

[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](#)

# Configuration

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

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

All configuration for starship is done in this [TOML](#) file:

```
# Get editor completions based on the config :
"$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 currently in
success_symbol = '[→](bold green)' # The 'success' symbol
```

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

## Config File Location

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

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

[Home](#)

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

[Guide](#)

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

[Advanced Installation](#)

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

[Configuration](#)[Prompt](#)

## Logging

[AWS](#)

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:

[Azure](#)

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

[Battery](#)

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

[Buf](#)

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

[Bun](#)

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

[C](#)

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

[Character](#)[CMake](#)[COBOL / GNUCOBOL](#)[Command Duration](#)[Conda](#)[Container](#)

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

[Crystal](#)[Daml](#)

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

[Dart](#)

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

[Deno](#)[Directory](#)

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

[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](#)[Escapes](#)

## 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         |
|------------------|---------------------------|---------------|
| <code>'</code>   | literal string            | less escaping |
| <code>"</code>   | string                    | more escaping |
| <code>'''</code> | multi-line literal string | less escaping |



[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](#)

"""

multi-line  
stringmore escaping,  
newlines in  
declarations can be  
ignored

For example:

```
# literal string
format = '@\n@ '
```

```
# regular string
format = "@\\@\n@ " 
```

```
# 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



[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 = """  
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

[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 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 ( ) 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



[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](#)

`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  |
|------------------------------|-------------------|--|
| <code>format</code>          | <code>link</code> | Configure the format of the prompt.                          |
| <code>right_format</code>    | <code>''</code>   | See <a href="#">Enable Right Prompt</a>                      |
| <code>scan_timeout</code>    | <code>30</code>   | Timeout for starship to scan files (in milliseconds).        |
| <code>command_timeout</code> | <code>500</code>  | Timeout for commands executed by starship (in milliseconds). |



[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

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 I
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](#)

\$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



[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](#)

```
# 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 `AWSumne` the profile is read from the `AWSUME_PROFILE` env var and the credentials



[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](#)

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   | Help  |
|--------------------------------|---|---|
| <code>format</code>            | <code>'on[\$symbol(\$profile)(\(\$region\))(\[\$duration\])](\$style)'</code> | The format string for displaying AWS credentials. |
| <code>symbol</code>            | <code>'cloud'</code>  | The symbol used for the AWS provider.             |
| <code>region_aliases</code>    | <code>{}</code>   | The region aliases configuration.                 |
| <code>profile_aliases</code>   | <code>{}</code>   | The profile aliases configuration.                |
| <code>style</code>             | <code>'bold yellow'</code>  | The style for AWS credentials.                    |
| <code>expiration_symbol</code> | <code>'X'</code>  | The symbol used for AWS credential expiration.    |



[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<br>mc                            |
| force_display | false | If<br>evi<br>c<br>c<br>s<br>s:<br>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](#)



[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](#)

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



[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](#)

## 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  |
|-----------------------------------|--|
| <code>format</code>               | 'on<br>[\$symbol(\$subscription)<br>](\$style) ' |
| <code>symbol</code>               | ' OFF ' ' ON '                                   |
| <code>style</code>                | 'blue bold'                                      |
| <code>disabled</code>             | true   |
| <code>subscription_aliases</code> | {}   |



[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](#)

## 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

[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                           |
|--------------------|-----------------------------------|
| 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](#)

## 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.



[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

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

| Option             | Default    | Description  |
|--------------------|------------|--|
| threshold          | 10         | The upper bound of the display option.   |
| style              | 'red bold' | The style used for the display option. Defaults to <code>'bold red'</code> .   |
| charging_symbol    |            | Optional symbol displayed if the <code>display</code> option is in use. Defaults to <code>'充满'</code> . See the <code>charging_symbol</code> option.   |
| discharging_symbol |            | Optional symbol displayed if the <code>display</code> option is in use. Defaults to <code>'空'</code> . See the <code>discharging_symbol</code> option. |

## Example

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

```
[[battery.display]] # 'bold yellow' style and charge symbol
threshold = 30
style = 'bold yellow'
discharging_symbol = '⚡'
```

# when capacity is over 30%, the battery indicator changes to yellow



[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](#)

## 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  | Description  |
|--------------------------------|--|--|
| <code>format</code>            | <code>'with [\$symbol(\$version )](\$style)'</code>        | The format for buf modules.  |
| <code>version_format</code>    | <code>'v\${raw}'</code>                                    | The format for buf versions.   |
| <code>symbol</code>            | <code>'🐘 '</code>  | The symbol used to display the version of the module.                      |
| <code>detect_extensions</code> | <code>[]</code>  | Will extend the detection logic to include files with specific extensions. |
| <code>detect_files</code>      | <code>['buf.yaml', 'buf.gen.yaml', 'buf.work.yaml']</code> | Will detect buf modules based on specific files.                           |

[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_folders | []          | Will<br>follow<br>the<br>folders<br>and<br>trigger<br>the<br>macro<br>for<br>each<br>one. |
|  | style          | 'bold blue' | The<br>style<br>for<br>macro  |
|  | disabled       | false       | Disable<br>the<br>macro   |

## 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  | Description   |
|--------------------------------|--|---|
| <code>format</code>            | <code>'via<br/>[\$symbol(\$version )](\$style)'</code> | The format string for the module name.                    |
| <code>version_format</code>    | <code>'v\${raw}'</code>                                | The format string for the module version.                 |
| <code>symbol</code>            | <code>'bread'</code>                                   | The symbol representing the module.                       |
| <code>detect_extensions</code> | <code>[]</code>  | Will extend the module detection for specific extensions. |



[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']

detect\_folders

[]

style

'bold red'

disabled

false

## 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]



[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](#)

## 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   |
|--------------------------------|---|
| <code>format</code>            | <code>'via<br/>[\$symbol(\$version(-\$name))])(\$style)'</code> |
| <code>version_format</code>    | <code>'v\${raw}'</code>   |
| <code>symbol</code>            | <code>'C '</code>   |
| <code>detect_extensions</code> | <code>['c', 'h']</code>   |
| <code>detect_files</code>      | <code>[]</code>   |



[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\_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

[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](#)

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



[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_symbol`, and  
`vimcmd_visual_symbol` are only supported in  
fish due to [upstream issues with mode detection in zsh](#).

## Options

| Option                      | Default           | Help   |
|-----------------------------|-------------------|--|
| <code>format</code>         | '\$symbol'        | <a href="#">T</a><br><a href="#">st</a><br><a href="#">b</a><br><a href="#">te</a>   |
| <code>success_symbol</code> | '[>](bold green)' | <a href="#">T</a><br><a href="#">st</a><br><a href="#">b</a><br><a href="#">te</a><br><a href="#">tl</a><br><a href="#">co</a><br><a href="#">si</a> |
| <code>error_symbol</code>   | '[>](bold red)'   | <a href="#">T</a><br><a href="#">st</a><br><a href="#">b</a><br><a href="#">te</a><br><a href="#">tl</a><br><a href="#">co</a><br><a href="#">fa</a> |
| <code>vimcmd_symbol</code>  | '[<](bold green)' | <a href="#">T</a><br><a href="#">st</a><br><a href="#">b</a><br><a href="#">te</a><br><a href="#">tl</a><br><a href="#">vi</a><br><a href="#">m</a>  |



[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

'[&lt;](bold purple)'

vimcmd\_replace\_symbol

'[&lt;](bold purple)'

vimcmd\_visual\_symbol

'[&lt;](bold yellow)'

disabled

false

## Variables

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

## Examples



[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](#)

### 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`



[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                                     |
|-------------------|---|
| format            | 'via<br>[\$symbol(\$version<br>)](\$style)' |
| version_format    | 'v\${raw}'                                  |
| symbol            | '△'   |
| detect_extensions | []  |
| detect_files      | ['CMakeLists.txt'<br>,'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](#)

## 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 | Documentation     |
|--------|---------|-------------------|
| symbol | ''      | The user's choice |



