it is twice as fast as running `dotnet --version`, but it may show an incorrect version if your .NET project has an unusual directory layout. If accuracy is more important than speed, you can disable the mechanism by setting `heuristic = false` in the module options.

The module will also show the Target Framework Moniker (https://docs.microsoft.com/en-us/dotnet/standard/frameworks#supported-target-frameworks) when there is a `.csproj` file in the current directory.

Options

Option	Default
`format`	`'via [$symbol($version )(🎯 $tfm )]($style)'`
`version_format`	`'v${raw}'`

Home

Guide

Advanced Installation

Configuration

Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`symbol`	`'.NET '`
`heuristic`	`true`
`detect_extensions`	`['csproj', 'fsproj', 'xproj']`
`detect_files`	`['global.json', 'project.json', 'Directory.Build.props' 'Directory.Build.target , 'Packages.props']`
`detect_folders`	`[]`
`style`	`'bold blue'`
`disabled`	`false`

Variables

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

version	`v3.1.201`	The version of `dotnet` sdk
tfm	`netstandard2.0`	The Target Framework Moniker that the current project is targeting
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[dotnet]
symbol = '🥅 '
style = 'green'
heuristic = false
```

Elixir

The `elixir` module shows the currently installed version of Elixir and Erlang/OTP. By default the module will be shown if any of the following conditions

- The current directory contains a `mix.exs` file.

Options

Option	Default	De
`format`	`'via [$symbol($version \(OTP $otp_version\))]($style)'`	Th for mo eli
`version_format`	`'v${raw}'`	Th for Av var ra ma mi pa
`symbol`	`'💧 '`	Th us dis the of Eli
`detect_extensions`	`[]`	Wh ext sh tri mo
`detect_files`	`['mix.exs']`	Wh file sh tri mo
`detect_folders`	`[]`	Wh fol sh

Home

Guide

Advanced Installation

Configuration

Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

		mo
`style`	`'bold purple'`	Th the
`disabled`	`false`	Dis e mo

Variables

Variable	Example	Description
version	`v1.10`	The version of `elixir`
otp_version		The otp version of `elixir`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[elixir]
symbol = '🔮 '
```

Elm

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

the following conditions are met:

- The current directory contains a `elm.json` file
- The current directory contains a `elm-package.json` file
- The current directory contains a `.elm-version` file
- The current directory contains a `elm-stuff` folder
- The current directory contains `*.elm` files

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for Av var ra ma mi pa
`symbol`	`'🌳 '`	A f str rep the of
`detect_extensions`	`['elm']`	Wh ext sh tri m
`detect_files`	`['elm.json', 'elm-`	Wh file sh

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

	`'.elm-version']`	mo
`detect_folders`	`['elm-stuff']`	Wh fol sh tri mo
`style`	`'cyan bold'`	Th the
`disabled`	`false`	Dis e mo

Variables

Variable	Example	Description
version	`v0.19.1`	The version of `elm`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[elm]
format = 'via [ $version](cyan bold) '
```

Environment Variable

The `env_var` module displays the current value of a selected environment variables. The module will be shown only if any of the following conditions are met:

- The `variable` configuration option matches an existing environment variable
- The `variable` configuration option is not defined, but the `default` configuration option is

> **TIP**
>
> The order in which env_var modules are shown can be individually set by including `${env_var.foo}` in the top level `format` (as it includes a dot, you need to use `${...}`). By default, the `env_var` module will simply show all env_var modules in the order they were defined.

> **TIP**
>
> Multiple environmental variables can be displayed by using a `.` . (see example) If the `variable` configuration option is not set, the module will display value of variable under the name of text after the `.` character.
>
> Example: following configuration will display value of USER environment variable

```
# ~/.config/starship.toml
```

default = "unknown user"

Options

Option	Default	Description
`symbol`	`""`	The symbol used before displaying the variable value.
`variable`		The environment variable to be displayed.
`default`		The default value to be displayed when the selected variable is not defined.
`format`	`"with [$env_value]($style) "`	The format for the module.
`description`	`"<env_var module>"`	The description of the module that is shown when running `starship explain`.

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`disabled`	`false`	Disables the `env_var` module.

Variables

Variable	Example	Description
env_value	`Windows NT` (if *variable* would be `$OS`)	The environment value of option `variable`
symbol		Mirrors the value of option `symbol`
style*	`black bold dimmed`	Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[env_var]
variable = 'SHELL'
default = 'unknown shell'
```

Displaying multiple environmental variables:

```
# ~/.config/starship.toml

[env_var.SHELL]
variable = 'SHELL'
```

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

default = 'unknown user'

Erlang

The `erlang` module shows the currently installed version of Erlang/OTP. By default the module will be shown if any of the following conditions are met:

- The current directory contains a `rebar.config` file.
- The current directory contains a `erlang.mk` file.

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for Av var ra ma mi pa
`symbol`	`'[] '`	Th us dis the of
`style`	`'bold red'`	Th fo mo

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`detect_extensions`	`[]`	Wh ext sh tri mo
`detect_files`	`['rebar.config', 'elang.mk']`	Wh file sh tri mo
`detect_folders`	`[]`	Wh fol sh tri mo
`disabled`	`false`	Dis the e mo

Variables

Variable	Example	Description
version	`v22.1.3`	The version of `erlang`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
[erlang]
format = 'via [e $version](bold red) '
```

Fennel

The `fennel` module shows the currently installed version of Fennel. By default the module will be shown if any of the following conditions are met:

- The current directory contains a file with the `.fnl` extension

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for Av var ra ma mi pa
`symbol`	`'🌰 '`	Th us dis the of

`style`	`'bold green'`	Th for mo
`detect_extensions`	`['fnl']`	Wh ext sh tri mo
`detect_files`	`[]`	Wh file sh tri mo
`detect_folders`	`[]`	Wh fol sh tri mo
`disabled`	`false`	Dis the fo mo

Variables

Variable	Example	Description
version	`v1.2.1`	The version of `fennel`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[fennel]
symbol = '⫰ '
```

Fill

The `fill` module fills any extra space on the line with a symbol. If multiple `fill` modules are present in a line they will split the space evenly between them. This is useful for aligning other modules.

Options

Option	Default	Description
`symbol`	`'.'`	The symbol used to fill the line.
`style`	`'bold black'`	The style for the module.
`disabled`	`false`	Disables the `fill` module

Example

```
# ~/.config/starship.toml
format = 'AA $fill BB $fill CC'

[fill]
```

Produces a prompt that looks like:

```
AA ----------------------------------------
```

Fossil Branch

The `fossil_branch` module shows the name of the active branch of the check-out in your current directory.

Options

Option	Default	D
`format`	`'on [$symbol$branch](($style) '`	The the Use `'$` refe curr nam
`symbol`	`'🦎 '`	The use bra the in y dire
`style`	`'bold purple'`	The the
`truncation_length`	`2^63 - 1`	Tru Fos na gra

`truncation_symbol`	`'…'`	The use a br was You '' sym
`disabled`	`true`	Disa fo h r

Variables

Variable	Example	Description
branch	`trunk`	The active Fossil branch
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[fossil_branch]
symbol = '🦎 '
truncation_length = 4
truncation_symbol = ''
```

Fossil Metrics

The `fossil_metrics` module will show the number of added and deleted lines in the check-out in your current directory. At least v2.14 (2021-01-20) of Fossil is required.

Options

Option	Default	
`format`	`'([+$added] ($added_style) )([-$deleted] ($deleted_style) )'`	Th the
`added_style`	`'bold green'`	Th ad
`deleted_style`	`'bold red'`	Th de
`only_nonzero_diffs`	`true`	Re on ite
`disabled`	`true`	Dis f s

Variables

Variable	Example	Description

added	`1`	The current number of added lines
deleted	`2`	The current number of deleted lines
added_style*		Mirrors the value of option `added_style`
deleted_style*		Mirrors the value of option `deleted_style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[fossil_metrics]
added_style = 'bold blue'
format = '[+$added]($added_style)/[-$deleted]
```

Google Cloud (`gcloud`)

The `gcloud` module shows the current configuration for `gcloud` CLI. This is based on the `~/.config/gcloud/active_config` file and the `~/.config/gcloud/configurations/config_{CONFIG NAME}` file and the `CLOUDSDK_CONFIG` env var.

When the module is enabled it will always be active,

environment variables has been set.

Options

Option	Default
`format`	`'on [$symbol$account(@$domain)(\($region\))]($style) '`
`symbol`	`'☁️ '`
`region_aliases`	`{}`
`project_aliases`	`{}`
`detect_env_vars`	`[]`
`style`	`'bold blue'`
`disabled`	`false`

Variables

Variable	Example	Description
region	`us-central1`	The current GCP region
account	`foo`	The current GCP profile
domain	`example.com`	The current GCP profile domain
project		The current GCP project
active	`default`	The active config name written in `~/.config/gcloud/active_config`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Examples

Display account and project

```
# ~/.config/starship.toml

[gcloud]
```

Display active config name only

```
# ~/.config/starship.toml

[gcloud]
format = '[$symbol$active]($style) '
style = 'bold yellow'
```

Display account and aliased region

```
# ~/.config/starship.toml

[gcloud]
symbol = '🇬 '
[gcloud.region_aliases]
us-central1 = 'uc1'
asia-northeast1 = 'an1'
```

Display account and aliased project

```
# ~/.config/starship.toml

[gcloud]
format = 'on [$symbol$account(@$domain)(\($pr
[gcloud.project_aliases]
very-long-project-name = 'vlpn'
```

Git Branch

The `git_branch` module shows the active branch of the repo in your current directory.

Options

Option	Default
`always_show_remote`	`false`
`format`	`'on [$symbol$branch(:$remo ](($style) '`
`symbol`	`'🅒 '`
`style`	`'bold purple'`
`truncation_length`	`2^63 - 1`
`truncation_symbol`	`'…'`

`only_attached`	`false`
`ignore_branches`	`[]`
`disabled`	`false`

Variables

Variable	Example	Description
branch	`master`	The current branch name, falls back to `HEAD` if there's no current branch (e.g. git detached `HEAD`).
remote_name	`origin`	The remote name.
remote_branch	`master`	The name of the branch tracked

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

		`remote_name .`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[git_branch]
symbol = '🌱 '
truncation_length = 4
truncation_symbol = ''
ignore_branches = ['master', 'main']
```

Git Commit

The `git_commit` module shows the current commit hash and also the tag (if any) of the repo in your current directory.

Options

Option	Default	Des
`commit_hash_length`	`7`	The l of th

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

		comr hash
`format`	`'[\($hash$tag\)]($style) '`	The f for th modu
`style`	`'bold green'`	The s the n
`only_detached`	`true`	Only git cc hash in de HEAI
`tag_disabled`	`true`	Disab show info i `git_t m`
`tag_max_candidates`	`0`	How comr consi tag d The c only exact matc
`tag_symbol`	`' 🏷 '`	Tag s prefi info s
`disabled`	`false`	Disab `git_t m`

Variables

Variable	Example	Description

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

hash	`b703eb3`	The current git commit hash
tag	`v1.0.0`	The tag name if showing tag info is enabled.
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[git_commit]
commit_hash_length = 4
tag_symbol = '🔖 '
```

Git State

The `git_state` module will show in directories which are part of a git repository, and where there is an operation in progress, such as: *REBASING*, *BISECTING*, etc. If there is progress information (e.g., REBASING 3/10), that information will be shown too.

