

```

vim.g.mapleader = " " vim.keymap.set("i", "", "", { desc = 'Ensure that using
will always exit out of insert mode' }) vim.keymap.set({ 'n', 'i' }, "", ":w",
{ desc = 'Save file in buffer' }) vim.keymap.set("n", "re", ":e %", { desc =
'Reload current buffer' }) -vim.keymap.set("n", "ex", vim.cmd.Ex, {desc =
'Open the native file explorer'}) vim.keymap.set("n", "sp", ":vsplitw:Telescope
file_browser", { desc = 'Split the current tab vertically and open file explorer on
the right' }) vim.keymap.set("n", "nt", vim.cmd.tabe, { desc = 'Open a new tab'
}) vim.keymap.set("n", "Q", "") vim.keymap.set("n", "[s", ":normal! mz[s]z='z",
{ desc = 'Correct the first spelling error found when searching backwards from
cursor position' }) vim.keymap.set("v", "J", ".m '>+1gv=gv", { desc = 'Shift
selected line down by one line' }) vim.keymap.set("v", "K", ".m '>-2gv=gv", { desc
= 'Shift selected line up by one line' }) vim.keymap.set("n", "", "zz", { desc =
'Scroll down half page and keep cursor centered' }) vim.keymap.set("n", "", "ZZ",
{ desc = 'Scroll up half page and keep cursor centered' }) vim.keymap.set("n",
"n", "nzzzv", { desc = 'Find next instance of search and keep result at centered'
}) vim.keymap.set("n", "N", "Nzzzv", { desc = 'Find next instance of search
and keep result at centered' }) vim.keymap.set("n", "", "j", { desc = 'Go to split
content below', noremap = true }) vim.keymap.set("n", "", "k", { desc = 'Go to
split content above', noremap = true }) vim.keymap.set("n", "", "h", { desc =
'Go to split content to left', noremap = true }) vim.keymap.set("n", "", "l", {
desc = 'Go to split content to right', noremap = true }) vim.keymap.set("x",
"p", ""_dP", { desc = 'Paste content and do not store deleted content in buffer'
}) vim.keymap.set("n", "tms", "silent !tmux neww tmux-sessionizer", { desc =
'Use tmux-sessionizer to launch tmux session' }) vim.keymap.set("n", "kb", "silent
!sioyek ~/.config/nvim/keybinds.pdf", { desc = 'Open the neovim custom key-
binds help page that I wrote' })

```

```

- LSP keymaps vim.api.nvim_create_autocmd("LspAttach", { group =
vim.api.nvim_create_augroup('UserLspConfig', {}), callback = function(ev)
vim.bo[ev.buf].omnifunc = "v:lua.vim.lsp.omnifunc" local opts = { buffer
= ev.buf, noremap = true, silent = true } vim.keymap.set("n", "gd",
function() vim.lsp.buf.definition() end, opts) vim.keymap.set("n", "gD",
function() vim.lsp.buf.declaration() end, opts) vim.keymap.set("n", "gi",
function() vim.lsp.buf.implementation() end, opts) vim.keymap.set("n", "K",
function() vim.lsp.buf.hover() end, opts) vim.keymap.set("i", "", function()
vim.lsp.buf.signature_help() end, opts) vim.keymap.set({ "n", "v" }, "ca",
function() vim.lsp.buf.code_action() end, opts) vim.keymap.set("n", "rf",
function() vim.lsp.buf.references() end, opts) vim.keymap.set("n", "rn",
function() vim.lsp.buf.rename() end, opts) end, })

```