

dotfiles

How to Use

やること

```
git clone https://github.com/kagerouttepasso/dotfiles
cd dotfiles
./install.sh
```

注意

結構依存しているパッケージがある。

必要なパッケージ一覧(Ubuntuのパッケージ名)以下のとおり

- git
- tmux
- zsh
- [pandoc](#)
- exuberant-ctags
- xclip
- curl
- ~~silversearcher-ag~~
- [pt](#)
- clang
- [rtags](#)
- npm
 - mermaid
 - markdown-pdf

ubuntu 12.04以上ならこつちを `git clone` してchefを走らせたほうが賢明ですー

[Chef_work](#)

Windows環境

MSYS2

- [ここからダウンロード](#)
- [/MSYS2/register_msys2_env.bat](#)使って環境変数を登録
- [/MSYS2/register_right_click.reg](#)使って右クリックメニュー登録
- X11転送使うときはXming使うこと。xmingは[/windows/choco/install_packages.bat](#)使ってインストール

Cygwin

Windows環境でも基本的に必要パッケージをインストールしてやること のコマンドを入れれば問題ない。

x11のパッケージセットをインストールすればssh接続先のクリップボードの共有もできる用になっています

Cygwinを使うのをやめました

入れるパッケージ

- git
- zsh
- ctags

- x11すべて
- curl
- openssh
- tmux

teraterm

設定ファイル を読みこめばいい感じになる

香り屋版Gvimへの対応

1. Source Treeをダウンロードしてインストール、この `Dotfiles` を好きなところにCloneしてくる。
2. `install.bat` を管理者権限で実行
手順は[unite-everythingのヘルプ](#)を参照
3. 香り屋版Gvimをダウンロード
4. `Windows版ag`をダウンロードしてくる
ダウンロードしてきた実行ファイルはgvimのフォルダにつっこむ
5. gvimを起動ここで色々エラーが出るけど気にしない。
プラグインのインストールするか聞かれるけどNOを選択。
`:NeobundleInstall` でプラグインをインストール!!
6. 快適なGvimライフへようこそ

Windows上だとrgrepがうまく動かないのでUniteGrepを使ってください

dotfiles	C:\Users\UserName
./_vimrc	_vimrc
./_gvimrc	_gvimrc
./_vim	.vim

その他メモ

- シンボリックリンクを貼るのには[このアプリ](#)を使用するといい感じです
- フォントのインストールを行うとかつこいいCygwin、teraterm環境を構築できる。
インストールの仕方は[Help](#)を参照
`install.bat`を実行すれば自動で入る
- mingwについて
 - [Mingw64のダウンロード](#)
 - [Mingw32とmsysのダウンロード](#)
 - [Node.jsのダウンロード](#)

コマンド

自前で設定したコマンドとか

zsh

all Platform

command	mean
tmux	tmux -2
sshx	ssh -XC

command	mean
zshrc	<code>vim ~/.zshrc</code>
vimrc	<code>vim ~/.vimrc</code>
sshconfig	<code>vim ~/.ssh/config</code>
j	run autojump

Linux

command	mean
tmuxx	<code>tmux attach -d</code>
md2html	file encoded HTML in pandoc, and open in browser

cygwin

command	mean
cdcyg	move to desktop

vim

主な変更

- バックアップ取らない
- スワップ作らない
- バックスペースで色々消せる
- ビープ音鳴らさない
- モードを表示
- マウスを使用する
- クリップボードを使用する
- 全角スペースを表示
- tagsファイルは6階層上のもので読み込む
- 前回終了したカーソル行に移動
- 補完にはNeocompleteを使用するが、使えない時はNeocomplecacheを使う

vimのシンタックス

javascript

vimを使ってjavascriptの編集をするときは `hshint` をインストールするとシンタックスのチェックをしてもらえる
windows環境でそれを行うときは、[nodejs](#)をダウンロードして下記のコマンドを実行