Options

Option	Default
`rebase`	`'REBASING'`

`merge`	`'MERGING'`
`revert`	`'REVERTING'`
`cherry_pick`	`'CHERRY-PICKING'`
`bisect`	`'BISECTING'`
`am`	`'AM'`

`am_or_rebase`	`'AM/REBASE'`
`style`	`'bold yellow'`
`format`	`'\([$state( $progress_current/$progress_ ](($style)\) '`
`disabled`	`false`

Variables

Variable	Example	Description
state	`REBASING`	The current state of the repo
progress_current	`1`	The current operation progress
progress_total	`2`	The total operation progress

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[git_state]
format = '[\($state( $progress_current of $pr
cherry_pick = '[🍒 PICKING](bold red)'
```

Git Metrics

The `git_metrics` module will show the number of added and deleted lines in the current git repository.

> **TIP**
>
> This module is disabled by default. To enable it, set `disabled` to `false` in your configuration file.

Options

Option	Default	D

`added_style`	`'bold green'`	Th the co
`deleted_style`	`'bold red'`	Th the co
`only_nonzero_diffs`	`true`	Re sta for ite
`format`	`'([+$added] ($added_style) )([-$deleted] ($deleted_style) )'`	Th for mo
`disabled`	`true`	Dis g s
`ignore_submodules`	`false`	Ign ch sul

Variables

Variable	Example	Description
added	`1`	The current number of added lines
deleted	`2`	The current number of deleted lines
added_style*		Mirrors the value of option `added_style`

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

deleted_style*		Mirrors the value of option `deleted_style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[git_metrics]
added_style = 'bold blue'
format = '[+$added]($added_style)/[-$deleted]
```

Git Status

The `git_status` module shows symbols representing the state of the repo in your current directory.

> **TIP**
>
> The Git Status module is very slow in Windows directories (for example under `/mnt/c/`) when in a WSL environment. You can disable the module or use the `windows_starship` option to use a Windows-native Starship executable to compute `git_status` for those paths.

Options

Home

Guide

Advanced Installation

Configuration

Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`format`	`'([\[$all_status$ahead_d\]]($style) )'`
`conflicted`	`'='`
`ahead`	`'⇡'`
`behind`	`'⇣'`
`diverged`	`'⇕'`
`up_to_date`	`''`
`untracked`	`'?'`
`stashed`	`'$'`
`modified`	`'!'`
`staged`	`'+'`
`renamed`	`'»'`
`deleted`	`'✘'`

`typechanged`	`""`
`style`	`'bold red'`
`ignore_submodules`	`false`
`disabled`	`false`
`windows_starship`	

Variables

The following variables can be used in `format`:

Variable	
`all_status`	Shortcut for `$conflicted$stashed$delet`
`ahead_behind`	Displays `diverged`, `ahead`, based on the current status of th
`conflicted`	Displays `conflicted` when this

`untracked`	Displays `untracked` when there directory.
`stashed`	Displays `stashed` when a stash
`modified`	Displays `modified` when there directory.
`staged`	Displays `staged` when a new fi
`renamed`	Displays `renamed` when a renar area.
`deleted`	Displays `deleted` when a file's area.
`typechanged`	Displays `typechange` when a fil area.
style*	Mirrors the value of option `styl`

*: This variable can only be used as a part of a style string

The following variables can be used in `diverged`:

Variable	Description
`ahead_count`	Number of commits ahead of the tracking branch
`behind_count`	Number of commits behind the tracking branch

The following variables can be used in `conflicted`, `ahead`, `behind`, `untracked`, `stashed`, `modified`, `staged`, `renamed` and `deleted`:

Variable	Description
`count`	Show the number of files

Example

```
# ~/.config/starship.toml

[git_status]
conflicted = '🏳'
ahead = '🏎💨'
behind = '😰'
diverged = '😵'
up_to_date = '✓'
untracked = '🤷'
stashed = '📦'
modified = '📝'
staged = '[++\($count\)](green)'
renamed = '👅'
deleted = '🗑'
```

Show ahead/behind count of the branch being tracked

```
# ~/.config/starship.toml

[git_status]
ahead = '⇡${count}'
diverged = '⇕⇡${ahead_count}⇣${behind_count}'
behind = '⇣${count}'
```

Use Windows Starship executable on Windows paths in WSL

```
# ~/.config/starship.toml

[git_status]
windows_starship = '/mnt/c/Users/username/sc
```

Go

The `golang` module shows the currently installed version of Go. By default the module will be shown if any of the following conditions are met:

- The current directory contains a `go.mod` file
- The current directory contains a `go.sum` file
- The current directory contains a `go.work` file
- The current directory contains a `glide.yaml` file
- The current directory contains a `Gopkg.yml` file
- The current directory contains a `Gopkg.lock` file
- The current directory contains a `.go-version` file
- The current directory contains a `Godeps` directory
- The current directory contains a file with the `.go` extension

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for Av var ra ma mi pa
`symbol`	`'🐹 '`	A f str rep

		of
`detect_extensions`	`['go']`	Wh ext sh tri mo
`detect_files`	`['go.mod', 'go.sum', 'go.work', 'glide.yaml', 'Gopkg.yml', 'Gopkg.lock', '.go-version']`	Wh file sh tri mo
`detect_folders`	`['Godeps']`	Wh fol sh tri mo
`style`	`'bold cyan'`	Th the
`not_capable_style`	`'bold red'`	Th the wh dir the file ma ins ver
`disabled`	`false`	Dis g mo

Variables

Variable	Example	Description

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

version	`v1.12.1`	The version of `go`
mod_version	`1.16`	`go` version requirement as set in the go directive of `go.mod`. Will only show if the version requirement does not match the `go` version.
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[golang]
format = 'via [🏎💨 $version](bold cyan) '
```

Using `mod_version`

```
# ~/.config/starship.toml

[golang]
format = 'via [$symbol($version )($mod_versio
```

Guix-shell

The `guix_shell` module shows the guix-shell environment. The module will be shown when inside a guix-shell environment.

Options

Option	Default	Description
`format`	`'via [$symbol]($style) '`	The format for the module.
`symbol`	`'🐃 '`	A format string representing the symbol of guix-shell.
`style`	`'yellow bold'`	The style for the module.
`disabled`	`false`	Disables the `guix_shell` module.

Variables

Variable	Example	Description
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

Example

```
# ~/.config/starship.toml

[guix_shell]
disabled = true
format = 'via [🐃](yellow bold) '
```

Gradle

The `gradle` module shows the version of the Gradle Wrapper currently used in the project directory.

By default the module will be shown if any of the following conditions are met:

- The current directory contains a `gradle/wrapper/gradle-wrapper.properties` directory.
- The current directory contains a file ending with `.gradle` or `.gradle.kts`.

The `gradle` module is only able to read your Gradle Wrapper version from your config file, we don't execute your wrapper, because of the security concerns.

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	Th fo mo

Home

Guide

Advanced Installation

Configuration

Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`version_format`	`'v${raw}'`	Th for Av var ra ma mi pa
`symbol`	`'G '`	A f str rep the of
`detect_extensions`	`['gradle', 'gradle.kts']`	Wh ext sh tri mo
`detect_files`	`[]`	Wh file sh tri mo
`detect_folders`	`['gradle']`	Wh fol sh tri mo
`style`	`'bold bright-cyan'`	Th the
`disabled`	`false`	Dis g mo
`recursive`	`false`	En re fin the g

Variables

Variable	Example	Description
version	`v7.5.1`	The version of `gradle`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Haskell

The `haskell` module finds the current selected GHC version and/or the selected Stack snapshot.

By default the module will be shown if any of the following conditions are met:

- The current directory contains a `stack.yaml` file
- The current directory contains any `.hs`, `.cabal`, or `.hs-boot` file

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	T fo mo

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`symbol`	`'λ '`	A str rep the of
`detect_extensions`	`['hs', 'cabal', 'hs-boot']`	Wh ext sh tri mo
`detect_files`	`['stack.yaml', 'cabal.project']`	Wh file sh tri mo
`detect_folders`	`[]`	Wh fol sh tri mo
`style`	`'bold purple'`	Th the
`disabled`	`false`	Dis ha mo

Variables

Variable	Example	Description
version		`ghc_version` or `snapshot` depending on whether the current project is a Stack project

snapshot	`lts-18.12`	Currently selected Stack snapshot
ghc_version	`9.2.1`	Currently installed GHC version
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Haxe

The `haxe` module shows the currently installed version of Haxe. By default the module will be shown if any of the following conditions are met:

- The current directory contains a `project.xml`, `Project.xml`, `application.xml`, `haxelib.json`, `hxformat.json` or `.haxerc` file
- The current directory contains a `.haxelib` or a `haxe_libraries` directory
- The current directory contains a file with the `.hx` or `.hxml` extension

Options

Option	Default	D

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`format`	`'via [$symbol($version )]($style)'`	T fo m
`version_format`	`'v${raw}'`	T fo A va
`detect_extensions`	`['hx', 'hxml']`	W ex sh tr m
`detect_files`	`['project.xml', 'Project.xml', 'application.xml', 'haxelib.json', 'hxformat.json', '.haxerc']`	W fi sh tr m
`detect_folders`	`['.haxelib', 'haxe_libraries']`	W fo sh tr m
`symbol`	`'⌘ '`	A sh re th o
`style`	`'bold fg:202'`	T th
`disabled`	`false`	m

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

Variables

Variable	Example	Description
version	`v4.2.5`	The version of `haxe`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[haxe]
format = "via [⌘ $version](bold fg:202) "
```

Helm

The `helm` module shows the currently installed version of Helm. By default the module will be shown if any of the following conditions are met:

- The current directory contains a `helmfile.yaml` file
- The current directory contains a `Chart.yaml` file

Options

`format`	`'via [$symbol($version )]($style)'`	T fo m
`version_format`	`'v${raw}'`	T fo A va
`detect_extensions`	`[]`	W ex sh tr m
`detect_files`	`['helmfile.yaml', 'Chart.yaml']`	W fi sh tr m
`detect_folders`	`[]`	W fo sh tr m
`symbol`	`'⎈ '`	A sh re th o
`style`	`'bold white'`	T th
`disabled`	`false`	D m

Variables

Variable	Example	Description
version	`v3.1.1`	The version of `helm`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[helm]
format = 'via [⎈ $version](bold white) '
```

Hostname

The `hostname` module shows the system hostname.

Options

Option	Default
`ssh_only`	`true`

`ssh_symbol`	`'🌐 '`
`trim_at`	`'.'`
`detect_env_vars`	`[]`
`format`	`'[$ssh_symbol$hostname](($style) in '`
`style`	`'bold dimmed green'`
`disabled`	`false`

Variables

hostname	`computer`	The hostname of the computer
style*		Mirrors the value of option `style`
ssh_symbol	`'🌍 '`	The symbol to represent when connected to SSH session

*: This variable can only be used as a part of a style string

Examples

Always show the hostname

```
# ~/.config/starship.toml

[hostname]
ssh_only = false
format = '[$ssh_symbol](bold blue) on [$hostna
trim_at = '.companyname.com'
disabled = false
```

Hide the hostname in remote tmux sessions

