Starship Cross-shell prompt

START HERE

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

`starship config`
Edit starship.toml

`eval "$(starship init bash)"`
Enable in ~/.bashrc

`starship preset --list`
Browse presets

`eval "$(starship init zsh)"`
Enable in ~/.zshrc

`starship explain`
What is in my prompt?

`starship init fish | source`
Enable in config.fish

`starship timings`
Find slow modules

CONCEPTS

Starship is one binary for every shell: the shell calls it to draw each prompt, and one starship.toml configures them all.

Shell rc file runs starship init
↓ each prompt
starship prompt detects context
↓ reads
starship.toml format + modules
↓ renders
Your prompt git, dir, versions

Icons need a Nerd Font set in your terminal; without one use the no-nerd-font preset.

INSTALL AND INIT

`curl -sS https://starship.rs/install.sh | sh`
Install latest (Linux, macOS)

`brew install starship`
Install with Homebrew

`eval "$(starship init bash)"`
Enable in ~/.bashrc

`eval "$(starship init zsh)"`
Enable in ~/.zshrc

`starship init fish | source`
Enable in ~/.config/fish/config.fish

`Invoke-Expression (&starship init powershell)`
Enable in PowerShell $PROFILE

`starship init nu | save -f <autoload-file>`
Enable in Nushell

Needs Nushell 0.96+; save into $nu.data-dir/vendor/autoload/starship.nu

`curl -sS https://starship.rs/install.sh | sh -s -- -b ~/.local/bin`
Install without sudo

CONFIG FILE

`~/.config/starship.toml`
Default config file

`export STARSHIP_CONFIG=<path>`
Use another config file

`starship config`
Open config in $VISUAL / $EDITOR

`starship config <key> <value>`
Set a key, e.g. directory.format

`starship print-config`
Show config merged with defaults

`starship print-config <module>`
Show only one module's settings

FORMAT STRINGS

Syntax

Write	Meaning
`$name`	Insert variable or module
`[text](style)`	Styled text group
`(...)`	Hide if all variables inside are empty
`\$ \[ \]`	Escape literal characters

`format = '$all'`
Prompt layout (default)

`right_format = '$time'`
Right-side prompt

`add_newline = false`
No blank line between prompts

`palette = '<name>'`
Pick a palette from [palettes.<name>]

`bold fg:green bg:#1e1e2e`
Style string: words, colours, hex

Put values in single quotes: TOML then treats backslashes and $ literally.

MODULES

`success_symbol = '[❯](bold green)'`
Symbol after success

`error_symbol = '[❯](bold red)'`
Symbol after failure

`[directory] truncation_length = 3`
Parent dirs shown

`[git_status] ahead = '⇡'`
Git status symbols

`conflicted modified staged untracked`
Other git_status keys

`[cmd_duration] min_time = 2_000`
Show slow commands (ms)

`[time] disabled = false`
Show a clock (off by default)

`[username] show_always = true`
Always show the user

`[status] disabled = false`
Show exit codes (off by default)

`[nodejs] disabled = true`
Switch a module off

CUSTOM MODULES

`[custom.<name>]`
Define your own module

`command = 'git rev-parse --short HEAD'`
Output to display

DEBUG AND TUNING

`starship explain`
List what each shown module says

`starship timings`
Show module render times

`starship module <name>`
Render one module

DEBUG AND TUNING (cont.)

`starship prompt --status=<n> --jobs=<n>`
Render the prompt by hand

PRESETS

`starship preset --list`
List presets

`starship preset <name> -o ~/.config/starship.toml`
Write a preset as your config

`starship preset <name> -o <file> -f`
Overwrite an existing file

`starship preset <name>`
Print a preset to stdout

`no-runtimes`
Hide language versions

RECIPES

Minimal one-line prompt

```
# ~/.config/starship.toml
add_newline = false
format =
"$directory$git_branch$character"
```

Left prompt, info on the right

```
format = "$character"
right_format = "$all"
# zsh: ZLE_RPROMPT_INDENT=0
removes the gap
```

Colour palette

```
palette = "night"

[palettes.night]
blue = "#7aa2f7"

[directory]
style = "bold blue"
```

Fast start, then trim slow modules

```
starship timings
starship config command_timeout
1000
starship config nodejs.disabled
true
```