## 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

Expo

|  |                   |  |  |
|--|-------------------|--|--|
|  |                   |  | the<br>of                                      |
|  | format            | 'via<br>[\$symbol(\$versio<br>n )](\$style)' | Th<br>for<br>mc                                |
|  | version_format    | 'v\${raw}'                                   | Th<br>for<br>Av<br>val<br>ri<br>me<br>m.<br>pi |
|  | style             | 'bold blue'                                  | Th<br>for<br>mc                                |
|  | detect_extensions | ['cbl', 'cob',<br>'CBL', 'COB']              | Wl<br>ext<br>sho<br>tri<br>mc                  |
|  | detect_files      | []   | Wl<br>file<br>sho<br>tri<br>mc                 |
|  | detect_folders    | []   | Wl<br>fol<br>sho<br>tri<br>mc                  |
|  | disabled          | false  | Dis<br>the<br>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  | 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 pexec-like functionality can use rcaloras's bash\_pexec framework. Simply define the arrays pexec\_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                             | Description   |
|----------------------|-------------------------------------|---|
| min_time             | 2_000                               | Shortest duration to show a command (in milliseconds) |
| show_milliseconds    | false                               | Show milliseconds addition seconds in the duration    |
| format               | 'took<br>[\$duration]<br>(\$style)' | The format for the message                            |
| style                | 'bold yellow'                       | The style for the message                             |
| disabled             | false                               | Disable command monitoring                            |
| show_notifications   | false                               | Show notifications when command completed             |
| min_time_to_notify   | 45_000                              | Shortest duration to notify (in milliseconds)         |
| notification_timeout |                                     | Duration to show notifications (in milliseconds)      |

[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](#)

If unset  
notific  
timeo  
deterr  
daeme  
all not  
daeme  
honor  
option

## 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.



[Home](#)[Guide](#)[Advanced Installation](#)

## Configuration

- Prompt
- AWS
- Azure
- Battery
- Buf
- Bun
- C
  - truncation\_length
- Character
- CMake
- COBOL / GNUCOBOL
- Command Duration
- Conda
- Container
- Crystal
- Daml
- Dart
  - symbol
- Deno
- Directory
- Direnv
  - style
- Docker Context
- Dotnet
- Elixir
  - format
- Elm
- Environment Variable
  - ignore\_base
- ESLint

modifier, you may want to run `conda config --set change_ps1 False`.

## Options

| Option      | Default  |
|-------------|--|
|             | 1  |
| symbol      | 'C'  |
| style       | 'bold green'                                   |
| format      | 'via<br>[\$symbol\$environment<br>](\$style) ' |
| ignore_base | true   |



[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                        |
|-------------|------------|------------------------------------|
| 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

[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                               |
|----------|----------------------------------|---|
| 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

[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](#)`[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                                 | Description  |
|-----------------------------|---|--|
| <code>symbol</code>         | '\$symbol'                              | The symbol to use for displaying the name of the module. |
| <code>format</code>         | 'via<br>[\$symbol(\$version)](\$style)' | The format string for displaying the module information. |
| <code>version_format</code> | 'v\${raw}'                              | The format string for displaying the raw module version. |



[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>style</p> <p>detect_extensions</p> <p>detect_files</p> <p>detect_folders</p> <p>disabled</p> | <p>'bold red'</p> <p>['cr']</p> <p>['shard.yml']</p> <p>[]</p> <p>false</p> | <p>IN<br/>for<br/>mc</p> <p>Wi<br/>ext<br/>sho<br/>tri<br/>mc</p> <p>Wi<br/>file<br/>sho<br/>tri<br/>mc</p> <p>Wi<br/>fol<br/>sho<br/>tri<br/>mc</p> <p>Dis<br/>the<br/>c<br/>mc</p> |
|---|---|--|

## 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



[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

[crystal]
format = 'via [star $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  | Description   |
|-----------------------------|--|---|
| <code>format</code>         | <code>'via [\$symbol(\$version )](\$style)'</code> | The <code>format</code> option controls the output format of the Daml version string. It uses template-like syntax where <code>\$symbol</code> is the name of the Daml module, <code>\$version</code> is the current Daml version, and <code>\$style</code> is the style of the output. |
| <code>version_format</code> | <code>'v\${raw}'</code>                            | The <code>version_format</code> option controls the output format of the raw Daml version string. It uses template-like syntax where <code>v\${raw}</code> represents the raw Daml version string.  |

[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 '          | AT<br>str<br>rep<br>the<br>of  |
|  | style             | 'bold cyan'   | The<br>the                     |
|  | detect_extensions | []            | Wl<br>ext<br>sho<br>tri<br>mc  |
|  | detect_files      | ['daml.yaml'] | Wl<br>file<br>sho<br>tri<br>mc |
|  | detect_folders    | []            | Wl<br>fol<br>sho<br>tri<br>mc  |
|  | disabled          | false         | Dis<br>di<br>mc                |

## 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  |



[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

[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  | Description   |
|-----------------------------|--|---|
| <code>format</code>         | <code>'via [\$symbol(\$version )](\$style)'</code> | The <code>format</code> option controls the output of the Dart module. It can be a string or a function. If it's a string, it will be formatted using the <code>\$symbol</code> placeholder for the symbol name and the <code>\$version</code> placeholder for the version number. The <code>\$style</code> placeholder is used to control the styling of the output. |
| <code>version_format</code> | <code>'v\${raw}'</code>                            | The <code>version_format</code> option controls the output of the Dart module when the <code>format</code> option is a function. It specifies a template string that will be used to format the version number. The <code>raw</code> placeholder is used to prevent the version number from being escaped.  |

[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](#)

|  |                   |   |  |
|--|-------------------|---|--|
|  |                   |   | path   |
|  | symbol            | ''  | A friendly string representation of the value of option symbol |
|  | detect_extensions | ['dart']  | Will extend shell-trig mode                                    |
|  | detect_files      | ['pubspec.yaml', 'pubspec.yml', 'pubspec.lock'] | Will file shell-trig mode                                      |
|  | detect_folders    | ['.dart_tool']                                  | Will folder shell-trig mode                                    |
|  | style             | 'bold blue'                                     | The theme  |
|  | disabled          | false   | Disable  |

## 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                                  | Description   |
|----------------|--|---|
| format         | 'via<br>[\$symbol(\$version )](\$style)' | The format string for the module. It can contain the \$symbol, \$version, and \$style placeholders. |
| version_format | 'v\${raw}'                               | The format string for the module's version. It can contain the \${raw} placeholder.                 |

[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](#)

symbol

'🦕'

detect\_extensions

[]

detect\_files

['deno.json',  
'deno.jsonc',  
'mod.ts',  
'mod.js',  
'deps.ts',  
'deps.js']

detect\_folders

[]

style

'green bold'

disabled

false

## 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 |
|--------|---------|
|--------|---------|



[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](#)

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

'[\$before\_root\_pa  
(\$before\_repo\_roo  
)[repo\_root]  
(\$repo\_root\_styl  
[\$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  |
|--------------------------------|--|
| <code>format</code>            | <code>'[\$symbol\$loaded/\$allo](\$style) '</code>                                       |
| <code>symbol</code>            | <code>'direnv '</code>   |
| <code>style</code>             | <code>'bold orange'</code>   |
| <code>disabled</code>          | <code>true</code>  |
| <code>detect_extensions</code> | <code>[]</code>  |
| <code>detect_files</code>      | <code>['.envrc']</code>  |
| <code>detect_folders</code>    |  [] |

[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](#)

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. |



[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](#)

|                      |                                |   |
|----------------------|--------------------------------|---|
| <code>rc_path</code> | <code>/home/test/.envrc</code> | The current rc file path.                         |
| <code>symbol</code>  |                                | Mirrors the value of option <code>symbol</code> . |
| <code>style*</code>  | <code>red bold</code>          | Mirrors the value of option <code>style</code> .  |

\*: 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  | Help |
|---------------------|--|------|
| <code>format</code> | <code>'via[\$symbol\$context](\$style)'</code> | 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](#)

|  |                   |   |                           |
|--|-------------------|---|---------------------------|
|  | symbol            | '🐳'   | In<br>be<br>the<br>co     |
|  | only_with_files   | true  | Or<br>the                 |
|  | detect_extensions | []  | Wi<br>sh<br>me<br>oi<br>s |
|  | detect_files      | ['docker-compose.yml',<br>'docker-compose.yaml',<br>'Dockerfile'] | Wi<br>sh<br>me<br>oi<br>s |
|  | detect_folders    | []  | Wi<br>sh<br>me<br>oi<br>s |
|  | style             | 'blue bold'   | The<br>me                 |
|  | disabled          | false   | Dis<br>ab<br>me           |

## 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                                      |
|-----------------------------|--|
| <code>format</code>         | 'via [\$symbol(\$version )(\$tfm)](\$style)' |
| <code>version_format</code> | 'v\${raw}'                                   |



[Home](#)

symbol

' .NET '

[Guide](#)

heuristic

true

[Advanced Installation](#)

detect\_extensions

['csproj', 'fsproj',  
'xproj'][Configuration](#)

Prompt

AWS

Azure

Battery

Buf

Bun

C

Character

CMake

COBOL / GNUCOBOL

Command Duration

Conda

detect\_files

['global.json',  
'project.json',  
'Directory.Build.props'  
'Directory.Build.targets',  
'Packages.props']

Container

detect\_folders

[]

Crystal

style

'bold blue'

Daml

Dart

Deno

Directory

disabled

false

Direnv

Docker Context

Dotnet

Elixir

Elm

Environment Variable

## Variables



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

| 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.