```
# ~/.config/starship.toml

[hostname]
ssh_only = false
detect_env_vars = ['!TMUX', 'SSH_CONNECTION']
disabled = false
```

Java

The `java` module shows the currently installed version of Java. By default the module will be shown if any of the following conditions are met:

- The current directory contains a `pom.xml`, `build.gradle.kts`, `build.sbt`, `.java-version`, `deps.edn`, `project.clj`, `build.boot`, or `.sdkmanrc` file
- The current directory contains a file with the `.java`, `.class`, `.gradle`, `.jar`, `.clj`, or `.cljc` extension

Options

Option	Default	
`format`	`'via [${symbol}(${version})](${style})'`	
`version_format`	`'v${raw}'`	
`detect_extensions`	`['java', 'class', 'gradle', 'jar', 'cljs', 'cljc']`	

`detect_files`	`['pom.xml', 'build.gradle.kts', 'build.sbt', '.java-version', 'deps.edn', 'project.clj', 'build.boot', '.sdkmanrc']`
`detect_folders`	`[]`
`symbol`	`'☕ '`
`style`	`'red dimmed'`
`disabled`	`false`

Variables

Variable	Example	Description
version	`v14`	The version of `java`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style

Example

```
# ~/.config/starship.toml

[java]
symbol = '🌟 '
```

Jobs

The `jobs` module shows the current number of jobs running. The module will be shown only if there are background jobs running. The module will show the number of jobs running if there are at least 2 jobs, or more than the `number_threshold` config value, if it exists. The module will show a symbol if there is at least 1 job, or more than the `symbol_threshold` config value, if it exists. You can set both values to 0 in order to *always* show the symbol and number of jobs, even if there are 0 jobs running.

The default functionality is:

- 0 jobs -> Nothing is shown.
- 1 job -> `symbol` is shown.
- 2 jobs or more -> `symbol` + `number` are shown.

> **WARNING**
>
> This module is not supported on tcsh and nu.

> **WARNING**
>
> The `threshold` option is deprecated, but if

job, or more than the `threshold` config value, if it exists. If `threshold` is set to 0, then the module will also show when there are 0 jobs running.

Options

Option	Default	
`threshold` *	`1`	Sh job
`symbol_threshold`	`1`	Sh the lea sy d .
`number_threshold`	`2`	Sh of co nu d .
`format`	`'[$symbol$number ]($style) '`	Th mo
`symbol`	`'✦'`	Th rep sy
`style`	`'bold blue'`	Th mo
`disabled`	`false`	Dis mo

*: This option is deprecated, please use the `number_threshold` and `symbol_threshold` options

Variables

Variable	Example	Description
number	`1`	The number of jobs
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[jobs]
symbol = '+ '
number_threshold = 4
symbol_threshold = 0
```

Julia

The `julia` module shows the currently installed version of Julia. By default the module will be shown if any of the following conditions are met:

- The current directory contains a `Project.toml` file
- The current directory contains a `Manifest.toml` file
- The current directory contains a file with the `.jl`

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for Av var ra ma mi pa
`detect_extensions`	`['jl']`	Wh ext sh tri mo
`detect_files`	`['Project.toml', 'Manifest.toml']`	Wh file sh tri mo
`detect_folders`	`[]`	Wh fol sh tri mo
`symbol`	`'🅒 '`	A f str rep the of
`style`	`'bold purple'`	Th the

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`disabled`	`false`	Dis ju mo

Variables

Variable	Example	Description
version	`v1.4.0`	The version of `julia`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[julia]
symbol = '∴ '
```

Kotlin

The `kotlin` module shows the currently installed version of Kotlin. By default the module will be shown if any of the following conditions are met:

- The current directory contains a `.kt` or a `.kts` file

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for Av var ra ma mi pa
`detect_extensions`	`['kt', 'kts']`	Wh ext sh tri mo
`detect_files`	`[]`	Wh file sh tri mo
`detect_folders`	`[]`	Wh fol sh tri mo
`symbol`	`'🅺 '`	A f str rep the o

`style`	`'bold blue'`	Th the
`kotlin_binary`	`'kotlin'`	Co the bir Sta exe wh ge ver
`disabled`	`false`	Dis k mo

Variables

Variable	Example	Description
version	`v1.4.21`	The version of `kotlin`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[kotlin]
symbol = '🅺 '
```

```
[kotlin]
# Uses the Kotlin Compiler binary to get the
kotlin_binary = 'kotlinc'
```

Kubernetes

Displays the current Kubernetes context name and, if set, the namespace, user and cluster from the kubeconfig file. The namespace needs to be set in the kubeconfig file, this can be done via `kubectl config set-context starship-context --namespace astronaut`. Similarly, the user and cluster can be set with `kubectl config set-context starship-context --user starship-user` and `kubectl config set-context starship-context --cluster starship-cluster`. If the `$KUBECONFIG` env var is set the module will use that if not it will use the `~/.kube/config`.

> **TIP**
>
> This module is disabled by default. To enable it, set `disabled` to `false` in your configuration file.
>
> When the module is enabled it will always be active, unless any of `detect_extensions`, `detect_files` or `detect_folders` have been set in which case the module will only be active in directories that match those conditions.

Options

WARNING

The `context_aliases` and `user_aliases` options are deprecated. Use `contexts` and the corresponding `context_alias` and `user_alias` options instead.

Option	Default	
`symbol`	`'☸ '`	
`format`	`'[$symbol$context ( \($namespace \))](style) in '`	
`style`	`'cyan bold'`	
`context_aliases *`	`{}`	
`user_aliases *`	`{}`	
`detect_extensions`	`[]`	

`detect_files`	`[]`	
`detect_folders`	`[]`	
`contexts`	`[]`	
`disabled`	`true`	

*: This option is deprecated, please add `contexts` with the corresponding `context_alias` and `user_alias` options instead.

To customize the style of the module for specific environments, use the following configuration as part of the `contexts` list:

Variable	Description
`context_pattern`	**Required** Regular expression to match current Kubernetes context name.
`user_pattern`	Regular expression to match current Kubernetes user name.

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`context_alias`	Context alias to display instead of the full context name.
`user_alias`	User alias to display instead of the full user name.
`style`	The style for the module when using this context. If not set, will use module's style.
`symbol`	The symbol for the module when using this context. If not set, will use module's symbol.

Note that all regular expression are anchored with `^<pattern>$` and so must match the whole string. The `*_pattern` regular expressions may contain capture groups, which can be referenced in the corresponding alias via `$name` and `$N` (see example below and the rust Regex::replace() documentation).

Variables

Variable	Example	Description
context	`starship-context`	The current kubernetes context name
namespace	`starship-namespace`	If set, the current kubernetes namespace

user	`starship-user`	If set, the current kubernetes user
cluster	`starship-cluster`	If set, the current kubernetes cluster
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[kubernetes]
format = 'on [⛵ ($user on )($cluster in )$cor
disabled = false
contexts = [
  { context_pattern = "dev.local.cluster.k8s",
]
```

Only show the module in directories that contain a `k8s` file.

```
# ~/.config/starship.toml

[kubernetes]
disabled = false
```

Kubernetes Context specific config

The `contexts` configuration option is used to customise what the current Kubernetes context name looks like (style and symbol) if the name matches the defined regular expression.

```
# ~/.config/starship.toml

[[kubernetes.contexts]]
# "bold red" style + default symbol when Kube
# equals "admin_user"
context_pattern = "production"
user_pattern = "admin_user"
style = "bold red"
context_alias = "prod"
user_alias = "admin"

[[kubernetes.contexts]]
# "green" style + a different symbol when Kub
context_pattern = ".*openshift.*"
style = "green"
symbol = "💔 "
context_alias = "openshift"

[[kubernetes.contexts]]
# Using capture groups
# Contexts from GKE, AWS and other cloud prov
# The following entry matches on the GKE form
# and renames every matching kube context int
context_pattern = "gke_.*_(?P<cluster>[\\w-]+
context_alias = "gke-$cluster"
```

Line Break

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

Options

Option	Default	Description
`disabled`	`false`	Disables the `line_break` module, making the prompt a single line.

Example

```
# ~/.config/starship.toml

[line_break]
disabled = true
```

Local IP

The `localip` module shows the IPv4 address of the primary network interface.

Options

Option	Default	Description
`ssh_only`	`true`	Only show IP address when connected to an SSH session.
`format`	`'[$localipv4](($style) '`	The format for the

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`style`	`'bold yellow'`	The style for the module.
`disabled`	`true`	Disables the `localip` module.

Variables

Variable	Example	Description
localipv4	192.168.1.13	Contains the primary IPv4 address
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[localip]
ssh_only = false
format = '@[$localipv4](bold red) '
disabled = false
```

Lua

The `lua` module shows the currently installed version of Lua. By default the module will be shown if any of

- The current directory contains a `.lua-version` file
- The current directory contains a `lua` directory
- The current directory contains a file with the `.lua` extension

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for Av var ra ma mi pa
`symbol`	`'🌙 '`	A f str rep the of
`detect_extensions`	`['lua']`	Wh ext sh tri mo
`detect_files`	`['.lua-version']`	Wh file sh tri m
`detect_folders`	`['lua']`	Wh fol

		tri mo
`style`	`'bold blue'`	Th the
`lua_binary`	`'lua'`	Co the bir Sta exe wh ge ver
`disabled`	`false`	Dis lu mo

Variables

Variable	Example	Description
version	`v5.4.0`	The version of `lua`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[lua]
```

Memory Usage

The `memory_usage` module shows current system memory and swap usage.

By default the swap usage is displayed if the total system swap is non-zero.

> **TIP**
>
> This module is disabled by default. To enable it, set `disabled` to `false` in your configuration file.

Options

Option	Default	Description
`threshold`	`75`	Hide the memory usage unless it exceeds this percentage.
`format`	`'via $symbol [${ram}( | ${swap})] ($style) '`	The format for the module.
`symbol`	`'🐏'`	The symbol used before displaying the memory usage.

`style`	`'bold dimmed white'`	The style for the module.
`disabled`	`true`	Disables the `memory_usage` module.

Variables

Variable	Example	Description
ram	`31GiB/65GiB`	The usage/total RAM of the current system memory.
ram_pct	`48%`	The percentage of the current system memory.
swap**	`1GiB/4GiB`	The swap memory size of the current system swap memory file.
swap_pct**	`77%`	The swap memory percentage of the current system swap memory file.
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

*: This variable can only be used as a part of a style string **: The SWAP file information is only displayed if detected on the current system

Example

```
# ~/.config/starship.toml

[memory_usage]
disabled = false
threshold = -1
symbol = ' '
style = 'bold dimmed green'
```

Meson

The `meson` module shows the current Meson developer environment status.

By default the Meson project name is displayed, if `$MESON_DEVENV` is set.

Options

Option	Default	De
`truncation_length`	`2^32 - 1`	Tru pr na N g
`truncation_symbol`	`'…'`	Th us

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

		pr na tru Yo ' sy
`format`	`'via [$symbol$project ]($style) '`	Th for mo
`symbol`	`'⬢ '`	Th us dis the na
`style`	`'blue bold'`	Th for mo
`disabled`	`false`	Dis the mo

Variables

Variable	Example	Description
project	`starship`	The current Meson project name
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[meson]
disabled = false
truncation_symbol = '--'
symbol = ' '
style = 'bold dimmed green'
```

Mercurial Branch

The `hg_branch` module shows the active branch and topic of the repo in your current directory.

