**终端**

**term2**

**标签**

新建标签：command + t

关闭标签：command + w

切换标签：command + 数字 command + 左右方向键

切换全屏：command + enter

查找：command + f

**分屏**

垂直分屏：command + d

水平分屏：command + shift + d

切换屏幕：command + option + 方向键 command + [ 或 command + ]

查看历史命令：command + ;

查看剪贴板历史：command + shift + h

**其他**

清除当前行：ctrl + u

到行首：ctrl + a

到行尾：ctrl + e

前进后退：ctrl + f/b (相当于左右方向键)

上一条命令：ctrl + p

搜索命令历史：ctrl + r

删除当前光标的字符：ctrl + d

删除光标之前的字符：ctrl + h

删除光标之前的单词：ctrl + w

删除到文本末尾：ctrl + k

交换光标处文本：ctrl + t

清屏1：command + r

清屏2：ctrl + l

**omz配置**

Oh my zsh 模板 = crcandy https://github.com/ohmyzsh/ohmyzsh/wiki/Themes

**安装**

|  |
| --- |
| Shell #!/bin/sh  # Set source and target directories powerline\_fonts\_dir="$( cd "$( dirname "$0" )" && pwd )"  # if an argument is given it is used to select which fonts to install prefix="$1"  if test "$(uname)" = "Darwin" ; then  # MacOS  font\_dir="$HOME/Library/Fonts" else  # Linux  font\_dir="$HOME/.local/share/fonts"  mkdir -p $font\_dir fi  # Copy all fonts to user fonts directory echo "Copying fonts..." find "$powerline\_fonts\_dir" \( -name "$prefix\*.[ot]tf" -or -name "$prefix\*.pcf.gz" \) -type f -print0 | xargs -0 -n1 -I % cp "%" "$font\_dir/"  # Reset font cache on Linux if which fc-cache >/dev/null 2>&1 ; then  echo "Resetting font cache, this may take a moment..."  fc-cache -f "$font\_dir" fi  echo "Powerline fonts installed to $font\_dir" |

**卸载**

|  |
| --- |
| Shell #!/bin/sh  # Set source and target directories powerline\_fonts\_dir="$( cd "$( dirname "$0" )" && pwd )"  # if an argument is given it is used to select which fonts to uninstall prefix="$1"  if test "$(uname)" = "Darwin" ; then  # MacOS  font\_dir="$HOME/Library/Fonts" else  # Linux  font\_dir="$HOME/.local/share/fonts"  if test ! -d "$font\_dir" ; then  echo "It seems there are no powerline fonts installed on your system. Uninstall not needed."  exit 0  fi fi  # Remove all fonts from user fonts directory echo "Removing fonts..." find "$powerline\_fonts\_dir" \( -name "$prefix\*.[ot]tf" -or -name "$prefix\*.pcf.gz" \) -type f -print0 | xargs -n1 -0 -I % sh -c "rm -f \"\$0/\${1##\*/}\"" "$font\_dir" %  # Reset font cache on Linux if which fc-cache >/dev/null 2>&1 ; then  echo "Resetting font cache, this may take a moment..."  fc-cache -f "$font\_dir" fi  echo "Powerline fonts uninstalled from $font\_dir" |

**插件**

|  |
| --- |
| Shell 安装zsh-autosuggestions和zsh-syntax-highlighting插件：  git clone https://github.com/zsh-users/zsh-autosuggestions.git $ZSH\_CUSTOM/plugins/zsh-autosuggestions git clone https://github.com/zsh-users/zsh-syntax-highlighting.git $ZSH\_CUSTOM/plugins/zsh-syntax-highlighting  plugins=(  # 其他插件...  zsh-autosuggestions  zsh-syntax-highlighting ) |

**vim配置**

代码补全（coc）https://github.com/neoclide/coc.nvim

|  |
| --- |
| Shell 安装nodejs  https://nodejs.org/en/download/  安装vim-plug curl -fLo ~/.vim/autoload/plug.vim --create-dirs \  https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim -----------  " 插件列表 call plug#begin('~/.vim/plugged')  " 添加 Coc.nvim 插件 Plug 'neoclide/coc.nvim', {'branch': 'release'}  call plug#end() -----------   :CocInstall coc-pyright coc-java coc-yaml coc-sh coc-docker  let g:coc\_global\_extensions = ['coc-pyright', 'coc-java', 'coc-yaml', 'coc-sh', 'coc-docker'] |