[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                               |
|-------------------|---|---|
| format            | 'via<br>[\$symbol(\$version \\(OTP<br>\$otp_version\\))]<br>](\$style)' | The format string for the output.         |
| version_format    | 'v\${raw}'  | The format string for the version output. |
| symbol            | '\ud83d\udcbb'  | The symbol used for the status indicator. |
| detect_extensions | []  | The list of file extensions to detect.    |
| detect_files      | ['mix.exs']   | The list of files to detect.              |
| detect_folders    | []  | The list of folders to detect.            |



[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

|          |               |                         |
|----------|---------------|-------------------------|
|          |               |                         |
| style    | 'bold purple' | The style of the prompt |
| disabled | false         | Disable the prompt      |

## 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 = '👾 '
```





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

[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  |
|--------------------------------|--|--|
| <code>format</code>            | <code>'via<br/>[\$symbol(\$version )](\$style)'</code> | The format string for the output.                  |
| <code>version_format</code>    | <code>'v\${raw}'</code>                                | The format string for the raw version output.      |
| <code>symbol</code>            | <code>'\u26bd'</code>                                  | A symbol representing the version of the language. |
| <code>detect_extensions</code> | <code>['elm']</code>                                   | File extensions to detect for Elm files.           |
| <code>detect_files</code>      | <code>['elm.json', 'elm-'</code>                       | File names to detect for Elm files.                |

[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'] | mc                           |
|  | detect_folders | ['elm-stuff']    | Wi<br>fol<br>sh<br>tri<br>mc |
|  | style          | 'cyan bold'      | Th<br>the                    |
|  | disabled       | false            | Di:<br>e<br>mc               |

## 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) '
```



[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](#)

# 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

[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](#)[Explain](#)

## 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<br>[\$env_value]<br>(\$style) " | The format for the module.   |
| description | "<env_var<br>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](#)

## Variables

| Variable  | Example   | Description                                    |
|-----------|---|--|
| env_value | Windows NT<br>(if <i>variable</i><br>would be<br>\$OS ) | The environment<br>value of option<br>variable |
| symbol    |   | Mirrors the value<br>of option<br>symbol       |
| style*    | black bold<br>dimmed                                    | Mirrors the value<br>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'
```





Default

Unknown user

[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](#)

## 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  | Description  |
|-----------------------------|--|--|
| <code>format</code>         | <code>'via<br/>[\$symbol(\$version )](\$style)'</code> | The format string for the output. The <code>\$symbol</code> placeholder is replaced by the symbol name, and the <code>\$version</code> placeholder is replaced by the Erlang version number. The <code>\$style</code> placeholder is replaced by the style defined in the <code>style</code> option. |
| <code>version_format</code> | <code>'v\${raw}'</code>                                | The format string for the raw Erlang version number. This is used when the <code>symbol</code> option is not specified.  |
| <code>symbol</code>         | <code>'( Erlang )'</code>                              | The symbol to use for the Erlang logo. The symbol is enclosed in parentheses.  |
| <code>style</code>          | <code>'bold red'</code>                                | The style to use for the output. The style is defined as a color (e.g., <code>red</code> ) and a font weight (e.g., <code>bold</code> ). The style is applied to the entire output.  |

[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

[]

detect\_files

['rebar.config'  
, 'erlang.mk']

detect\_folders

[]

disabled

false

## 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



[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](#)`[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  | Description                            |
|-----------------------------|--|--|
| <code>format</code>         | <code>'via [\$symbol(\$version )](\$style)'</code> | The format string for the module.      |
| <code>version_format</code> | <code>'v\${raw}'</code>                            | The format string for the raw version. |
| <code>symbol</code>         | <code>' Erlang '</code>                            | The symbol to use for the language.    |



[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 green' |  |
|  | detect_extensions | ['fnl']      |  |
|  | detect_files      | []           |  |
|  | detect_folders    | []           |  |
|  | disabled          | false        |  |

## 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



[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
```

```
[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                           |
|-----------------------|--------------------|---------------------------------------|
| <code>symbol</code>   | '.'                | The symbol used to fill the line.     |
| <code>style</code>    | 'bold black'       | The style for the module.             |
| <code>disabled</code> | <code>false</code> | Disables the <code>fill</code> module |

## Example

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

```
[fill]
```



[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](#)

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<br>[\$symbol\$branch<br>](\$style) ' | The<br>the<br>Use<br>'\$<br>refe<br>curr<br>nan |
| symbol            | 'ψ '                                     | The<br>use<br>bra<br>the<br>in y<br>dire        |
| style             | 'bold purple'                            | The<br>the                                      |
| truncation_length | 2^63 - 1                                 | Tru<br>Fos<br>na<br>gra                         |

[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

truncation\_symbol

'...'

disabled

true

line  
use  
a br  
was  
You  
''  
synDis  
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 = ''
```



[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](#)

## 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  |                 |
|---------------------------------|--|-----------------|
| <code>format</code>             | <code>'([+\$added](\$added_style)([-\$deleted](\$deleted_style)) )'</code> | Th<br>the       |
| <code>added_style</code>        | <code>'bold green'</code>  | Th<br>ad        |
| <code>deleted_style</code>      | <code>'bold red'</code>  | Th<br>de        |
| <code>only_nonzero_diffs</code> | <code>true</code>  | Re<br>on<br>ite |
| <code>disabled</code>           | <code>true</code>  | Dis<br>fo<br>s  |

## 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

|                |   |   |
|----------------|---|---|
| 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.

[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  |
|-----------------|--|
| format          | 'on<br>[\$symbol\$account(@\$domain)(\(\$region\))](\$style) ' |
| symbol          | 'cloud'  |
| region_aliases  | {}   |
| project_aliases | {}   |
| detect_env_vars | []   |
| style           | 'bold blue'  |
| disabled        | false  |



[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   |
|----------|-------------|---|
| 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<br>~/.config<br>/gcloud<br>/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]
```



[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](#)

### 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 = 'G '
[gcloud.region_aliases]
us-central1 = 'uc1'
asia-northeast1 = 'an1'
```

### Display account and aliased project

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

```
[gcloud]
format = 'on [${symbol$account(@$domain)($project)} ${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.



[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   |
|--------------------|---|
| always_show_remote | false   |
| format             | 'on<br>[\$symbol\$branch(:\$remote)](\$style) ' |
| symbol             | '⠇ '  |
| style              | 'bold purple'                                   |
| truncation_length  | 2^63 - 1  |
| truncation_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](#)

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 | Description       |
|--------------------|---------|-------------------|
| commit_hash_length | 7       | The length of 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

|  |                    |  | com<br>hash  |
|--|--------------------|--|--|
|  | format             | '[\\(\$hash\$ta<br>g\\)](\$style)<br>' | The f<br>for th<br>modi  |
|  | style              | 'bold green'                           | The s<br>the n   |
|  | only_detached      | true                                   | Only<br>git cc<br>hash<br>in de<br>HEAL                        |
|  | tag_disabled       | true                                   | Disab<br>show<br>info i<br>git<br>t m                          |
|  | tag_max_candidates | 0                                      | How<br>com<br>consi<br>tag d<br>The c<br>only<br>exact<br>matc |
|  | tag_symbol         | ' ↗ '                                  | Tag s<br>prefi<br>info s                                       |
|  | disabled           | false                                  | Disab<br>git<br>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 = '.tagName'
```

## 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' |



[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](#)

merge

'MERGING'

revert

'REVERTING'

cherry\_pick

'CHERRY-PICKING'

bisect

'BISECTING'

am

'AM'