Options

Option	Default
`symbol`	`' '`
`style`	`'bold purple'`
`format`	`'on [$symbol$branch(:$topic)]($style) '`

`truncation_length`	`2^63 - 1`
`truncation_symbol`	`'…'`
`disabled`	`true`

Variables

Variable	Example	Description
branch	`master`	The active mercurial branch
topic	`feature`	The active mercurial topic
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
[hg_branch]
format = 'on [🌱 $branch](bold purple)'
truncation_length = 4
truncation_symbol = ''
```

Nim

The `nim` module shows the currently installed version of Nim. By default the module will be shown if any of the following conditions are met:

- The current directory contains a `nim.cfg` file
- The current directory contains a file with the `.nim` extension
- The current directory contains a file with the `.nims` extension
- The current directory contains a file with the `.nimble` extension

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for Av var ra m pa

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`symbol`	`'👑 '`	Th us dis the of
`detect_extensions`	`['nim', 'nims', 'nimble']`	Wh ext sh tri mo
`detect_files`	`['nim.cfg']`	Wh file sh tri mo
`detect_folders`	`[]`	Wh fol sh tri mo
`style`	`'bold yellow'`	Th for mo
`disabled`	`false`	Dis the mo

Variables

Variable	Example	Description
version	`v1.2.0`	The version of `nimc`
symbol		Mirrors the value of option `symbol`

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[nim]
style = 'yellow'
symbol = '🎣 '
```

Nix-shell

The `nix_shell` module shows the nix-shell environment. The module will be shown when inside a nix-shell environment.

Options

Option	Default	Description
`format`	`'via [$symbol$state( \($name\))]($style) '`	The format for the module.
`symbol`	`'❄️ '`	A format string representing the symbol of nix-shell.

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`style`	`'bold blue'`	The style fo the module.
`impure_msg`	`'impure'`	A format string show when the shell is impure.
`pure_msg`	`'pure'`	A format string show when the shell is pure
`unknown_msg`	`''`	A format string show when it is unknown if the shell is pure/impure
`disabled`	`false`	Disables the `nix_shell` module.
`heuristic`	`false`	Attempts to detect new `nix shell` -style shells with a heuristic.

Variables

Variable	Example	Description
state	`pure`	The state of the nix-shell
name	`lorri`	The name of the nix-shell

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[nix_shell]
disabled = true
impure_msg = '[impure shell](bold red)'
pure_msg = '[pure shell](bold green)'
unknown_msg = '[unknown shell](bold yellow)'
format = 'via [ $state( \($name\))](bold blu
```

Node.js

The `nodejs` module shows the currently installed version of Node.js. By default the module will be shown if any of the following conditions are met:

- The current directory contains a `package.json` file
- The current directory contains a `.node-version` file
- The current directory contains a `.nvmrc` file
- The current directory contains a `node_modules` directory
- The current directory contains a file with the `.js`, `.mjs` or `.cjs` extension

Options

Option	Default	D
`format`	`'via [$symbol($version )]($style)'`	T fo m
`version_format`	`'v${raw}'`	T fo A va
`symbol`	`'⬢ '`	A st re th o
`detect_extensions`	`['js', 'mjs', 'cjs', 'ts', 'mts', 'cts']`	W ex sh tr m
`detect_files`	`['package.json', '.node-version']`	W fi sh tr m
`detect_folders`	`['node_modules']`	W fo sh m

`style`	`'bold green'`	T th
`disabled`	`false`	D i m
`not_capable_style`	`'bold red'`	T th w e p p d m N v

Variables

Variable	Example	Description
version	`v13.12.0`	The version of `node`
engines_version	`>=12.0.0`	`node` version requirement as set in the engines property of `package.json`. Will only show if the version requirement does not match the `node` version.
symbol		Mirrors the value of option `symbol`

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[nodejs]
format = 'via [🤖 $version](bold green) '
```

OCaml

The `ocaml` module shows the currently installed version of OCaml. By default the module will be shown if any of the following conditions are met:

- The current directory contains a file with `.opam` extension or `_opam` directory
- The current directory contains a `esy.lock` directory
- The current directory contains a `dune` or `dune-project` file
- The current directory contains a `jbuild` or `jbuild-ignore` file
- The current directory contains a `.merlin` file
- The current directory contains a file with `.ml`, `.mli`, `.re` or `.rei` extension

Options

`format`	`'via [$symbol($v )(\($switch_indic ) )]($style)'`
`version_format`	`'v${raw}'`
`symbol`	`'🐪 '`
`global_switch_indicator`	`''`
`local_switch_indicator`	`'*'`
`detect_extensions`	`['opam', 'ml', 'r 'rei']`
`detect_files`	`['dune', 'dune-p 'jbuild', 'jbuild '.merlin']`

`detect_folders`	`['_opam', 'esy.l`
`style`	`'bold yellow'`
`disabled`	`false`

Variables

Variable	Example	Description
version	`v4.10.0`	The version of `ocaml`
switch_name	`my-project`	The active OPAM switch
switch_indicator		Mirrors the value of `indicator` for currently active OPAM switch
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

Example

```
# ~/.config/starship.toml

[ocaml]
format = 'via [🐫 $version]($style) '
```

Open Policy Agent

The `opa` module shows the currently installed version of the OPA tool. By default the module will be shown if the current directory contains a `.rego` file.

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for Av var ra ma mi pa
`symbol`	`'🪖 '`	A f str re the of

`detect_extensions`	`['rego']`	ext sh tri mo
`detect_files`	`[]`	Wh file sh tri mo
`detect_folders`	`[]`	Wh fol sh tri mo
`style`	`'bold blue'`	Th the
`disabled`	`false`	Dis op mo

Variables

Variable	Example	Description
version	`v0.44.0`	The version of `opa`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[opa]
format = 'via [⛑  $version](bold red) '
```

OpenStack

The `openstack` module shows the current OpenStack cloud and project. The module only active when the `OS_CLOUD` env var is set, in which case it will read `clouds.yaml` file from any of the default locations. to fetch the current project in use.

Options

Option	Default	Description
`format`	`'on [$symbol$cloud(\($project\))]($style) '`	The format for the module.
`symbol`	`'☁️ '`	The symbol used before displaying the current OpenStack cloud.
`style`	`'bold yellow'`	The style for the module.

`disabled`	`false`	Disables the `openstack` module.

Variables

Variable	Example	Description
cloud	`corp`	The current OpenStack cloud
project	`dev`	The current OpenStack project
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[openstack]
format = 'on [$symbol$cloud(\($project\))]($st
style = 'bold yellow'
symbol = '☁️ '
```

OS

The `os` module shows the current operating system.

WARNING

The os_info crate used by this module is known to be inaccurate on some systems.

TIP

This module is disabled by default. To enable it, set `disabled` to `false` in your configuration file.

Options

Option	Default	Description
`format`	`'[$symbol] ($style)'`	The format for the module.
`style`	`'bold white'`	The style for the module.
`disabled`	`true`	Disables the `os` module.
`symbols`		A table that maps each operating system to its symbol.

`symbols` allows you to define arbitrary symbols to display for each operating system type. Operating system types not defined by your configuration use the default symbols table below. All operating systems currently supported by the module are listed below. If you would like an operating system to be added, feel

```
# This is the default symbols table.
[os.symbols]
Alpaquita = "🔔 "
Alpine = "🏔️ "
Amazon = "🙂 "
Android = "🤖 "
Arch = "🎗️ "
Artix = "🎗️ "
CentOS = "💠 "
Debian = "🌀 "
DragonFly = "🐉 "
Emscripten = "🔗 "
EndeavourOS = "🚀 "
Fedora = "🎩 "
FreeBSD = "😈 "
Garuda = "🦅 "
Gentoo = "🗜️ "
HardenedBSD = "🛡️ "
Illumos = "🐦 "
Linux = "🐧 "
Mabox = "📦 "
Macos = "🍎 "
Manjaro = "🥭 "
Mariner = "🌊 "
MidnightBSD = "🌘 "
Mint = "🌿 "
NetBSD = "🚩 "
NixOS = "❄️ "
OpenBSD = "🐡 "
OpenCloudOS = "☁️ "
openEuler = "🦉 "
openSUSE = "🦎 "
OracleLinux = "🦴 "
Pop = "🍭 "
Raspbian = "🍓 "
Redhat = "🎩 "
RedHatEnterprise = "🎩 "
Redox = "🧪 "
Solus = "⛵ "
SUSE = "🦎 "
```

Variables

Variable	Example	Description
symbol	🎗	The current operating system symbol from advanced option `symbols`
name	`Arch Linux`	The current operating system name
type	`Arch`	The current operating system type
codename		The current operating system codename, if applicable
edition		The current operating system edition, if applicable
version		The current operating system version, if applicable
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
[os]
format = "on [($name )]($style)"
style = "bold blue"
disabled = false

[os.symbols]
Windows = "🪟 "
Arch = "Arch is the best! "
```

Package Version

The `package` module is shown when the current directory is the repository for a package, and shows its current version. The module currently supports `npm`, `nimble`, `cargo`, `poetry`, `python`, `composer`, `gradle`, `julia`, `mix`, `helm`, `shards`, `daml` and `dart` packages.

- **npm** – The `npm` package version is extracted from the `package.json` present in the current directory
- **Cargo** – The `cargo` package version is extracted from the `Cargo.toml` present in the current directory
- **Nimble** - The `nimble` package version is extracted from the `*.nimble` file present in the current directory with the `nimble dump` command
- **Poetry** – The `poetry` package version is extracted from the `pyproject.toml` present in the current directory
- **Python** - The `python` package version is extracted from a PEP 621 compliant `pyproject.toml` or a `setup.cfg` present in the current directory

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

current directory

- **Gradle** – The `gradle` package version is extracted from the `build.gradle` present in the current directory
- **Julia** - The package version is extracted from the `Project.toml` present in the current directory
- **Mix** - The `mix` package version is extracted from the `mix.exs` present in the current directory
- **Helm** - The `helm` chart version is extracted from the `Chart.yaml` present in the current directory
- **Maven** - The `maven` package version is extracted from the `pom.xml` present in the current directory
- **Meson** - The `meson` package version is extracted from the `meson.build` present in the current directory
- **Shards** - The `shards` package version is extracted from the `shard.yml` present in the current directory
- **V** - The `vlang` package version is extracted from the `v.mod` present in the current directory
- **SBT** - The `sbt` package version is extracted from the `build.sbt` present in the current directory
- **Daml** - The `daml` package version is extracted from the `daml.yaml` present in the current directory
- **Dart** - The `dart` package version is extracted from the `pubspec.yaml` present in the current directory

> ⚠️ The version being shown is that of the package whose source code is in your current directory, not your package manager.

Options

Option	Default	Desc
`format`	`'is [$symbol$version ]($style) '`	The f for th modu
`symbol`	`'📦 '`	The s used displ the v the pack
`version_format`	`'v${raw}'`	The v form Avail vars raw maj min pat
`style`	`'bold 208'`	The s for th modu
`display_private`	`false`	Enab displ versi pack mark priva
`disabled`	`false`	Disab the pac modu

Variables

version	`v1.0.0`	The version of your package
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[package]
format = 'via [🎁 $version](208 bold) '
```

Perl

The `perl` module shows the currently installed version of Perl. By default the module will be shown if any of the following conditions are met:

- The current directory contains a `Makefile.PL` or `Build.PL` file
- The current directory contains a `cpanfile` or `cpanfile.snapshot` file
- The current directory contains a `META.json` file or `META.yml` file
- The current directory contains a `.perl-version` file
- The current directory contains a `.pl`, `.pm` or

Options

Option	Default
`format`	`'via [$symbol($version )]($style)'`
`version_format`	`'v${raw}'`
`symbol`	`'🐪 '`
`detect_extensions`	`['pl', 'pm', 'pod']`
`detect_files`	`['Makefile.PL', 'Build.PL', 'cpanfile', 'cpanfile.snapshot', 'META.json', 'META.yml', '.perl-version']`
`detect_folders`	`[]`