拡張コマンド

挿入モード

command	mean
<code><C-p></code>	貼付け
<code><C-j></code>	<code><Down></code>
<code><C-k></code>	<code><Up></code>
<code><C-h></code>	<code><Left></code>
<code><C-l></code>	<code><Right></code>

command	mean
jj	<esc>
,	,<Space>

ノーマルモード

command	mean
<Space>	次のバッファへ
b	前のバッファへ
<C-g><C-g>	カーソル下の単語をバッファを対象にGrep
<C-g><C-r>	カーソル下の単語をカレントフォルダから再帰的にGrep
Ctags	!:ctags -R
tt	タグジャンプ
tb	現在開いているバッファを閉じてタグスタックをひとつ戻す
t1~t9	対応したタブ番号へ飛ぶ
tn	次のタブへ
tp	前のタブへ
tr	バッファをタブ展開
0	行末へ
9	行頭へ
+	フレームサイズ増加
-	フレームサイズ減少
(対応するカッコに移動
)	対応するカッコに移動
vy	カーソル位置の単語をヤंक
<C-j>	<C-w>j
<C-k>	<C-w>k
<C-l>	<C-w>l
<C-h>	<C-w>h
gl	:GitGutterNextHunk
gh	:GitGutterPrevHunk
<C-p>	ペーストするレジスタを一つ古いものにする(ペーストした直後のみ)
<C-n>	ペーストするレジスタを一つ新しいものにする(ペーストした直後のみ)
ss	置換開始
sw	カーソル以下の単語を置換
gx	カーソル以下の単語でブラウザ検索

ビジュアルモード

command	mean
<code>0</code>	行末へ
<code>v</code>	行末へ
<code>9</code>	行頭へ
<code>sw</code>	文書全体で、選択した文章を置換
<code>ss</code>	選択した範囲を置換
<code><Enter></code>	Alignの修正モードへ
<code>gx</code>	カーソル以下の単語でブラウザ検索

コマンド

command	mean
Tig	gitのログビューアー
QB	ウィンドウの形を変えずにバッファを閉じる
QB	バッファを閉じてウィンドウも閉じる
TN	NERDTree起動
TS	SourceExplorer起動
TT	TagList起動
TA	上記3つを起動
Rgrep	カーソル下の単語をカレントディレクトリからgrep
Bgrep	カーソル下の単語をバッファからgrep
VimShell	VimShellの起動
VimFiler	VimFilerの起動

Unite

Grepの検索エンジンにはagが使えればそちらを使う

command	mean
f	[unite]
[unite]u	<code><C-u>Unite -no-split<Space></code>
[unite]a	全部乗せ <code><C-u>Unite -buffer-name=all buffer file_mru bookmark file grep:. find:.<CR><C-r><C-w><CR>*<C-r><C-w>*<CR></code>
[unite]b	バッファ一覧 <code><C-u>Unite buffer<CR></code>
[unite]h	最近使用したファイル一覧 <code><C-u>Unite buffer file_mru<CR></code>
[unite]d	カレントディレクトリからファイル一覧 <code><C-u>Unite -buffer-name=files file<CR></code>
[unite]D	Gitのカレントディレクトリからファイル一覧 <code><C-u>Unite -buffer-name=git_files file_rec/async:! -winheight=15<CR></code>
[unite]c	現在のバッファのディレクトリからファイル一覧 <code><C-u>UniteWithBufferDir -buffer-name=files file<CR></code>
[unite]r	レジスター一覧 <code><C-u>Unite register<CR></code>
[unite]g	グレップ検索 <code><C-u>Unite grep:. -buffer-name=serch-buffer -no-quit<CR><C-r><C-w><CR></code>
[unite]G	グレップ検索(対話的) <code><C-u>Unite grep -no-quit<CR></code>

command	mean
[unite]f	ファイル検索(win以外) <C-u>Unite find:. -buffer-name=serch-file -no-quit<CR>*<C-r><C-w>*<CR>
[unite]f	ファイル検索(win) 日本語が文字化けしてしまう... <C-u>UniteWithCursorWord everything/async -no-quit<CR>
[unite]F	ファイル検索(win) <C-u>Unite everything/async -no-quit<CR>
[unite]s	ソースリスト <C-u>Unite source<CR>
[unite]o	アウトラインを展開 <C-u>Unite outline -buffer-name=outline -no-focus -no-start-insert<CR>
[unite]t	アウトラインを展開 <C-u>Unite outline -vertical -winwidth=50 -buffer-name=outline -no-focus -no-start-insert -no-quit<CR>

teratermとscreen

キーバインドを変更してある。とりあえず`tmux.conf`と`screenrc`を見てください。
 バインドはteratermが `Ctrl+o` screenが `ctrl+a` にバインドされてます

```
npm config set proxy http://proxy.com:port --global
npm install jshint --global
```

memo

cygwinでctagsを使うときはExuburantCtagsを使わないとうまく行かなそう

todo

OSS

- ./vscode\github-markdown.css
 MIT : <https://github.com/sindresorhus/github-markdown-css>