## 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

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 $progress_total $percentage) $branch $files) $cherry_pick]($bold $color)'
```

## 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 |
|--------|---------|---|
|        |         |   |

[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](#)

|                    |   |  |
|--------------------|---|--|
| added_style        | 'bold green'  | In the config file, the value of option added_style is 'bold green'. |
| deleted_style      | 'bold red'  | The deleted_style option is set to 'bold red'.                       |
| only_nonzero_diffs | true  | Re-enabling starship's diffing logic.                                |
| format             | '([+\$added](\$added_style)([-\$deleted](\$deleted_style)) )' | The format option is set to a complex string.                        |
| disabled           | true  | Disabling the configuration.   |
| ignore_submodules  | false   | Ignores changes in submodules.                                       |

## 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    | 'x'                                       |



[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](#)

typechanged

""

style

'bold red'

ignore\_submodules

false

disabled

false

windows\_starship

## Variables

The following variables can be used in `format :`

| Variable     |   |
|--------------|---|
| all_status   | Shortcut<br>for <code>\$conflicted\$stashed\$delet</code>       |
| ahead_behind | Displays diverged , ahead ,  based on the current status of the |
| conflicted   | Displays conflicted when this                                   |

[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](#)

|             |  |
|-------------|--|
| untracked   | Displays untracked when there is no file in the directory. |
| stashed     | Displays stashed when a stash is present.                  |
| modified    | Displays modified when there is a file in the directory.   |
| staged      | Displays staged when a new file is added.                  |
| renamed     | Displays renamed when a rename operation is performed.     |
| deleted     | Displays deleted when a file's path is removed.            |
| typechanged | Displays typechange when a file's type changes.            |
| style*      | Mirrors the value of option style.                         |

\*: 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 |



[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

[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/sco
```



[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](#)

## 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  | Description                        |
|-----------------------------|--|------------------------------------|
| <code>format</code>         | <code>'via<br/>[\$symbol(\$version )](\$style)'</code> | The format string for the module.  |
| <code>version_format</code> | <code>'v\${raw}'</code>                                | The format string for the version. |
| <code>symbol</code>         | <code>'\${symbol}'</code>                              | A placeholder for the 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

detect\_extensions

['go']

detect\_files

['go.mod',  
'go.sum',  
'go.work',  
'glide.yaml',  
'Gopkg.yml',  
'Gopkg.lock',  
'.go-version']

detect\_folders

['Godeps']

style

'bold cyan'

not\_capable\_style

'bold red'

disabled

false

**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 | line version or 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 [${symbol} ${version}](${mod_version})'
```

## Using mod\_version

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

[golang]
format = 'via [${symbol} ${version}](${mod_version})'
```



[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](#)

## 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  |
|-----------------------|--|--|
| <code>format</code>   | <code>'via<br/>[\$symbol]<br/>(\$style)<br/>'</code> | The format for the module.                             |
| <code>symbol</code>   | <code>'λ '</code>                                    | A format string representing the symbol of guix-shell. |
| <code>style</code>    | <code>'yellow<br/>bold'</code>                       | The style for the module.                              |
| <code>disabled</code> | <code>false</code>                                   | Disables the <code>guix_shell</code> module.           |

## Variables

| Variable            | Example | Description                                     |
|---------------------|---------|---|
| <code>symbol</code> |         | Mirrors the value of option <code>symbol</code> |
| <code>style*</code> |         | Mirrors the value of option <code>style</code>  |



[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 [晌](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  | Description  |
|---------------------|--|--|
| <code>format</code> | <code>'via [\$symbol(\$version )](\$style)'</code> | The format of the output. You can use <code>\$symbol</code> to get the symbol name, <code>\$version</code> to get the version and <code>\$style</code> to get the 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

'G'

detect\_extensions

['gradle',  
'gradle.kts']

detect\_files

[]

detect\_folders

['gradle']

style

'bold bright-cyan'

disabled

false

recursive

false

[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  | 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                                  | Description                                 |
|--------|--|---|
| format | 'via<br>[\$symbol(\$version )](\$style)' | To format the output of the Haskell 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            | 'λ '                            | AT<br>str<br>rep<br>the<br>of  |
|  | detect_extensions | ['hs', 'cabal', 'hs-boot']      | Wi<br>ext<br>sho<br>tri<br>mc  |
|  | detect_files      | ['stack.yaml', 'cabal.project'] | Wi<br>file<br>sho<br>tri<br>mc |
|  | detect_folders    | []                              | Wi<br>fol<br>sho<br>tri<br>mc  |
|  | style             | 'bold purple'                   | Th<br>the                      |
|  | disabled          | false                           | Dis<br>ha<br>mc                |

## Variables

| Variable | Example  | Description |
|----------|--|-------------|
| version  | ghc_version<br>or snapshot<br>depending on<br>whether the<br>current project is<br>a Stack project |             |



[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

|  | 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 | C |
|--------|---------|---|
|        |         |   |





|                   |   |                                 |
|-------------------|---|---------------------------------|
| format            | 'via<br>[\$symbol(\$version<br>)](\$style)'   | I<br>f<br>m                     |
| version_format    | 'v\${raw}'  | T<br>f<br>A<br>V<br>I<br>I<br>I |
| detect_extensions | ['hx', 'hxml']  | V<br>e:<br>sl<br>tr<br>m        |
| detect_files      | ['project.xml',<br>'Project.xml',<br>'application.xml'<br>, 'haxelib.json',<br>'hxformat.json',<br>'.haxerc'] | V<br>fi<br>sl<br>tr<br>m        |
| detect_folders    | ['.haxelib',<br>'haxe_libraries']   | V<br>f<br>sl<br>tr<br>m         |
| symbol            | '⌘ '  | A<br>st<br>re<br>th<br>o        |
| style             | 'bold fg:202'   | T<br>th                         |
| disabled          | false   | L<br>I<br>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

[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](#)[Explain](#)

|                   |   |                                 |
|-------------------|---|---------------------------------|
| format            | 'via<br>[\$symbol(\$version<br>)](\$style)' | I<br>f<br>or                    |
| version_format    | 'v\${raw}'                                  | T<br>f<br>A<br>v<br>i<br>l<br>i |
| detect_extensions | []  | V<br>e:<br>sl<br>tr<br>n        |
| detect_files      | ['helmfile.yaml'<br>, 'Chart.yaml']         | V<br>fi<br>sl<br>tr<br>n        |
| detect_folders    | []  | V<br>fc<br>sl<br>tr<br>n        |
| symbol            | '*' '                                       | A<br>st<br>re<br>tl<br>o        |
| style             | 'bold white'                                | T<br>t!                         |
| disabled          | false                                       | D<br>n                          |

[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  | v3.1.1  | The version of <code>helm</code>                |
| symbol   |         | Mirrors the value of option <code>symbol</code> |
| style*   |         | Mirrors the value of option <code>style</code>  |

\*: 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 |
|-----------------------|---------|
| <code>ssh_only</code> | true    |



[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](#)

|  |                 |   |
|--|-----------------|---|
|  | ssh_symbol      | '🌐'                                     |
|  | trim_at         | '..'                                    |
|  | detect_env_vars | []                                      |
|  | format          | '[\$ssh_symbol\$hostname](\$style) in ' |
|  | style           | 'bold dimmed green'                     |
|  | 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

|            |          |   |
|------------|----------|---|
| hostname   | computer | The hostname or 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 = '[${ssh_symbol}](bold blue) on [$hostname](bold magenta)'
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
```



[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](#)

## 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 `.cljs` extension

## Options

| Option                         | Default   |
|--------------------------------|---|
| <code>format</code>            | <code>'via [\${symbol}(\${version})](\${style})'</code>         |
| <code>version_format</code>    | <code>'v\${raw}'</code>   |
| <code>detect_extensions</code> | <code>['java', 'class', 'gradle', 'jar', 'cljs', 'cljc']</code> |



[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

```
[ '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



[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
```

```
[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



[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](#)

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                                    |  |
|-------------------------------|--|--|
| <code>threshold</code> *      | 1  | Shows the number of jobs.                      |
| <code>symbol_threshold</code> | 1  | The threshold for the least significant digit. |
| <code>number_threshold</code> | 2  | The threshold for the count of commands.       |
| <code>format</code>           | <code>'[\$symbol\$number](\$style)'</code> | The template for the output.                   |
| <code>symbol</code>           | <code>'♦'</code>                           | The symbol for the count.                      |
| <code>style</code>            | <code>'bold blue'</code>                   | The style for the output.                      |
| <code>disabled</code>         | <code>false</code>                         | Disables the module.                           |