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`style`	`'bold 149'`
`disabled`	`false`

Variables

Variable	Example	Description
version	`v5.26.1`	The version of `perl`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

Example

```
# ~/.config/starship.toml

[perl]
format = 'via [🐪 $version]($style) '
```

PHP

The `php` module shows the currently installed version of PHP. By default the module will be shown if any of the following conditions are met:

- The current directory contains a `composer.json` file

Options

Option	Default	D
`format`	`'via [$symbol($version)]($style)'`	T fo m
`version_format`	`'v${raw}'`	T fo A va
`symbol`	`'🐘 '`	T us d th o
`detect_extensions`	`['php']`	W ex sl tr m
`detect_files`	`['composer.json', '.php-version']`	W fi sl tr m
`detect_folders`	`[]`	W fo sl m

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`style`	`'147 bold'`	fc m
`disabled`	`false`	D th m

Variables

Variable	Example	Description
version	`v7.3.8`	The version of `php`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[php]
format = 'via [🔹 $version](147 bold) '
```

Pijul Channel

The `pijul_channel` module shows the active channel of the repo in your current directory.

Options

Option	Default	De
`symbol`	`'🅒 '`	Th us the ch na rep cu dir
`style`	`'bold purple'`	Th for mo
`format`	`'on [$symbol$channel ]($style) '`	Th for mo
`truncation_length`	`2^63 - 1`	Tru the ch na N gra
`truncation_symbol`	`'…'`	Th us inc bra na tru
`disabled`	`true`	Dis the mo

Pulumi

> **TIP**
>
> By default the Pulumi version is not shown, since it takes an order of magnitude longer to load then most plugins (~70ms). If you still want to enable it, follow the example shown below.

By default the module will be shown if any of the following conditions are met:

- The current directory contains either `Pulumi.yaml` or `Pulumi.yml`
- A parent directory contains either `Pulumi.yaml` or `Pulumi.yml` unless `search_upwards` is set to `false`

Options

Option	Default
`format`	`'via [$symbol($username@)$stack]($style) '`
`version_format`	`'v${raw}'`

`symbol`	'📦 '
`style`	'bold 5'
`search_upwards`	`true`
`disabled`	`false`

Variables

Variable	Example	Description
version	`v0.12.24`	The version of `pulumi`
stack	`dev`	The current Pulumi stack
username	`alice`	The current Pulumi username
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

Example

With Pulumi Version

```
# ~/.config/starship.toml

[pulumi]
format = '[ ($version )$stack]($style) '
```

Without Pulumi version

```
# ~/.config/starship.toml
[pulumi]
symbol = ' '
format = '[$symbol$stack]($style) '
```

PureScript

The `purescript` module shows the currently installed version of PureScript version. By default the module will be shown if any of the following conditions are met:

- The current directory contains a `spago.dhall` file
- The current directory contains a file with the `.purs` extension

Options

Option	Default	D
`format`	`'via [$symbol($versio`	Th for

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`version_format`	`'v${raw}'`	Th for Av var ra ma mi pa
`symbol`	`'<=> '`	Th use dis the of Pu
`detect_extensions`	`['purs']`	Wh ext sho trig mo
`detect_files`	`['spago.dhall']`	Wh file sho trig mo
`detect_folders`	`[]`	Wh fol sho trig mo
`style`	`'bold white'`	Th the
`disabled`	`false`	Dis pu t

Variables

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

version	`0.13.5`	The version of `purescript`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[purescript]
format = 'via [$symbol$version](bold white)'
```

Python

The `python` module shows the currently installed version of Python and the current Python virtual environment if one is activated.

If `pyenv_version_name` is set to `true`, it will display the pyenv version name. Otherwise, it will display the version number from `python --version`.

By default the module will be shown if any of the following conditions are met:

- The current directory contains a `.python-version` file
- The current directory contains a `Pipfile` file

file
- The current directory contains a `requirements.txt` file
- The current directory contains a `setup.py` file
- The current directory contains a `tox.ini` file
- The current directory contains a file with the `.py` extension.
- A virtual environment is currently activated

Options

Option	Default
`format`	`'via [${symbol}${pyenv_pref }(${version} )(\($virtualenv\) )](${style)'`
`version_format`	`'v${raw}'`
`symbol`	`'🅒 '`
`style`	`'yellow bold'`

`pyenv_version_name`	`false`
`pyenv_prefix`	`'pyenv'`
`python_binary`	`['python', 'python3', 'python2']`
`detect_extensions`	`['py']`
`detect_files`	`['.python-version', 'Pipfile', '__init__.py', 'pyproject.toml', 'requirements.txt', 'setup.py', 'tox.ini']`
`detect_folders`	`[]`
`disabled`	`false`

The `python_binary` variable accepts either a string or a list of strings. Starship will try executing each binary until it gets a result. Note you can only change the binary that Starship executes to get the version of Python not the arguments that are used.

The default values and order for `python_binary` was chosen to first identify the Python version in a virtualenv/conda environments (which currently still add a `python` , no matter if it points to `python3` or `python2`). This has the side effect that if you still have a system Python 2 installed, it may be picked up before any Python 3 (at least on Linux Distros that always symlink `/usr/bin/python` to Python 2). If you do not work with Python 2 anymore but cannot remove the system Python 2, changing this to `'python3'` will hide any Python version 2, see example below.

Variables

Variable	Example	Description
version	`'v3.8.1'`	The version of `python`
symbol	`'🅒 '`	Mirrors the value of option `symbol`
style	`'yellow bold'`	Mirrors the value of option `style`

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

pyenv_prefix	`'pyenv '`	Mirrors the value of option `pyenv_prefix`
virtualenv	`'venv'`	The current `virtualenv` name

Example

```
# ~/.config/starship.toml

[python]
symbol = '👾 '
pyenv_version_name = true
```

```
# ~/.config/starship.toml

[python]
# Only use the `python3` binary to get the ve
python_binary = 'python3'
```

```
# ~/.config/starship.toml

[python]
# Don't trigger for files with the py extensi
detect_extensions = []
```

```
# ~/.config/starship.toml

[python]
# Display the version of python from inside a
#
# Note this will only work when the venv is i
# work in the directory that contains the ven
```

R

The `rlang` module shows the currently installed version of R. The module will be shown if any of the following conditions are met:

- The current directory contains a file with the `.R` extension.
- The current directory contains a file with the `.Rd` extension.
- The current directory contains a file with the `.Rmd` extension.
- The current directory contains a file with the `.Rproj` extension.
- The current directory contains a file with the `.Rsx` extension.
- The current directory contains a `.Rprofile` file
- The current directory contains a `.Rproj.user` folder

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for Av va

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`symbol`	`'📐'`	A f str rep the of
`style`	`'blue bold'`	Th the
`detect_extensions`	`['R', 'Rd', 'Rmd', 'Rproj', 'Rsx']`	Wh ext sh tri mo
`detect_files`	`['.Rprofile']`	Wh file sh tri mo
`detect_folders`	`['.Rproj.user']`	Wh fol sh tri mo
`disabled`	`false`	Dis r

Variables

Variable	Example	Description
version	`v4.0.5`	The version of `R`
symbol		Mirrors the value of option `symbol`
style	`'blue bold'`	Mirrors the value of option `style`

Example

```
# ~/.config/starship.toml

[rlang]
format = 'with [📐 $version](blue bold) '
```

Raku

The `raku` module shows the currently installed version of Raku. By default the module will be shown if any of the following conditions are met:

- The current directory contains a `META6.json` file
- The current directory contains a `.p6`, `.pm6`, `.raku`, `.rakumod` or `.pod6`

Options

Option	Default
`format`	`'via [$symbol($version-$vm_v n )]($style)'`
`version_format`	`'v${raw}'`

`symbol`	`'🦋 '`
`detect_extensions`	`['p6', 'pm6', 'pod6', 'raku', 'rakumod']`
`detect_files`	`['META6.json']`
`detect_folders`	`[]`
`style`	`'bold 149'`
`disabled`	`false`

Variables

Variable	Example	Description
version	`v6.d`	The version of `raku`
vm_version	`moar`	The version of VM `raku` is built on

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

Example

```
# ~/.config/starship.toml

[raku]
format = 'via [🦪 $version]($style) '
```

Red

By default the `red` module shows the currently installed version of Red. The module will be shown if any of the following conditions are met:

- The current directory contains a file with `.red` or `.reds` extension

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for A va r ma

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

		p
`symbol`	`'▲ '`	A f str rep the of
`detect_extensions`	`['red']`	Wh ext sh tri mo
`detect_files`	`[]`	Wh file sh tri mo
`detect_folders`	`[]`	Wh fol sh tri mo
`style`	`'red bold'`	Th the
`disabled`	`false`	Dis r mo

Variables

Variable	Example	Description
version	`v2.5.1`	The version of `red`
symbol		Mirrors the value of option `symbol`

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[red]
symbol = '🔴 '
```

Ruby

By default the `ruby` module shows the currently installed version of Ruby. The module will be shown if any of the following conditions are met:

- The current directory contains a `Gemfile` file
- The current directory contains a `.ruby-version` file
- The current directory contains a `.rb` file
- The environment variables `RUBY_VERSION` or `RBENV_VERSION` are set

Starship gets the current Ruby version by running `ruby -v`.

Options

Option	Default	De

`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for Av va ra ma mi pa
`symbol`	`'💎 '`	A f str rep the of
`detect_extensions`	`['rb']`	Wh ext sh tri mo
`detect_files`	`['Gemfile', '.ruby-version']`	Wh file sh tri mo
`detect_folders`	`[]`	Wh fol sh tri mo
`detect_variables`	`['RUBY_VERSION', 'RBENV_VERSION']`	Wh en va sh tr mo

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`style`	`'bold red'`	Th the
`disabled`	`false`	Dis r mo

Variables

Variable	Example	Description
version	`v2.5.1`	The version of `ruby`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[ruby]
symbol = '🔺 '
```

Rust

By default the `rust` module shows the currently installed version of Rust. The module will be shown if any of the following conditions are met:

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

extension

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for Av var ra ma mi pa
`symbol`	`'🦀 '`	A f str rep the of
`detect_extensions`	`['rs']`	Wh ext sh tri mo
`detect_files`	`['Cargo.toml']`	Wh file sh tri mo
`detect_folders`	`[]`	Wh fo sh tri mo

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`style`	`'bold red'`	Th the
`disabled`	`false`	Dis r mo

Variables

Variable	Example	Description
version	`v1.43.0-nightly`	The version of `rustc`
numver	`1.51.0`	The numeric component of the `rustc` version
toolchain	`beta`	The toolchain version
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[rust]
format = 'via [⚙ $version](red bold)'
```

Scala

The `scala` module shows the currently installed version of Scala. By default the module will be shown if any of the following conditions are met:

- The current directory contains a `build.sbt`, `.scalaenv` or `.sbtenv` file
- The current directory contains a file with the `.scala` or `.sbt` extension
- The current directory contains a directory named `.metals`

Options

Option	Default	Descr
`format`	`'via [${symbol} (${version} )]($style)'`	The fo for the modu
`version_format`	`'v${raw}'`	The ve forma Availa vars a `raw` `majo` `mino` `patc`
`detect_extensions`	`['sbt', 'scala']`	Which extens shoul trigg modu

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`detect_files`	`['.scalaenv', '.sbtenv', 'build.sbt']`	Which filenar should trigge modu
`detect_folders`	`['.metals']`	Which folder should trigge modu
`symbol`	`'🅢 '`	A form string repres the sy of Sca
`style`	`'red dimmed'`	The st the m
`disabled`	`false`	Disabl `scal` modu

Variables

Variable	Example	Description
version	`2.13.5`	The version of `scala`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[scala]
symbol = '🌟 '
```

Shell

The `shell` module shows an indicator for currently used shell.

