


```

let g:mapleader = " "
" This needs to go before plugins are loaded
let g:aledisablelsp = 1

call plug#begin('~/.vim/plugged')

" Plugs
Plug 'github/copilot.vim'
Plug 'scrooloose/nerdtree'
Plug 'scrooloose/nerdcommenter'
Plug 'tpope/vim-surround'
Plug 'tpope/vim-fugitive'
"Plug 'nvie/vim-flake8'
Plug 'christoomey/vim-tmux-navigator'
Plug 'mattn/emmet-vim'
"Plug 'unblevable/quick-scope'
"Plug 'sirver/ultisnips' /taking this one out because I
need tab to be
"unmapped
Plug 'dense-analysis/ale'
Plug 'neoclide/coc.nvim', {'branch': 'release'}
Plug 'vim-airline/vim-airline'
Plug 'junegunn/fzf', { 'do': { -> fzf#install() } }
Plug 'junegunn/fzf.vim'
Plug 'iberianpig/ranger-explorer.vim'
Plug 'kblin/vim-fountain'
nnoremap <silent><Leader>n :RangerOpenCurrentFile<CR>
nnoremap <silent><Leader>c :RangerOpenCurrentDir<CR>
nnoremap <silent><Leader>f :RangerOpenProjectRootDir<CR>
Plug 'rbgrouleff/bclose.vim' " Necessary for
ranger-explorer

"These are the javascript/react syntax bunch, spray and
pray here
Plug 'pangloss/vim-javascript'
"Plug 'mxw/vim-jsx'
Plug 'maxmellon/vim-jsx-pretty'
"Plug 'leafgarland/typescript-vim'
"Plug 'peitalin/vim-jsx-typescript'
"Plug 'styled-components/vim-styled-components', {
'branch': 'main' }

"This is for my Node templating engine
"Plug 'digitaltoad/vim-pug'

"Let's get LaTeX
Plug 'lervag/vimtex'
"LaTeX Setup
let g:vimtexviewmethod = 'zathura'
let g:texflavor='latex'
"let g:vimtexquickfixmode=0
set conceallevel=1
let g:texconceal='abdmg'

"Better searching
"Plug 'junegunn/vim-pseudocl'
"Plug 'junegunn/vim-oblique'

```

```

Plug 'JamshedVesuna/vim-markdown-preview'
let vimmarkdownpreviewgithub=1

"Python syntax highlighting
Plug 'numirias/semshi', {'do': ':UpdateRemotePlugins'}

"React snippets
Plug 'dsznajder/vscode-es7-javascript-react-snippets', {
'do': 'yarn install --frozen-lockfile && yarn compile' }

call plug#end()

"Fuzzy finding
nnoremap <silent> <leader><Space> :Files<CR>
nnoremap <silent> <leader>a :Buffers<CR>
nnoremap <silent> <leader>g :Rg<CR>
nnoremap <silent> <leader>l :Lines<CR>
nnoremap <silent> <leader>L :BLines<CR>
nnoremap <silent> <leader>A :Windows<CR>

set nohlsearch

"Coc Snippets Shortcut
:nnoremap <leader>sn :CocCommand snippets.editSnippets<cr>

"Quick Scope
let g:qshighlightonkeys = ['f','F','t','T']

" Indent
set autoindent
set smartindent

"" Setup ale
let g:alelinters = {
\   'javascript': ['eslint'],
\   'typescript': ['eslint'],
\   'python': ['flake8'],
\}
let g:alefixers = {
\   'javascript': ['prettier', 'eslint'],
\   'typescript': ['prettier', 'eslint'],
\   'python': ['black'],
\   '': ['removetrailinglines', 'trimwhitespace'],
\}

let g:alesignererror = ' '
let g:alesignwarning = ' '
let g:alefixonsave = 1

" Deal with annoying escape delay
set timeoutlen=1000 ttimeoutlen=0

"Set tab to always be 4 spaces
set tabstop=2
set softtabstop=2
set shiftwidth=2

```

```
"Automatically run python script with f9
autocmd FileType python map <buffer> <F9> :w<CR>:exec
'!python3' shellescape(@%, 1)<CR>
autocmd FileType python imap <buffer> <F9>
<esc>:w<CR>:exec '!python3' shellescape(@%, 1)<CR>
```

```
"Line numbering
set nu
```

```
"Vim splits
nnoremap <C-J> <C-W><C-J>
nnoremap <C-K> <C-W><C-K>
nnoremap <C-L> <C-W><C-L>
nnoremap <C-H> <C-W><C-H>
set splitbelow
set splitright
```

```
"Better background highlight color than pink
:highlight Pmenu ctermbg=gray guibg=gray
```

```
" Use the right python
let g:python3hostprog = '/opt/homebrew/bin/python3.11'
```

```
"Smartf
"" press <esc> to cancel.
nmap f <Plug>(coc-smartf-forward)
nmap F <Plug>(coc-smartf-backward)
nmap ; <Plug>(coc-smartf-repeat)
nmap , <Plug>(coc-smartf-repeat-opposite)
```

```
augroup Smartf
  autocmd User SmartfEnter :hi Conceal ctermfg=220
  guifg=#6638F0
  autocmd User SmartfLeave :hi Conceal ctermfg=239
  guifg=#504945
augroup end
```

```
let g:copilotfiletypes = {
  \      'tex': v:false,
  \      'fountain': v:false,
  \    }

```

```
"COC
source ~/.config/nvim/config/coc.vim
```