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

## Variables

| Variable | Example | Description                                     |
|----------|---------|---|
| number   | 1       | The number of jobs                              |
| symbol   |         | Mirrors the value of option <code>symbol</code> |
| style*   |         | Mirrors the value of option <code>style</code>  |

\*: 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`



[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](#)

## Options

| Option            | Default                                  | Description                               |
|-------------------|--|---|
| format            | 'via<br>[\$symbol(\$version )](\$style)' | The format string for the output.         |
| version_format    | 'v\${raw}'                               | The format string for the version output. |
| detect_extensions | ['jl']                                   | File extensions to detect.                |
| detect_files      | ['Project.toml',<br>'Manifest.toml']     | Files to detect.                          |
| detect_folders    | []                                       | Folders to detect.                        |
| symbol            | 'ø'                                      | A symbol to represent the current shell.  |
| style             | 'bold purple'                            | The style for the detected symbols.       |

[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<br>j<br>mc |
|----------|-------|----------------|

## 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



[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   |
|-------------------|--|---|
| format            | 'via<br>[\$symbol(\$version )](\$style)' | The format string for the prompt.                     |
| version_format    | 'v\${raw}'                               | The format string for the version part of the prompt. |
| detect_extensions | ['kt', 'kts']                            | File extensions to detect for Kotlin support.         |
| detect_files      | []                                       | File names to detect for specific language support.   |
| detect_folders    | []                                       | Folder names to detect for specific language support. |
| symbol            | 'K'                                      | The symbol used for the Kotlin language indicator.    |





## 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' |  |
|  | kotlin_binary | 'kotlin'    |  |
|  | disabled      | false       |  |
|  |               |             |  |

## 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 = '₭ '
```



[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](#)[`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.



[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

### WARNING

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

| Option                         | Default   |
|--------------------------------|---|
| <code>symbol</code>            | '*  |
| <code>format</code>            | '[\$symbol\$context(\$namespace)](\$style) in ' |
| <code>style</code>             | 'cyan bold'                                     |
| <code>context_aliases</code> * | {}  |
| <code>user_aliases</code> *    | {}  |
| <code>detect_extensions</code> | []  |



[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

[]

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 | <b>Required</b> 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](#)

|                            |  |
|----------------------------|--|
| <code>context_alias</code> | Context alias to display instead of the full context name.                               |
| <code>user_alias</code>    | User alias to display instead of the full user name.                                     |
| <code>style</code>         | The style for the module when using this context. If not set, will use module's style.   |
| <code>symbol</code>        | 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                              |
|------------------------|---------------------------------|--|
| <code>context</code>   | <code>starship-context</code>   | The current kubernetes context name      |
| <code>namespace</code> | <code>starship-namespace</code> | If set, the current kubernetes namespace |



[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](#)

|         |                  |  |
|---------|------------------|--|
| 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
```



[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](#)

## 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 Kube
context_pattern = ".*openshift.*"
style = "green"
symbol = "\u266a "
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 <code>line_break</code> 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

[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
- ESLint

## Options

| Option            | Default                                  | Description  |
|-------------------|--|--|
| format            | 'via<br>[\$symbol(\$version )](\$style)' | The format string to use for the output. This can be a simple string or a template-like expression.                |
| version_format    | 'v\${raw}'                               | The format string to use for the version output. This can be a simple string or a template-like expression.        |
| symbol            | '🌙 '                                     | The symbol to use for the version output. This can be a simple string or a template-like expression.               |
| detect_extensions | ['lua']                                  | Whether to detect extensions for the current directory. This can be a simple string or a template-like expression. |
| detect_files      | ['.lua-version']                         | Whether to detect files for the current directory. This can be a simple string or a template-like expression.      |
| detect_folders    | ['lua']                                  | Whether to detect folders for the current directory. This can be a simple string or a template-like expression.    |

**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' |  | trig<br>me                                       |
|  | lua_binary | 'lua'       |  | The<br>the                                       |
|  | disabled   | false       |  | Co<br>the<br>bir<br>Sta<br>exe<br>wt<br>ge<br>ve |
|  |            |             |  | Dis<br>l<br>mc                                   |

**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]
```



[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](#)

## 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  |
|------------------------|---|--|
| <code>threshold</code> | 75  | Hide the memory usage unless it exceeds this percentage. |
| <code>format</code>    | <code>'via \$symbol [\${ram}]\` \${swap}] (\$style)'</code> | The format for the module.                               |
| <code>symbol</code>    | <code>'🐏'</code>  | The symbol used before displaying the memory usage.      |



[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 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               | Description  |
|--------------------------------|-----------------------|--|
| <code>truncation_length</code> | <code>2^32 - 1</code> | The truncation length for the Meson project name. If set to 0, the entire name is shown. |
| <code>truncation_symbol</code> | <code>'...'</code>    | The symbol used to indicate truncation of the Meson project 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](#)

|  |          |  |                                  |
|--|----------|--|----------------------------------|
|  |          |  | pr<br>na<br>tr<br>Yo<br>'<br>syi |
|  | format   | 'via<br>[\$symbol\$project<br>](\$style) ' | The<br>for<br>mc                 |
|  | symbol   | '● '                                       | The<br>us<br>dis<br>the<br>na    |
|  | style    | 'blue bold'                                | The<br>for<br>mc                 |
|  | disabled | false                                      | Dis<br>the<br>mc                 |

## 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



[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](#)

## 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  |
|---------------------|--|
| <code>symbol</code> | ' ' ' ' ' '  |
| <code>style</code>  | 'bold purple'                                      |
| <code>format</code> | 'on<br>[\$symbol\$branch(:\$topic)<br>](\$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

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**

[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](#)

```
[hg_branch]
format = 'on [Y $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  | Description                               |
|-----------------------------|--|---|
| <code>format</code>         | <code>'via<br/>[\$symbol(\$version )](\$style)'</code> | The format string for the module output.  |
| <code>version_format</code> | <code>'v\${raw}'</code>                                | The format string for the version output. |

[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

'👑'

detect\_extensions

['nim', 'nims',  
'nimble']

detect\_files

['nim.cfg']

detect\_folders

[]

style

'bold yellow'

disabled

false

## 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<br>[\$symbol\$state<br>( \$(name<br>\\$))](\\$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 to use for the module.  |
|  | impure_msg  | 'impure'    | A format string showing when the shell is impure.                       |
|  | pure_msg    | 'pure'      | A format string showing when the shell is pure.                         |
|  | unknown_msg | ''          | A format string showing 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



[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   |
|-------------------|--|---|
| format            | 'via<br>[\$symbol(\$version )](\$style)'       | Format string for the prompt.                       |
| version_format    | 'v\${raw}'                                     | Format string for the version.                      |
| symbol            | '⟫ '   | Symbol used for the environment variable indicator. |
| detect_extensions | ['js', 'mjs',<br>'cjs', 'ts',<br>'mts', 'cts'] | File extensions to detect.                          |
| detect_files      | ['package.json',<br>.node-version']            | Files to detect.                                    |
| detect_folders    | ['node_modules']                               | Folders to detect.                                  |

**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

|                   |  |  |              |        |          |       |             |                   |            |  |  |
|-------------------|--|--|--------------|--------|----------|-------|-------------|-------------------|------------|--|--|
|                   | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">style</td><td style="padding: 5px;">'bold green'</td><td style="width: 10%; text-align: right; padding: 5px;">I<br/>t</td></tr> <tr> <td style="padding: 5px;">disabled</td><td style="padding: 5px;">false</td><td style="width: 10%; text-align: right; padding: 5px;">D<br/>i<br/>n</td></tr> <tr> <td style="padding: 5px;">not_capable_style</td><td style="padding: 5px;">'bold red'</td><td style="width: 10%; text-align: right; padding: 5px;">T<br/>t<br/>w<br/>e<br/>p<br/>p<br/>d<br/>m<br/>N<br/>v</td></tr> </table> | style  | 'bold green' | I<br>t | disabled | false | D<br>i<br>n | not_capable_style | 'bold red' | T<br>t<br>w<br>e<br>p<br>p<br>d<br>m<br>N<br>v |  |
| style             | 'bold green'   | I<br>t   |              |        |          |       |             |                   |            |  |  |
| disabled          | false  | D<br>i<br>n                                    |              |        |          |       |             |                   |            |  |  |
| not_capable_style | 'bold red'   | T<br>t<br>w<br>e<br>p<br>p<br>d<br>m<br>N<br>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