> **TIP**
>
> This module is disabled by default. To enable it, set `disabled` to `false` in your configuration file.

Options

Option	Default	
`bash_indicator`	`'bsh'`	A f rep
`fish_indicator`	`'fsh'`	A f rep
`zsh_indicator`	`'zsh'`	A f rep
`powershell_indicator`	`'psh'`	A rep

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`pwsh_indicator`		A r rep de val p r .
`ion_indicator`	`'ion'`	A f rep
`elvish_indicator`	`'esh'`	A f rep
`tcsh_indicator`	`'tsh'`	A f rep
`xonsh_indicator`	`'xsh'`	A f rep
`cmd_indicator`	`'cmd'`	A f rep
`nu_indicator`	`'nu'`	A f rep
`unknown_indicator`	`''`	Th dis is u
`format`	`'[$indicator ]($style) '`	Th mo
`style`	`'white bold'`	Th mo
`disabled`	`true`	Dis mo

Variables

Variable	Default	Description

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

indicator		Mirrors the value of `indicator` for currently used shell.
style*		Mirrors the value of option `style`.

*: This variable can only be used as a part of a style string

Examples

```
# ~/.config/starship.toml

[shell]
fish_indicator = '  '
powershell_indicator = '>_'
unknown_indicator = 'mystery shell'
style = 'cyan bold'
disabled = false
```

SHLVL

The `shlvl` module shows the current `SHLVL` ('shell level') environment variable, if it is set to a number and meets or exceeds the specified threshold.

Options

Option	Default	Descri
`threshold`	`2`	Displa thresh

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`format`	`'[$symbol$shlvl](($style) '`	The for for the module
`symbol`	`'↕️ '`	The syr used to represe the `SHLVL`
`repeat`	`false`	Causes `symbo` be repe by the current `SHLVL` amoun
`repeat_offset`	`0`	Decrem numbe times `symbo` repeate the offs value
`style`	`'bold yellow'`	The sty for the module
`disabled`	`true`	Disable the `sh` module

Variables

Variable	Example	Description
shlvl	3	The current value of `SHLVL`

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[shlvl]
disabled = false
format = '$shlvl level(s) down'
threshold = 3
```

Using `repeat` and `repeat_offset` along with `character` module, one can get prompt like `❯❯❯` where last character is colored appropriately for return status code and preceeding characters are provided by `shlvl` .

```
# ~/.config/starship.toml

[shlvl]
disabled = false
format = '[$symbol$shlvl]($style)'
repeat = true
symbol = '❯'
repeat_offset = 1
threshold = 0
```

Singularity

The `singularity` module shows the current Singularity image, if inside a container and `$SINGULARITY_NAME` is set.

Options

Option	Default	Description
`format`	`'[$symbol\[$env\]] ($style) '`	The format for the module.
`symbol`	`''`	A format string displayed before the image name.
`style`	`'bold dimmed blue'`	The style for the module.
`disabled`	`false`	Disables the `singularity` module.

Variables

Variable	Example	Description
env	`centos.img`	The current Singularity image
symbol		Mirrors the value of option `symbol`

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[singularity]
format = '[📦 \[$env\]]($style) '
```

Solidity

The `solidity` module shows the currently installed version of Solidity The module will be shown if any of the following conditions are met:

- The current directory contains a file with the `.sol` extension

Options

Option	Default
`format`	`'via [$symbol($version)]($style)'`
`version_format`	`'v${major}.${minor}.${`

`symbol`	`'S '`
`compiler	['solc']
`detect_extensions`	`['sol']`
`detect_files`	`[]`
`detect_folders`	`[]`
`style`	`'bold blue'`
`disabled`	`false`

Variables

Variable	Example	Description
version	`v0.8.1`	The version of `solidity`

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml
[solidity]
format = "via [S $version](blue bold)"
```

Spack

The `spack` module shows the current Spack environment, if `$SPACK_ENV` is set.

Options

Option	Default

`truncation_length`	`1`
`symbol`	`'🅢 '`
`style`	`'bold blue'`
`format`	`'via [$symbol$environment](($style) '`
`disabled`	`false`

Variables

Variable	Example	Description
environment	`astronauts`	The current spack environment
symbol		Mirrors the value of option `symbol`

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[spack]
format = '[$symbol$environment](dimmed blue)
```

Status

The `status` module displays the exit code of the previous command. If $success_symbol is empty (default), the module will be shown only if the exit code is not `0`. The status code will cast to a signed 32-bit integer.

> **TIP**
>
> This module is disabled by default. To enable it, set `disabled` to `false` in your configuration file.

Options

Option	

`format`	`'[$symbol$stat`
`symbol`	`'❌'`
`success_symbol`	`''`
`not_executable_symbol`	`'🚫'`
`not_found_symbol`	`'🔍'`
`sigint_symbol`	`'🧱'`
`signal_symbol`	`'⚡'`
`style`	`'bold red'`
`recognize_signal_code`	`true`

`map_symbol`	`false`
`pipestatus`	`false`
`pipestatus_separator`	`|`
`pipestatus_format`	`'\[$pipestatus\] [$symbol$common_msg]($style)'`
`pipestatus_segment_format`	
`disabled`	`true`

Variables

status	`127`	The exit code of the last commar
hex_status	`0x7F`	The exit code of the last commar in hex
int	`127`	The exit code of the last commar
common_meaning	`ERROR`	Meaning of the code if not a signal
signal_number	`9`	Signal number corresponding t the exit code, on if signalled
signal_name	`KILL`	Name of the signal corresponding t the exit code, on if signalled
maybe_int	`7`	Contains the exi code number when no meanin has been found
pipestatus		Rendering of in pipeline programs' exit codes, this is onl available in pipestatus_form
symbol		Mirrors the valu of option `symbol`
style*		Mirrors the valu of option `style`

Example

```
# ~/.config/starship.toml

[status]
style = 'bg:blue'
symbol = '🔴 '
success_symbol = '🟢 SUCCESS'
format = '[\[$symbol$common_meaning$signal_nai
map_symbol = true
disabled = false
```

Sudo

The `sudo` module displays if sudo credentials are currently cached. The module will only be shown if credentials are cached.

> **TIP**
>
> This module is disabled by default. To enable it, set `disabled` to `false` in your configuration file.

Options

Option	Default	Description
`format`	`'[as $symbol] ($style)'`	The format of the module

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`symbol`	`'🧙 '`	The symbol displayed when credentials are cached
`style`	`'bold blue'`	The style for the module.
`allow_windows`	`false`	Since windows has no default sudo, default is disabled.
`disabled`	`true`	Disables the `sudo` module.

Variables

Variable	Example	Description
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[sudo]
```

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

```
disabled = false

# On windows
# $HOME\.starship\config.toml

[sudo]
allow_windows = true
disabled = false
```

Swift

By default the `swift` module shows the currently installed version of Swift. The module will be shown if any of the following conditions are met:

- The current directory contains a `Package.swift` file
- The current directory contains a file with the `.swift` extension

Options

Option	Default	
`format`	`'via [$symbol($version )]($style)'`	

Home

Guide

Advanced Installation

Configuration

Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`version_format`	`'v${raw}'`
`symbol`	`' '`
`detect_extensions`	`['swift']`
`detect_files`	`['Package.swift']`
`detect_folders`	`[]`
`style`	`'bold 202'`
`disabled`	`false`

Variables

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

version	`v5.2.4`	The version of `swift`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[swift]
format = 'via [🏎  $version](red bold)'
```

Terraform

The `terraform` module shows the currently selected Terraform workspace and version.

> **TIP**
>
> By default the Terraform version is not shown, since this is slow for current versions of Terraform when a lot of plugins are in use. If you still want to enable it, follow the example shown below.

By default the module will be shown if any of the following conditions are met:

`.tfplan` or `.tfstate` extensions

Options

Option	Default
`format`	`'via [$symbol$workspace ]($style) '`
`version_format`	`'v${raw}'`
`symbol`	`'💠'`
`detect_extensions`	`['tf', 'tfplan', 'tfstate']`
`detect_files`	`[]`
`detect_folders`	`['.terraform']`

`style`	`'bold 105'`	
`disabled`	`false`	

Variables

Variable	Example	Description
version	`v0.12.24`	The version of `terraform`
workspace	`default`	The current Terraform workspace
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

With Terraform Version

```
# ~/.config/starship.toml

[terraform]
format = '[🏎💨 $version$workspace]($style) '
```

Without Terraform version

```
# ~/.config/starship.toml

[terraform]
format = '[🏎💨 $workspace](($style) '
```

Time

The `time` module shows the current **local** time. The `format` configuration value is used by the `chrono` crate to control how the time is displayed. Take a look at the chrono strftime docs to see what options are available.

> **TIP**
>
> This module is disabled by default. To enable it, set `disabled` to `false` in your configuration file.

Options

Option	Default	Description
`format`	`'at [$time]($style) '`	The format string for the module.
`use_12hr`	`false`	Enables 12 hour formatting

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`time_format`	see below	The chrono format string used to format the time.
`style`	`'bold yellow'`	The style for the module time
`utc_time_offset`	`'local'`	Sets the UTC offset to use. Range from -24 < x < 24. Allows floats to accommodate 30/45 minute timezone offsets.
`disabled`	`true`	Disables the `time` module.
`time_range`	`'-'`	Sets the time range during which the module will be shown. Times must be specified in 24-hours format

If `use_12hr` is `true`, then `time_format` defaults to `'%r'`. Otherwise, it defaults to `'%T'`. Manually setting `time_format` will override the `use_12hr` setting.

