

## iTerm2

iTerm是mac上的一款终端模拟器辅助软件，极大的方便和自定义terminal的配置和使用。

### ⌘常用快捷键

- ⌘ + Click：可以打开文件，文件夹和链接
- ⌘ + n：新建窗口
- ⌘ + t：新建标签页
- ⌘ + w：关闭当前页
- ⌘ + 数字 | ⌘ + 方向键：切换标签页
- ⌘ + 数字：切换窗口
- ⌘ + enter：切换全屏
- ⌘ + d：左右分屏
- ⌘ + d：上下分屏
- ⌘ + ;：自动补全历史记录
- ⌘ + h：自动补全剪贴板历史
- ⌘ + e：查找所有来定位某个标签页
- ⌘ + r | ^ + l：清屏
- ⌘ + /：显示光标位置
- ⌘ + b：历史回放
- ⌘ + f：查找，然后用 tab 和 ⌘ + tab 可以向右和向左补全，补全之后的内容会被自动复制，还可以用 ⌘ + enter 将查找结果输入终端
- 选中即复制，鼠标中键粘贴

### ^常用快捷键

- ^ + u：清空当前行
- ^ + a：移动到行首
- ^ + e：移动到行尾
- ^ + f：向前移动
- ^ + b：向后移动
- ^ + p：上一条命令
- ^ + n：下一条命令
- ^ + r：搜索历史命令
- ^ + y：召回最近用命令删除的文字
- ^ + h：删除光标之前的字符
- ^ + d：删除光标所指的字符
- ^ + w：删除光标之前的单词
- ^ + k：删除从光标到行尾的内容
- ^ + t：交换光标和之前的字符

## Zsh

zsh 的命令补全功能非常强大,可以补齐路径,补齐命令,补齐参数等

//安装

```
brew install zsh
```

//查看版本

```
zsh --version
```

## ohmyz

主要用来管理 Zsh 配置的，还提供了各种主题

### zsh环境变量失效问题

新版的zsh在登录的时候默认只会加载~/.zshrc的配置文件，以前配置的环境变量会无效，尝试网上的各种方法均无效

发现配置文件有这样一条说明

If you come from bash you might have to change your \$PATH.

于是简单粗暴的解决方案：

编辑~/.zshrc文件

在开头添加一下配置

```
export PATH=$HOME/bin:/usr/local/bin:$PATH
```

```
source $HOME/.bashrc
```

```
source $HOME/.bash_profile
```

作者：卖梦想的孩纸

链接：<https://www.jianshu.com/p/8ef18883c958>

来源：简书

简书著作权归作者所有，任何形式的转载都请联系作者获得授权并注明出处。

## git

```
alias g='git'
```

### git fetch

```
alias gf='git fetch'
```

```
alias gfa='git fetch --all --prune'
```

```
alias gfo='git fetch origin'
```

### git pull

# git pull --rebase 这样的结局就完全不一样。—rebase 并不会产生一个commit提交,详见 <https://www.cnblogs.com/wangiqngpei557/p/6056624.html>

```
# git pull --rebase = git fetch + git rebase
```

```
alias gup='git pull --rebase'
```

```
alias gupv='git pull --rebase -v'
```

```
alias glum='git pull upstream master'
```

### git status

#显示当前目录下文件的状态，红色的问号表示git不知道这份文件是什么，需要add，绿色的A表示add的缩写。M表示有修改的文件

```
alias gsb='git status -sb'
```

```
alias gsd='git svn dcommit'
```

```
alias gsi='git submodule init'
```

```
alias gsps='git show --pretty=short --show-signature'
```

```
alias gsr='git svn rebase'
```

```
alias gss='git status -s'
```

```
alias gst='git status'
```

```
alias gsu='git submodule update'
```

### **git add**

```
alias ga='git add'
alias gaa='git add --all'
alias gapa='git add --patch'
alias gau='git add --update'
alias gap='git apply'
```

### **git stash**

```
alias gsta='git stash save'
alias gstaa='git stash apply'
alias gstc='git stash clear'
alias gstd='git stash drop'
alias gstl='git stash list'
alias gstp='git stash pop'
alias gsts='git stash show --text'
```

### **git reset**

```
# 将某个已经暂存的文件取消暂存
alias grh='git reset HEAD'
```

# 虽然在调用时加上 --hard 选项可以令 git reset 成为一个危险的命令（译注：可能导致工作目录中所有当前进度丢失！），但本例中工作目录内的文件并不会被修改。不加选项地调用 git reset 并不危险 — 它只会修改暂存区域。

```
alias grhh='git reset HEAD --hard'
alias gru='git reset --'
```