Extras

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

[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(\$view)}  
)(\(\$switch\_indicator  
) ](\$style)'

version\_format

'v\${raw}'

symbol

'⬢'

global\_switch\_indicator

' '

local\_switch\_indicator

'\*'

detect\_extensions

['opam', 'ml', 'rei']

detect\_files

['dune', 'dune-project', 'jbuild', 'jbuild', '.merlin']

[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](#)

|                             |   |
|-----------------------------|---|
| <code>detect_folders</code> | <code>[ '_opam', 'esy', 'cabal', 'stack', 'Cargo.toml', 'Cargo.lock', 'mix.exs', 'mix.exs.lock', 'yarn.lock', 'yarn', 'pnpm-lock.yaml', 'pnpm-lock.yaml.journal', 'npm-shrinkwrap.json', 'npm-shrinkwrap.json.journal', 'package-lock.json', 'package-lock.json.journal', 'Cargo.lock.journal', 'Cargo.toml.journal' ]</code> |
| <code>style</code>          | <code>'bold yellow'</code>  |
| <code>disabled</code>       | <code>false</code>  |

## Variables

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

[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                                      | Description                                     |
|-----------------------------|--|---|
| <code>format</code>         | 'via<br>[\$symbol(\$versio<br>n )](\$style)' | The<br>for<br>mc                                |
| <code>version_format</code> | 'v\${raw}'                                   | The<br>for<br>Av<br>val<br>ri<br>mi<br>m.<br>pi |
| <code>symbol</code>         | '🐫 '   | A f<br>str<br>re<br>the<br>of                   |

[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

['rego']

detect\_files

[]

detect\_folders

[]

style

'bold blue'

disabled

false

## 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



[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
```

```
[opa]
format = 'via [❶ $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  |
|---------------------|--|--|
| <code>format</code> | 'on<br>[\$symbol\$cloud(<br>\\(\$project<br>\\))](\$style) ' | The format for the module.                                     |
| <code>symbol</code> | 'cloud'  | The symbol used before displaying the current OpenStack cloud. |
| <code>style</code>  | 'bold yellow'  | The style for the module.                                      |



[Home](#)

disabled

false

Disables  
the  
openstack  
module.

[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                        |
|----------|---------|------------------------------------|
| 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\))]($style $symbol $)'
style = 'bold yellow'
symbol = '☁ '
```

## OS



The os module shows the current operating system.

[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](#)

## 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  |
|-----------------------|------------------------------------|--|
| <code>format</code>   | <code>'[\$symbol](\$style)'</code> | The format for the module.                             |
| <code>style</code>    | <code>'bold white'</code>          | The style for the module.                              |
| <code>disabled</code> | <code>true</code>                  | Disables the <code>os</code> module.                   |
| <code>symbols</code>  |                                    | 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



[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 is the default symbols table.

[os.symbols]

Alpaquita = "🔔 "

Alpine = "🏔️ "

Amazon = "😊 "

Android = "🤖 "

Arch = "⚡️ "

Artix = "⚡️ "

CentOS = "💠 "

Debian = "🕒 "

DragonFly = "🐉 "

Emscripten = "🔗 "

EndeavourOS = "🚀 "

Fedora = "🎩 "

FreeBSD = "😈 "

Garuda = "🦅 "

Gentoo = "ᴳ "

HardenedBSD = "🛡️ "

Illumos = "🐦 "

Linux = "🐧 "

Mabox = "📦 "

Macos = "🍎 "

Manjaro = "🫐 "

Mariner = "🌊 "

MidnightBSD = "🌙 "

Mint = "🌿 "

NetBSD = "🚩 "

NixOS = "❄️ "

OpenBSD = "🐟 "

OpenCloudOS = "☁️ "

openEuler = "🦉 "

openSUSE = "🦎 "

OracleLinux = "🦴 "

Pop = "🔍 "

Raspbian = "🍓 "

Redhat = "🎩 "

RedHatEnterprise = "🎩 "

Redox = "🧪 "

Solus = "⛵️ "

SUSE = "🦎 "





WINDOWS -

1

[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  |
|----------|---------------|--|
| symbol   |               | The current operating system symbol from advanced option symbols |
| name     | Arch<br>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



[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](#)`[os]`

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

`[os.symbols]`

```
Windows = "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.



[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   |
|-----------------|---|--|
| format          | 'is<br>[\$symbol\$version<br>](\$style) ' | The f<br>for th<br>modi                                      |
| symbol          | '📦 '                                      | The s<br>used<br>displ<br>the v<br>the<br>pack               |
| version_format  | 'v\${raw}'                                | The v<br>form<br>Avail<br>vars:<br>raw<br>maj<br>min<br>path |
| style           | 'bold 208'                                | The s<br>for th<br>modi                                      |
| display_private | false                                     | Enab<br>displ<br>versi<br>pack<br>mark<br>priv               |
| disabled        | false                                     | Disa<br>the<br>pac<br>modi                                   |

## 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 | v1.0.0 | The version or 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 `.pm5` file



[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  |
|-------------------|--|
| format            | 'via<br>[\$symbol(\$version<br>)](\$style)'  |
| version_format    | 'v\${raw}'   |
| symbol            | '🐘 '   |
| detect_extensions | ['pl', 'pm',<br>'pod']   |
| detect_files      | ['Makefile.PL',<br>'Build.PL',<br>'cpanfile',<br>'cpanfile.snapshot'<br>, 'META.json',<br>'META.yml', '.perl-<br>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



[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                             |
|-------------------|---|---|
| format            | 'via<br>[\$symbol(\$version<br>)](\$style)' | Template for the output format.         |
| version_format    | 'v\${raw}'                                  | Template for the version output format. |
| symbol            | '\${symbol}'                                | Placeholder for the symbol name.        |
| detect_extensions | ['php']                                     | Extensions to detect for PHP.           |
| detect_files      | ['composer.json', '.php-version']           | Files to detect for PHP.                |
| detect_folders    | []  | Folders to detect for PHP.              |

[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' | I<br>fc<br>m |
| disabled | false      | D<br>tl<br>m |

## Variables

| Variable | Example | Description                                     |
|----------|---------|---|
| version  | v7.3.8  | The version of <code>php</code>                 |
| symbol   |         | Mirrors the value of option <code>symbol</code> |
| style*   |         | Mirrors the value of option <code>style</code>  |

\*: 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.



[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  |
|-------------------|------------------------------------|--|
| symbol            | '\$'                               | The symbol used to represent the channel name in the prompt. |
| style             | 'bold purple'                      | The style for the channel name.                              |
| format            | 'on [\$symbol\$channel](\$style) ' | The format string for the channel name.                      |
| truncation_length | 2^63 - 1                           | The truncation length for the channel name.                  |
| truncation_symbol | '...'.                             | The truncation symbol for the channel name.                  |
| disabled          | true                               | Disables the channel name.                                   |

## 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](#)

## TIP

By default the Pulumi version is not shown, since it takes an order of magnitude longer to load than 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   |
|-----------------------------|---|
| <code>format</code>         | <code>'via<br/>[\$symbol(\$username@\$stack<br/>][\$style) '</code> |
| <code>version_format</code> | <code>'v\${raw}'</code>   |



[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

' '

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  |



[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

### 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                    | Help      |
|---------------------|----------------------------|-----------|
| <code>format</code> | 'via<br>[\$symbol(\$versio | Th<br>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](#)

## Variables

version\_format

'v\${raw}'

symbol

'&lt;=&gt;'

detect\_extensions

['purs']

detect\_files

['spago.dhall']

detect\_folders

[]

style

'bold white'

disabled

false

The  
for  
Av  
val  
r  
m  
m  
pThe  
us  
dis  
the  
of  
PuWi  
file  
sh  
tri  
mcWi  
fol  
sh  
tri  
mcTh  
theDi  
pi  
t

[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



[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](#)

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  |
|-----------------------------|--|
| <code>format</code>         | <code>'via<br/>[\${symbol}\${pyenv_pref}<br/>](\${version}<br/>)(\(\$virtualenv\<br/>)\])(\$style)'</code> |
| <code>version_format</code> | <code>'v\${raw}'</code>  |
| <code>symbol</code>         | <code>'🔗'</code>   |
| <code>style</code>          | <code>'yellow bold'</code>   |



