Hyprland

Dynamic tiling Wayland compositor

CONCEPTS

Mod is Super in the stock config (local mainMod = "SUPER"). Every key below is a default bind from the example hyprland.lua and is yours to change.

Since 0.55 the config is Lua (hyprland.lua); the old hyprlang syntax is deprecated. A bind is hl.bind("SUPER + Q", hl.dsp.exec_cmd("kitty")), keys joined with ' + '.

LAUNCH AND CLOSE

Mod + **Q**
Open terminal

Mod + **R**
Open app launcher

Mod + **C**
Close focused window

FOCUS AND MOVE

Mod + **Arrows**
Focus left / right / up / down

`hl.dsp.window.move({ direction = "left" })`
Move window in a direction

LAYOUT AND STATE

Mod + **V**
Toggle floating

Mod + **J**
Toggle split direction

`hl.dsp.window.fullscreen()`
Toggle fullscreen

WORKSPACES

Mod + **1-9**
Switch to workspace N

Mod + **Shift** + **1-9**
Move window to workspace N

Mod + **S**
Toggle scratchpad workspace

Mod + **Shift** + **S**
Send window to scratchpad

WORKSPACES (cont.)

`focus({ workspace = "previous" })`
Back to previous workspace

`focus({ workspace = "e+1" })`
Next open workspace

MOUSE

Mod + **LMB**
Drag a window

Mod + **RMB**
Resize a window

BIND SYNTAX

`hl.bind(<keys>, <dispatcher>, { <flags> })`
Define a keybind

`hl.dsp.exec_cmd(<cmd>)`
Run a shell command

`{ repeating = true }`
Repeat while held

`{ locked = true }`
Work on the lock screen

SUBMAPS

`hl.dsp.submap("<name>")`
Enter a submap

`hl.dsp.submap("reset")`
Leave to the global binds

`hl.define_submap("<name>", function() ... end)`
Define the binds in a submap

HYPRCTL

`hyprctl reload`
Reload the config

`hyprctl dispatch '<lua dispatcher>'`
Run a dispatcher now

`hyprctl clients`
List windows with properties

`hyprctl monitors`
List monitors

`hyprctl activewindow`
Show the focused window

`hyprctl -j <command>`
JSON output