Variables

time	`13:08:10`	The current time.
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[time]
disabled = false
format = '🕙[\[ $time \]]($style) '
time_format = '%T'
utc_time_offset = '-5'
time_range = '10:00:00-14:00:00'
```

Typst

The `typst` module shows the current installed version of Typst used in a project.

By default, the module will be shown if any of the following conditions are met:

- The current directory contains a `template.typ` file
- The current directory contains any `*.typ` file

Options

Option	Default	D

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`format`	`'via [$symbol($version )]($style)'`	Fo m
`version_format`	`'v${raw}'`	Th fo A va
`symbol`	`'t '`	A sy re th of
`style`	`'bold #0093A7'`	Th th
`detect_extensions`	`['.typ']`	W ex sh tr m
`detect_files`	`['template.typ']`	W fi sh tr m
`detect_folders`	`[]`	W fo sh tr m
`disabled`	`false`	Di m

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

Variables

Variable	Example	Description
version	`v0.9.0`	The version of `typst`, alias for typst_version
typst_version	`default`	The current Typst version
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Username

The `username` module shows active user's username. The module will be shown if any of the following conditions are met:

- The current user is root/admin
- The current user isn't the same as the one that is logged in
- The user is currently connected as an SSH session
- The variable `show_always` is set to true

> **TIP**
>
> SSH connection is detected by checking

does not set up these variables, one workaround is to set one of them with a dummy value.

Options

Option	Default	Description
`style_root`	`'bold red'`	The style used when the user is root/admin.
`style_user`	`'bold yellow'`	The style used for non-root users.
`format`	`'[$user]($style) in '`	The format for the module.
`show_always`	`false`	Always shows the `username` module.
`disabled`	`false`	Disables the `username` module.

Variables

Variable	Example	Description

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`style`	`'red bold'`	Mirrors the value of option `style_root` when root is logged in and `style_user` otherwise.
`user`	`'matchai'`	The currently logged-in user ID.

Example

```
# ~/.config/starship.toml

[username]
style_user = 'white bold'
style_root = 'black bold'
format = 'user: [$user]($style) '
disabled = false
show_always = true
```

Vagrant

The `vagrant` module shows the currently installed version of Vagrant. By default the module will be shown if any of the following conditions are met:

- The current directory contains a `Vagrantfile` file

Options

Option	Default	De

`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for Av var ra ma mi pa
`symbol`	`'🅒 '`	A f str rep the of
`detect_extensions`	`[]`	Wh ext sh tri mo
`detect_files`	`['Vagrantfile']`	Wh file sh tri mo
`detect_folders`	`[]`	Wh fol sh tri mo
`style`	`'cyan bold'`	Th the
`disabled`	`false`	Dis mo

Variables

Variable	Example	Description
version	`Vagrant 2.2.10`	The version of `Vagrant`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml

[vagrant]
format = 'via [¥ $version](bold white) '
```

V

The `vlang` module shows you your currently installed version of V. By default the module will be shown if any of the following conditions are met:

- The current directory contains a file with `.v` extension
- The current directory contains a `v.mod`, `vpkg.json` or `.vpkg-lock.json` file

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for Av var ra ma mi pa
`symbol`	`'V '`	A f str rep the of
`detect_extensions`	`['v']`	Wh ext sh tri mo
`detect_files`	`['v.mod', 'vpkg.json', '.vpkg-lock.json' ]`	Wh file sh tri mo
`detect_folders`	`[]`	Wh fol sh tri m

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`style`	`'blue bold'`	Th the
`disabled`	`false`	Dis v mo

Variables

Variable	Example	Description
version	`v0.2`	The version of `v`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

Example

```
# ~/.config/starship.toml
[vlang]
format = 'via [V $version](blue bold) '
```

VCSH

The `vcsh` module displays the current active VCSH repository. The module will be shown only if a repository is currently in use.

Options

Option	Default	Description

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`symbol`	`''`	The symbol used before displaying the repository name.
`style`	`'bold yellow'`	The style for the module.
`format`	`'vcsh [$symbol$repo]($style) '`	The format for the module.
`disabled`	`false`	Disables the `vcsh` module.

Variables

Variable	Example	Description
repo	`dotfiles` if in a VCSH repo named dotfiles	The active repository name
symbol		Mirrors the value of option `symbol`
style*	`black bold dimmed`	Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
[vcsh]
format = '[V $repo](bold blue) '
```

Zig

By default the `zig` module shows the currently installed version of Zig. The module will be shown if any of the following conditions are met:

- The current directory contains a `.zig` file

Options

Option	Default	De
`format`	`'via [$symbol($version )]($style)'`	Th for mo
`version_format`	`'v${raw}'`	Th for Av var ra ma mi pa
`symbol`	`'↯ '`	Th us dis the of
`style`	`'bold yellow'`	Th fo mo

Home
Guide
Advanced Installation
Configuration
Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

`disabled`	`false`	Dis the mo
`detect_extensions`	`['zig']`	Wh ext sh tri mo
`detect_files`	`[]`	Wh file sh tri mo
`detect_folders`	`[]`	Wh fol sh tri mo

Variables

Variable	Example	Description
version	`v0.6.0`	The version of `zig`
symbol		Mirrors the value of option `symbol`
style*		Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Example

```
# ~/.config/starship.toml
```

Home

Guide

Advanced Installation

Configuration

Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

symbol = "⚡"

Custom commands

The `custom` modules show the output of some arbitrary commands.

These modules will be shown if any of the following conditions are met:

- The current directory contains a file whose name is in `detect_files`
- The current directory contains a directory whose name is in `detect_folders`
- The current directory contains a file whose extension is in `detect_extensions`
- The `when` command returns 0
- The current Operating System (std::env::consts::OS) matches with `os` field if defined.

> **TIP**
>
> Multiple custom modules can be defined by using a `.`.

> **TIP**
>
> The order in which custom modules are shown can be individually set by including `${custom.foo}` in the top level `format` (as it includes a dot, you need to use `${...}`). By default, the `custom` module will simply show all custom modules in the order they were

TIP

Issue #1252 contains examples of custom modules. If you have an interesting example not covered there, feel free to share it there!

Command output is printed unescaped to the prompt

Whatever output the command generates is printed unmodified in the prompt. This means if the output contains special sequences that are interpreted by your shell they will be expanded when displayed. These special sequences are shell specific, e.g. you can write a command module that writes bash sequences, e.g. `\h`, but this module will not work in a fish or zsh shell.

Format strings can also contain shell specific prompt sequences, e.g. Bash, Zsh.

Options

Option	Default	
`command`	`''`	T w sl T h

`when`	`false`	E va q sl u: c th c th b c a
`require_repo`	`false`	If m b c re o n d ir o
`shell`		S
`description`	`'<custom module>'`	T th sl ru e
`detect_files`	`[]`	T b w fc
`detect_folders`	`[]`	T th w fc

`detect_extensions`	`[]`	T th se w fo
`symbol`	`''`	T b th o
`style`	`'bold green'`	T m
`format`	`'[$symbol($outpu t )]($style)'`	T m
`disabled`	`false`	D
`os`		C n th b li w p
`use_stdin`		A b th w c b th st as u ir d th n th sl

		h
`ignore_timeout`	`false`	Ig t k e c m th

Variables

Variable	Description
output	The output of shell command in `shell`
symbol	Mirrors the value of option `symbol`
style*	Mirrors the value of option `style`

*: This variable can only be used as a part of a style string

Custom command shell

`shell` accepts a non-empty list of strings, where:

- The first string is the path to the shell to use to execute the command.
- Other following arguments are passed to the shell.

If unset, it will fallback to STARSHIP_SHELL and then to 'sh' on Linux, and 'cmd /C' on Windows.

The `command` will be passed in on stdin.

arguments will automatically be added: `-NoProfile -Command -` . If `shell` is not given or only contains one element and Starship detects Cmd will be used, the following argument will automatically be added: `/C` and `stdin` will be set to `false` . If `shell` is not given or only contains one element and Starship detects Nushell will be used, the following arguments will automatically be added: `-c` and `stdin` will be set to `false` . This behavior can be avoided by explicitly passing arguments to the shell, e.g.

```
shell = ['pwsh', '-Command', '-']
```

Make sure your custom shell configuration exits gracefully

If you set a custom command, make sure that the default Shell used by starship will properly execute the command with a graceful exit (via the `shell` option).

For example, PowerShell requires the `-Command` parameter to execute a one liner. Omitting this parameter might throw starship into a recursive loop where the shell might try to load a full profile environment with starship itself again and hence re-execute the custom command, getting into a never ending loop.

Parameters similar to `-NoProfile` in PowerShell are recommended for other shells as well to avoid extra loading time of a custom profile on every starship invocation.

Home

Guide

Advanced Installation

Configuration

Prompt
AWS
Azure
Battery
Buf
Bun
C
Character
CMake
COBOL / GNUCOBOL
Command Duration
Conda
Container
Crystal
Daml
Dart
Deno
Directory
Direnv
Docker Context
Dotnet
Elixir
Elm
Environment Variable
Erlang

> implemented, but it's possible that not all shells are covered. Please open an issue with shell details and starship configuration if you hit such scenario.

Example

```
# ~/.config/starship.toml

[custom.foo]
command = 'echo foo' # shows output of commanc
detect_files = ['foo'] # can specify filters l
when = ''' test "$HOME" = "$PWD" '''
format = ' transcending [$output]($style)'

[custom.time]
command = 'time /T'
detect_extensions = ['pst'] # filters *.pst fi
shell = ['pwsh.exe', '-NoProfile', '-Command',