[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_version_name | false  |
|  | pyenv_prefix       | 'pyenv'  |
|  | python_binary      | <pre>['python', 'python3', 'python2']</pre>  |
|  | detect_extensions  | <pre>['py']</pre>  |
|  | detect_files       | <pre>'.python-version', 'Pipfile', '__init__.py', 'pyproject.toml', 'requirements.txt', 'setup.py', 'tox.ini']</pre> |
|  | detect_folders     | <pre>[]</pre>  |
|  | disabled           | false  |



[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 `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                                     |
|----------------------|---------------|---|
| <code>version</code> | 'v3.8.1'      | The version of <code>python</code>              |
| <code>symbol</code>  | ' '           | Mirrors the value of option <code>symbol</code> |
| <code>style</code>   | 'yellow bold' | Mirrors the value of option <code>style</code>  |



[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<br>value of option<br>pyenv_prefix |
| virtualenv   | 'venv'  | The current<br>virtualenv<br>name              |

## Example

# ~/.config/starship.toml

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

# ~/.config/starship.toml

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

# ~/.config/starship.toml

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

# ~/.config/starship.toml

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



[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](#)

## 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  | Description   |
|-----------------------------|--|---|
| <code>format</code>         | <code>'via<br/>[\$symbol(\$version )](\$style)'</code> | The format string for the module output. It uses the same syntax as the <code>format</code> function in R. The <code>\$symbol</code> placeholder is replaced by the symbol name, and the <code>\$version</code> placeholder is replaced by the package version. The <code>\$style</code> placeholder is replaced by the style name. |
| <code>version_format</code> | <code>'v\${raw}'</code>                                | The format string for the package version. It uses the same syntax as the <code>format</code> function in R. The <code>v</code> prefix is added to the beginning of the version string.   |

[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            | '▲'                                | Af<br>str<br>rep<br>the<br>of  |
| style             | 'blue bold'                        | The<br>the                     |
| detect_extensions | ['R', 'Rd', 'Rmd', 'Rproj', 'Rsx'] | Wi<br>ext<br>sho<br>tri<br>mc  |
| detect_files      | '.Rprofile'                        | Wi<br>file<br>sho<br>tri<br>mc |
| detect_folders    | '.Rproj.user'                      | Wi<br>fol<br>sho<br>tri<br>mc  |
| disabled          | false                              | Dis<br>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  |



[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

[rlang]
format = 'with [triangle-down $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   |
|-----------------------------|---|
| <code>format</code>         | <code>'via<br/>[\$symbol(\$version-\$vm_version )](\$style)'</code> |
| <code>version_format</code> | <code>'v\${raw}'</code>   |



[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

' '

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                                 | Description  |
|----------------|---|--|
| format         | 'via<br>[\$symbol(\$version)](\$style)' | The format string for the red module. It uses the same syntax as the <code>format</code> option in the <code>starship.toml</code> configuration file.  |
| version_format | 'v\${raw}'                              | The format string for the red module when it is used to show the raw version number. It uses the same syntax as the <code>version_format</code> option in the <code>starship.toml</code> configuration file. |

[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](#)

|  |  |  |                   |            |   |
|--|--|--|-------------------|------------|---|
|  |  |  | symbol            | '▲'        | A friendly symbol representing the current status or configuration. |
|  |  |  | detect_extensions | ['red']    | Will extend the shell's style based on file extensions.             |
|  |  |  | detect_files      | []         | Will file shell's style based on files.                             |
|  |  |  | detect_folders    | []         | Will folder shell's style based on folders.                         |
|  |  |  | style             | 'red bold' | The style to use for the prompt.                                    |
|  |  |  | disabled          | false      | Disables the style.   |

## Variables

| Variable | Example | Description                         |
|----------|---------|-------------------------------------|
| version  | v2.5.1  | The version of starship.            |
| 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 |
|--------|---------|----|
|--------|---------|----|



**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

|  |                   | 'via<br>[\$symbol(\$versio<br>n )](\$style)' | In<br>for<br>mc                                 |
|--|-------------------|--|---|
|  | format            |  |   |
|  | version_format    | 'v\${raw}'                                   | The<br>for<br>Av<br>val<br>ri<br>mi<br>m.<br>pi |
|  | symbol            | ' '  | A f<br>str<br>rep<br>the<br>of                  |
|  | detect_extensions | ['rb']                                       | Wi<br>ext<br>sho<br>tri<br>mc                   |
|  | detect_files      | ['Gemfile',<br>.ruby-version']               | Wi<br>file<br>sho<br>tri<br>mc                  |
|  | detect_folders    | []   | Wi<br>fol<br>sho<br>tri<br>mc                   |
|  | detect_variables  | ['RUBY_VERSION'<br>,'RBENV_VERSION']         | Wi<br>en<br>val<br>sho<br>tri<br>mc             |



[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' | In the          |
| disabled | false      | Dis<br>ri<br>mc |

## 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:





extension

[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](#)

## Options

| Option            | Default                                  | Description  |
|-------------------|--|--|
| format            | 'via<br>[\$symbol(\$version )](\$style)' | The format string for the output.                        |
| version_format    | 'v\${raw}'                               | The format string for the version output.                |
| symbol            | '🦀'                                      | A friendly symbol to represent the detected environment. |
| detect_extensions | ['rs']                                   | File extensions to detect for the current context.       |
| detect_files      | ['Cargo.toml']                           | File names to detect for the current context.            |
| detect_folders    | []                                       | Folders to detect for the current context.               |

[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](#)

|          |            |                               |
|----------|------------|-------------------------------|
| style    | 'bold red' | In the style of the terminal. |
| disabled | false      | Disable the style.            |

## 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)'
```



[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](#)

## 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   | Description  |
|--------------------------------|---|--|
| <code>format</code>            | <code>'via<br/>[\${symbol}<br/>(\${version}<br/>)](\$style)'</code> | The format for the module  |
| <code>version_format</code>    | <code>'v\${raw}'</code>   | The version format. Available vars are <code>raw</code> , <code>major</code> , <code>minor</code> , <code>patch</code> . |
| <code>detect_extensions</code> | <code>['sbt',<br/>'scala']</code>                                   | Which extensions should trigger 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](#)

detect\_files

['.scalaenv',  
'.sbtenv',  
'build.sbt']VVNICR  
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



[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
```

```
[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 | Available |
|-----------------------------------|---------|-----------|
| <code>bash_indicator</code>       | 'bsh'   | Af<br>rep |
| <code>fish_indicator</code>       | 'fsh'   | Af<br>rep |
| <code>zsh_indicator</code>        | 'zsh'   | Af<br>rep |
| <code>powershell_indicator</code> | 'psh'   | A<br>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

ion\_indicator

elvish\_indicator

tcsh\_indicator

xonsh\_indicator

cmd\_indicator

nu\_indicator

unknown\_indicator

format

style

disabled

'ion'

'esh'

'tsh'

'xsh'

'cmd'

'nu'

''

'[\$indicator  
](\$style) '

'white bold'

true

AT  
rep  
de  
val  
p  
r .Af  
repAf  
repAf  
repAf  
repAf  
repAf  
repTh  
dis  
is lTh  
mcTh  
mcDis  
mc

## 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 | Description       |
|-----------|---------|-------------------|
| threshold | 2       | Display threshold |

[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<br>](\$style) ' | Indicator for the module  |
| symbol        | ' '                               | The symbol used to represent the SHLVL                                    |
| repeat        | false                             | Causes the symbol to be repeated by the current SHLVL amount              |
| repeat_offset | 0                                 | Decrements the number of times the symbol is repeated by the offset value |
| style         | 'bold yellow'                     | The style 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 preceding characters are provided by shlvl .

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

[shlvl]
disabled = false
format = '[$symbol$shlvl]($style)'
repeat = true
symbol = '>'
repeat_offset = 1
threshold = 0
```



[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](#)

# Singularity

The `singularity` module shows the current [Singularity](#) image, if inside a container and `$SINGULARITY_NAME` is set.