### **git commit**

```
alias gc='git commit -v'
alias gc!= 'git commit -v --amend'
alias gcn!= 'git commit -v --no-edit --amend'
alias gca='git commit -v -a'
alias gca!= 'git commit -v -a --amend'
alias gcan!= 'git commit -v -a --no-edit --amend'
alias gcans!= 'git commit -v -a -s --no-edit --amend'
alias gcam='git commit -a -m'
alias gcsn='git commit -s -m'
```

### **git push**

```
alias gp='git push'
alias gpd='git push --dry-run'
alias gpoat='git push origin --all && git push origin --tags'
compdef _git gpoat=git-push
alias gpu='git push upstream'
alias gpv='git push -v'
```

### **git log**

```
alias gl='git pull'
alias glg='git log --stat'
alias glgp='git log --stat -p'
alias glgg='git log --graph'
```

```
alias glgga='git log --graph --decorate --all'
alias glgm='git log --graph --max-count=10'
alias glo='git log --oneline --decorate'
alias glol='git log --graph --pretty="%Cred%h%Creset -%C(yellow)%d%Creset
%s %Cgreen(%cr) %C(bold blue)<%an>%Creset" --abbrev-commit'
alias glola="git log --graph --pretty='%Cred%h%Creset -%C(yellow)
%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit
--all"
alias glog='git log --oneline --decorate --graph'
# 它会输出你的提交历史、各个分支的指向以及项目的分支分叉情况。
alias gloga='git log --oneline --decorate --graph --all'
```

### **git branch**

```
alias gb='git branch'
alias gba='git branch -a'
alias gbd='git branch -d'
alias gbda='git branch --no-color --merged | command grep -vE "^(\*|
\s*(master|develop|dev)\s*$)" | command xargs -n 1 git branch -d'
alias gbl='git blame -b -w'
alias gbnm='git branch --no-merged'
alias gbr='git branch --remote'
```

### **git merge**

```
alias gm='git merge'
alias gmom='git merge origin/master'
alias gmt='git mergetool --no-prompt'
alias gmtvim='git mergetool --no-prompt --tool=vimdiff'
alias gmum='git merge upstream/master'
alias gma='git merge --abort'
```

### **git rebase**

```
alias grb='git rebase'
alias grba='git rebase --abort'
alias grbc='git rebase --continue'
alias grbi='git rebase -i'
alias grbm='git rebase master'
alias grbs='git rebase --skip'
```

### **git checkout**

```
alias gcb='git checkout -b'
alias gcf='git config --list'
alias gcl='git clone --recursive'
alias gclean='git clean -fd'
alias gpristine='git reset --hard && git clean -dfx'
alias gcm='git checkout master'
alias gcd='git checkout develop'
alias gcmmsg='git commit -m'
alias gco='git checkout'
alias gcount='git shortlog -sn'
compdef _git gcount
```

```
alias gcp='git cherry-pick'
alias gcpa='git cherry-pick --abort'
alias gcpc='git cherry-pick --continue'
alias gcs='git commit -S'
```

### **git diff**

```
alias gd='git diff'
alias gdca='git diff --cached'
alias gdcw='git diff --cached --word-diff'
alias gdct='git describe --tags `git rev-list --tags --max-count=1`'
alias gdt='git diff-tree --no-commit-id --name-only -r'
alias gdw='git diff --word-diff'
```

### **git bisect**

```
alias gbs='git bisect'
alias gbsb='git bisect bad'
alias gbsg='git bisect good'
alias gbsr='git bisect reset'
alias gbss='git bisect start'
```

### **git gui citoo**

```
alias gg='git gui citool'
alias gga='git gui citool --amend'
```

### **git remote**

# git remote add <shortname> <url> 添加一个新的远程 Git 仓库，同时指定一个你可以轻松引用的简写：

```
alias gra='git remote add'
alias grmv='git remote rename'
alias grrm='git remote remove'
alias grset='git remote set-url'
alias grt='cd $(git rev-parse --show-toplevel || echo ".")'
alias grup='git remote update'
alias grv='git remote -v'
```

### **git tag**

```
alias gts='git tag -s'
alias gtv='git tag | sort -V'
```

```
alias gunignore='git update-index --no-assume-unchanged'
alias gunwip='git log -n 1 | grep -q -c "\-\\-wip\\-\\-" && git reset HEAD~1'
```

```
alias gwch='git whatchanged -p --abbrev-commit --pretty=medium'
alias gwip='git add -A; git rm $(git ls-files --deleted) 2> /dev/null; git commit --no-verify -m "--wip-- [skip ci]"'
```

作者：lxyz22zp

链接：<https://www.jianshu.com/p/010939b5d466>

来源：简书

简书著作权归作者所有，任何形式的转载都请联系作者获得授权并注明出处。