[custom.time-as-arg]
command = 'time /T'
detect_extensions = ['pst'] # filters *.pst fi
shell = ['pwsh.exe', '-NoProfile', '-Command']
use_stdin = false
```

Edit this page on GitHub

← Advanced Installation Advanced Configuration →

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?

`<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 e
STATUS=$?

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

# Set the prompt to the output of `starship p
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

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 thes commands we set a time limit, if a command takes longer than this limit starship will stop the execution of

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.

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:

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`?

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 →

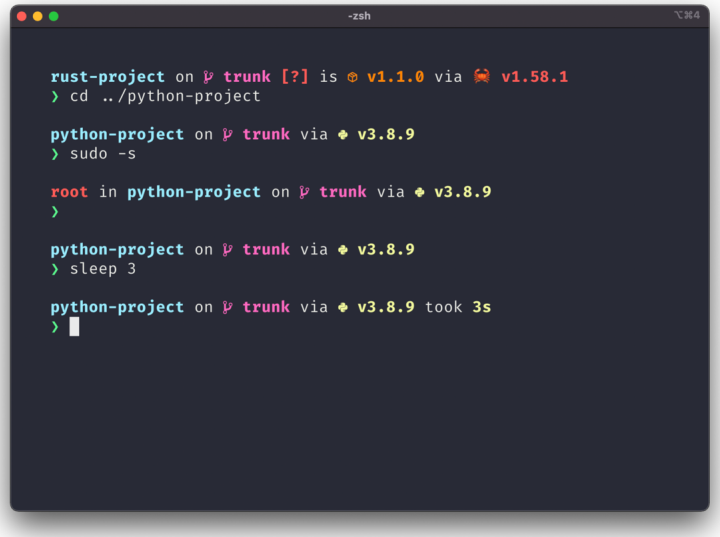
Presets

Here is a collection of community-submitted configuration presets for Starship. If you have a preset to share, please submit a PR updating this file! 😊

To get details on how to use a preset, simply click on the image.

Nerd Font Symbols

This preset changes the symbols for each module to use Nerd Font symbols.



No Nerd Fonts

This preset changes the symbols for several modules s that no Nerd Font symbols are used anywhere in the prompt.

> This preset will become the default preset in a future release of starship .

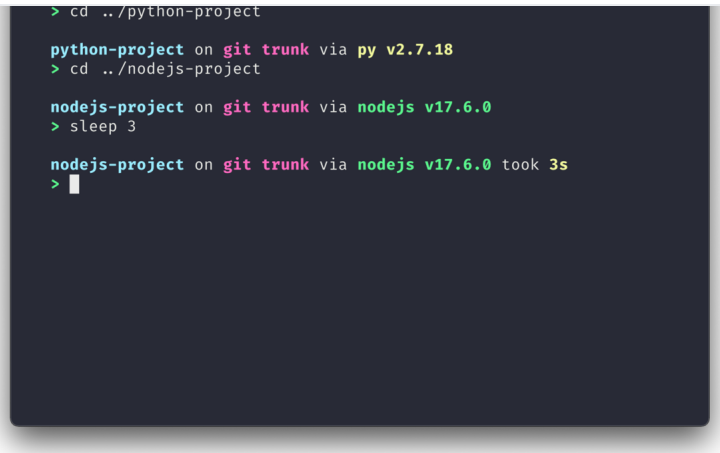
Click to view No Nerd Font preset

Bracketed Segments

This preset changes the format of all the built-in modules to show their segment in brackets instead of using the default Starship wording ("via", "on", etc.).

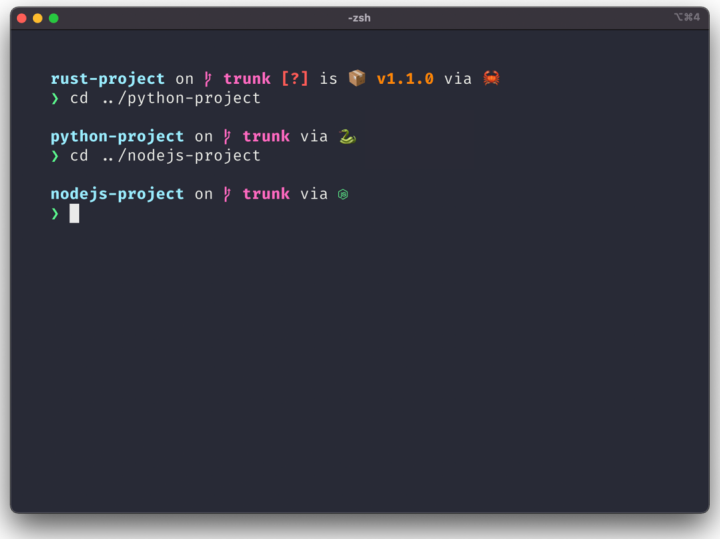
Plain Text Symbols

This preset changes the symbols for each module into plain text. Great if you don't have access to Unicode.



No Runtime Versions

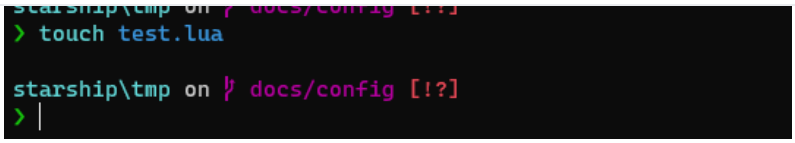
This preset hides the version of language runtimes. If you work in containers or virtualized environments, this one is for you!



No Empty Icons

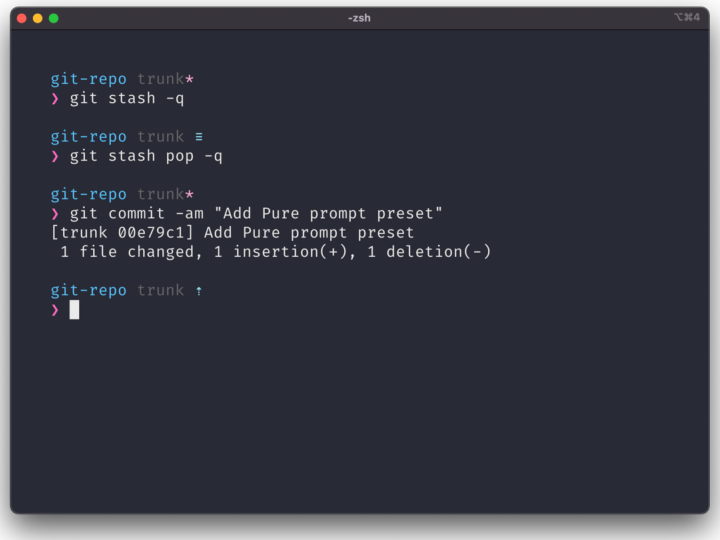
This preset does not show icons if the toolset is not found.

Home
Guide
Advanced Installation
Configuration
Advanced Configuration
Frequently Asked Questions
Presets
Nerd Font Symbols
No Nerd Fonts
Bracketed Segments
Plain Text Symbols
No Runtime Versions
No Empty Icons
Pure Prompt
Pastel Powerline
Tokyo Night
Gruvbox Rainbow



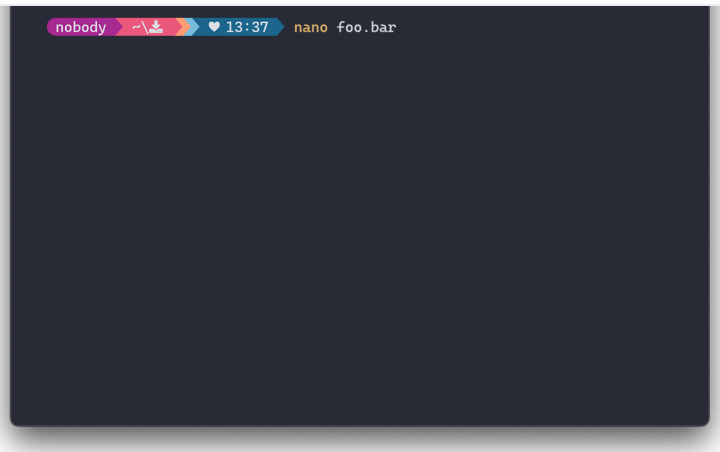
Pure Prompt

This preset emulates the look and behavior of Pure .



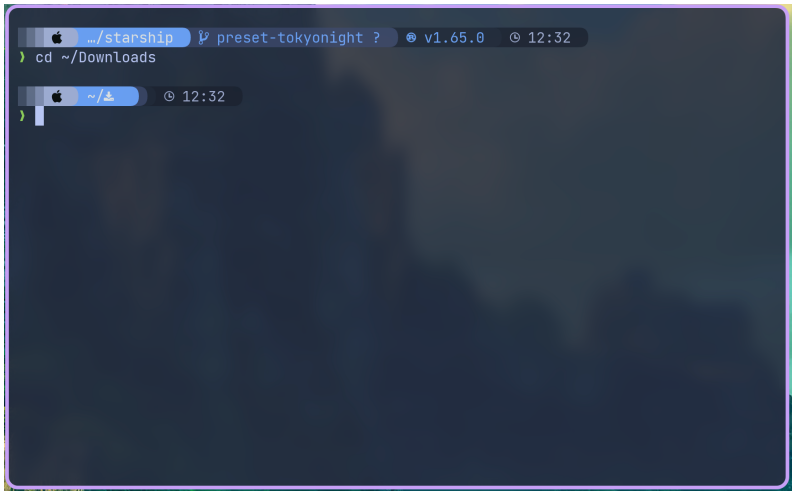
Pastel Powerline

This preset is inspired by M365Princess . It also shows how path substitution works in starship.



Tokyo Night

This preset is inspired by tokyo-night-vscode-theme .



Gruvbox Rainbow

This preset is heavily inspired by Pastel Powerline, and Tokyo Night.

Home

Guide

Advanced Installation

Configuration

Advanced Configuration

Frequently Asked Questions

Presets

Nerd Font Symbols

No Nerd Fonts

Bracketed Segments

Plain Text Symbols

No Runtime Versions

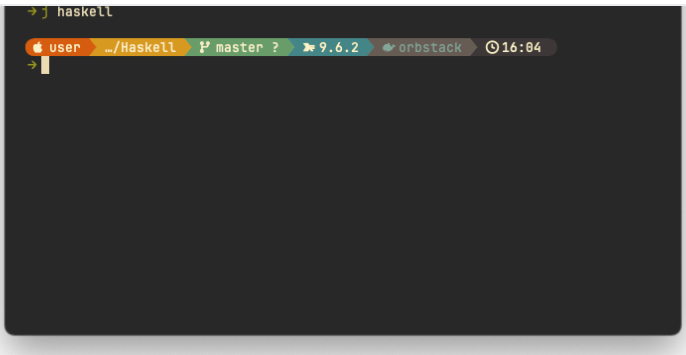
No Empty Icons

Pure Prompt

Pastel Powerline

Tokyo Night

Gruvbox Rainbow



Edit this page on GitHub

← Frequently Asked Questions

starship

CROSS-SHELL PROMPT

workflow passing | crates.io v1.17.1 | in repositories 22

discord 211 online | twitter @StarshipPrompt

#StandWithUkraine

Website · Installation · Configuration

The minimal, blazing-fast, and infinitely customizable prompt for any shell!

- **Fast:** it's fast – *really really* fast! 🚀
- **Customizable:** configure every aspect of your prompt.
- **Universal:** works on any shell, on any operating system.
- **Intelligent:** shows relevant information at a glance.
- **Feature rich:** support for all your favorite tools.
- **Easy:** quick to install – start using it in minutes.

Explore the Starship docs ▶

🚀 Installation

Prerequisites

- A Nerd Font installed and enabled in your terminal (for example, try the FiraCode Nerd Font).

Step 1. Install Starship

Select your operating system from the list below to view installation instructions:

▸ Android
▸ BSD
▾ Linux

Install the latest version for your system:

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

Alternatively, install Starship using any of the following package managers:

Distribution	Repository	Instructions
Any	crates.io	`cargo install starship --locked`
Any	conda-forge	`conda install -c conda-forge starship`
Any	Linuxbrew	`brew install starship`
Alpine Linux 3.13+	Alpine Linux Packages	`apk add starship`

Home

Guide

- 🚀 Installation
- 🤝 Contributing
- 💭 Inspired By
- ❤️ Sponsors
- 📝 License

Advanced Installation

Configuration

Advanced Configuration

Frequently Asked Questions

Presets

Arch Linux	Arch Linux Extra	`pacman -S starship`
CentOS 7+	Copr	`dnf copr enable atim/starship` `dnf install starship`
Gentoo	Gentoo Packages	`emerge app-shells/starship`
Manjaro		`pacman -S starship`
NixOS	nixpkgs	`nix-env -iA nixpkgs.starship`
openSUSE	OSS	`zypper in starship`
Void Linux	Void Linux Packages	`xbps-install -S starship`

▸ macOS
▸ Windows

Step 2. Set up your shell to use Starship

Configure your shell to initialize starship. Select yours from the list below:

▾ Bash

Add the following to the end of `~/.bashrc` :

```sh
eval "$(starship init bash)"
```

▸ Cmd
▸ Elvish
▾ Fish

```
starship init fish | source
```

▶ Ion
▶ Nushell
▶ PowerShell
▶ Tcsh
▶ Xonsh
▼ Zsh

Add the following to the end of `~/.zshrc` :

```sh
eval "$(starship init zsh)"
```

Step 3. Configure Starship

Start a new shell instance, and you should see your beautiful new shell prompt. If you're happy with the defaults, enjoy!

If you're looking to further customize Starship:

- Configuration – learn how to configure Starship to tweak your prompt to your liking
- Presets – get inspired by the pre-built configuration of others

🤝 Contributing

We are always looking for contributors of **all skill levels**! If you're looking to ease your way into the project, try out a good first issue .

If you are fluent in a non-English language, we greatly

Home

Guide

- 🚀 Installation
- 🤝 Contributing
- 💭 Inspired By
- ❤️ Sponsors
- 📝 License

Advanced Installation

Configuration

Advanced Configuration

Frequently Asked Questions

Presets

translations can be contributed on the Starship Crowdin .

If you are interested in helping contribute to starship, please take a look at our Contributing Guide . Also, feel free to drop into our Discord server and say hi. 👋

💭 Inspired By

Please check out these previous works that helped inspire the creation of starship. 🙏

- denysdovhan/spaceship-prompt – A ZSH prompt for astronauts.
- denysdovhan/robbyrussell-node – Cross-shell robbyrussell theme written in JavaScript.
- reujab/silver – A cross-shell customizable powerline-like prompt with icons.

❤️ Sponsors

Support this project by becoming a sponsor . Your name or logo will show up here with a link to your website.

Supporter Tier

- Appwrite

📝 License

Edit this page on GitHub

← Home Advanced Installation →