## Options

| Option                | Default                                     | Description                                      |
|-----------------------|---|--|
| <code>format</code>   | <code>'[\$symbol][\$env\'](\$style)'</code> | The format for the module.                       |
| <code>symbol</code>   | <code>''</code>                             | A format string displayed before the image name. |
| <code>style</code>    | <code>'bold dimmed blue'</code>             | The style for the module.                        |
| <code>disabled</code> | <code>false</code>                          | Disables the singularity module.                 |

## Variables

| Variable            | Example                 | Description                                     |
|---------------------|-------------------------|---|
| <code>env</code>    | <code>centos.img</code> | The current Singularity image                   |
| <code>symbol</code> |                         | Mirrors the value of option <code>symbol</code> |



[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   |
|-----------------------------|---|
| <code>format</code>         | <code>'via [\${symbol}(\${version})](\$style)'</code> |
| <code>version_format</code> | <code>'v\${major}.\${minor}.\${patch}'</code>         |



[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            | '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 |
|--------|---------|
|        |         |



[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](#)

truncation\_length

1

symbol

'\$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



[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\$status] |  |
| symbol                | 'X'                 |  |
| success_symbol        | ' '                 |  |
| not_executable_symbol | '🚫'                 |  |
| not_found_symbol      | '🔍'                 |  |
| sigint_symbol         | '📦'                 |  |
| signal_symbol         | '⚡'                 |  |
| style                 | 'bold red'          |  |
| recognize_signal_code | true                |  |

[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](#)

|                           |   |
|---------------------------|---|
| map_symbol                | false   |
| pipestatus                | false   |
| pipestatus_separator      |   |
| pipestatus_format         | '\\[\$pipestatus'[\$symbol\$common,\$style]'' |
| pipestatus_segment_format |   |
| disabled                  | true  |

## 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](#)

|                |       |   |
|----------------|-------|---|
| status         | 127   | The exit code of the last command   |
| hex_status     | 0x7F  | The exit code of the last command in hex  |
| int            | 127   | The exit code of the last command   |
| common_meaning | ERROR | Meaning of the code if not a signal   |
| signal_number  | 9     | Signal number corresponding to the exit code, or 0 if signalled                         |
| signal_name    | KILL  | Name of the signal corresponding to the exit code, or "" if signalled                   |
| maybe_int      | 7     | Contains the exit code number when no meaning has been found                            |
| pipestatus     |       | Rendering of individual programs' exit codes, this is only available in pipestatus_form |
| 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

[status]
style = 'bg:blue'
symbol = '● '
success_symbol = '● SUCCESS'
format = '[[$symbol$common_meaning$signal_name$disabled]]'
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              |
|---------------------|--|--------------------------|
| <code>format</code> | <code>'[as \$symbol] (\$style)'</code> | 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]
```





disabled = false

## 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](#)

```
# 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                                     |
|---------------------|---|
| <code>format</code> | 'via<br>[\$symbol(\$version<br>)][\$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 or 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

[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                                      |
|-------------------|--|
| format            | 'via<br>[\$symbol\$workspace<br>](\$style) ' |
| version_format    | 'v\${raw}'                                   |
| symbol            | '❖'  |
| detect_extensions | ['tf', 'tfplan',<br>'tfstate']               |
| detect_files      | []   |
| detect_folders    | '.terraform'                                 |



[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 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})'
```



[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](#)

## Without Terraform version

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

```
[terraform]
```

```
format = '[windy $workspace]($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                       |
|-----------------------|---|-----------------------------------|
| <code>format</code>   | <code>'at<br/>[\$time]<br/>(\$style<br/>)'</code> | The format string for the module. |
| <code>use_12hr</code> | <code>false</code>                                | 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

Line 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**

[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   | 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 |  |
|--------|---------|--|
|        |         |  |

[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](#)[Exclude](#)

|                   |  |                              |
|-------------------|--|------------------------------|
| format            | 'via<br>[\$symbol(\$versio<br>n )](\$style)' | I<br>fc<br>mr                |
| version_format    | 'v\${raw}'                                   | T<br>fc<br>A<br>vi<br>I<br>I |
| symbol            | 't '   | A<br>st<br>re<br>tl<br>o     |
| style             | 'bold #0093A7'                               | T<br>tl                      |
| detect_extensions | [ '.typ' ]                                   | V<br>e:<br>sl<br>tr<br>mr    |
| detect_files      | [ 'template.typ' ]                           | V<br>fi<br>sl<br>tr          |
| detect_folders    | [ ]  | V<br>fc<br>sl<br>tr<br>mr    |
| disabled          | false  | D<br>mr                      |

[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 <code>typst</code> , alias for <code>typst_version</code> |
| typst_version | default | The current Typst version  |
| symbol        |         | Mirrors the value of option <code>symbol</code>                          |
| style*        |         | Mirrors the value of option <code>style</code>                           |

\*: 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



[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](#)

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 | 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](#)

|  |                   | 'via<br>[\$symbol(\$versio<br>n )](\$style)' | In<br>for<br>mc                                 |
|--|-------------------|--|---|
|  | format            | 'v\${raw}'                                   | The<br>for<br>Av<br>val<br>ri<br>mi<br>m.<br>pi |
|  | version_format    | '\${raw}'                                    | A f<br>str<br>rep<br>the<br>of                  |
|  | symbol            | '\$'   | Wl<br>ext<br>sho<br>tri<br>mc                   |
|  | detect_extensions | []   | Wl<br>file<br>sho<br>tri<br>mc                  |
|  | detect_files      | ['Vagrantfile']                              | Wl<br>file<br>sho<br>tri<br>mc                  |
|  | detect_folders    | []   | Wl<br>fol<br>sho<br>tri<br>mc                   |
|  | style             | 'cyan bold'                                  | The<br>the                                      |
|  | disabled          | false  | Dis<br>pla<br>mc                                |

[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  | Vagrant<br>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 [v $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



[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                               |
|-------------------|---|---|
| format            | 'via<br>[\$symbol(\$version )](\$style)'            | The format string for the output.         |
| version_format    | 'v\${raw}'  | The format string for the version output. |
| symbol            | 'v '  | A prefix for the symbol output.           |
| detect_extensions | ['v']   | File extensions to detect for files.      |
| detect_files      | ['v.mod',<br>'vPKG.json',<br>.vPKG-<br>LOCK.json' ] | File names to detect for files.           |
| detect_folders    | []  | Folders to detect.                        |



[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' | In the terminal, the prompt will be displayed in blue and bold. |
| disabled | false       | Disables the module entirely.                                   |

## Variables

| Variable | Example | Description                         |
|----------|---------|-------------------------------------|
| version  | v0.2    | The version of the module.          |
| 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. |
| Advanced Installation | 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



[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]``format = '[ $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  | Description   |
|-----------------------------|--|---|
| <code>format</code>         | <code>'via<br/>[\$symbol(\$version )](\$style)'</code> | The format string for the Zig version.                  |
| <code>version_format</code> | <code>'v\${raw}'</code>                                | The format string for the raw Zig version.              |
| <code>symbol</code>         | <code>'↳ '</code>                                      | The symbol used to separate the version from the style. |
| <code>style</code>          | <code>'bold yellow'</code>                             | The style for the Zig version.                          |

[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 monitor.                  |
| detect_extensions | ['zig'] | Will extend the shell to trigger mode. |
| detect_files      | []      | Will file shell to trigger mode.       |
| detect_folders    | []      | Will folder shell to trigger mode.     |

## 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](#)

## 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



[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](#)

## 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 | Help |
|---------|---------|------|
| command | ..      |      |

[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](#)

when

false

require\_repo

false

shell

S

description

'<custom  
module>'

detect\_files

[]

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](#)

detect\_extensions

[]

symbol

''

style

'bold green'

format

'[\$symbol(\$output )](\$style)'

disabled

false

os

use\_stdin

I  
t  
S  
w  
fc  
T  
b  
tt  
o  
T  
n  
T  
n  
C  
n  
t  
b  
li  
w  
p  
A  
b  
t  
w  
c  
b  
t  
st  
a:  
u  
ir  
d  
t  
n  
t  
sl

[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](#)

|                |       |   |
|----------------|-------|---|
| ignore_timeout | false | ↳ |
|----------------|-------|---|

## Variables

| Variable | Description                                       |
|----------|---|
| output   | The output of shell command in <code>shell</code> |
| symbol   | Mirrors the value of option <code>symbol</code>   |
| style*   | Mirrors the value of option <code>style</code>    |

\*: 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`.



[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](#)

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 command
detect_files = ['foo'] # can specify filters !
when = ''' test "$HOME" = "$PWD" '''
format = ' transcending [$output]($style)'

[custom.time]
command = 'time /T'
detect_extensions = ['pst'] # filters *.pst files
shell = ['pwsh.exe', '-NoProfile', '-Command']
use_stdin = false
```

[Edit this page on GitHub ↗](#)[← Advanced Installation](#)[Advanced Configuration →](#)