VIM Hacks

Why Vim
Vim 能幫助我們什麼

c9s / Cornelius
林佑安
AIINK.com

編輯器

It doesn't matter

It does matter

好的編輯器帶你上天堂

差的編輯器帶你住套房

敏捷開発

敏捷

太多

選擇

想當年…

年輕不懂事

第一次

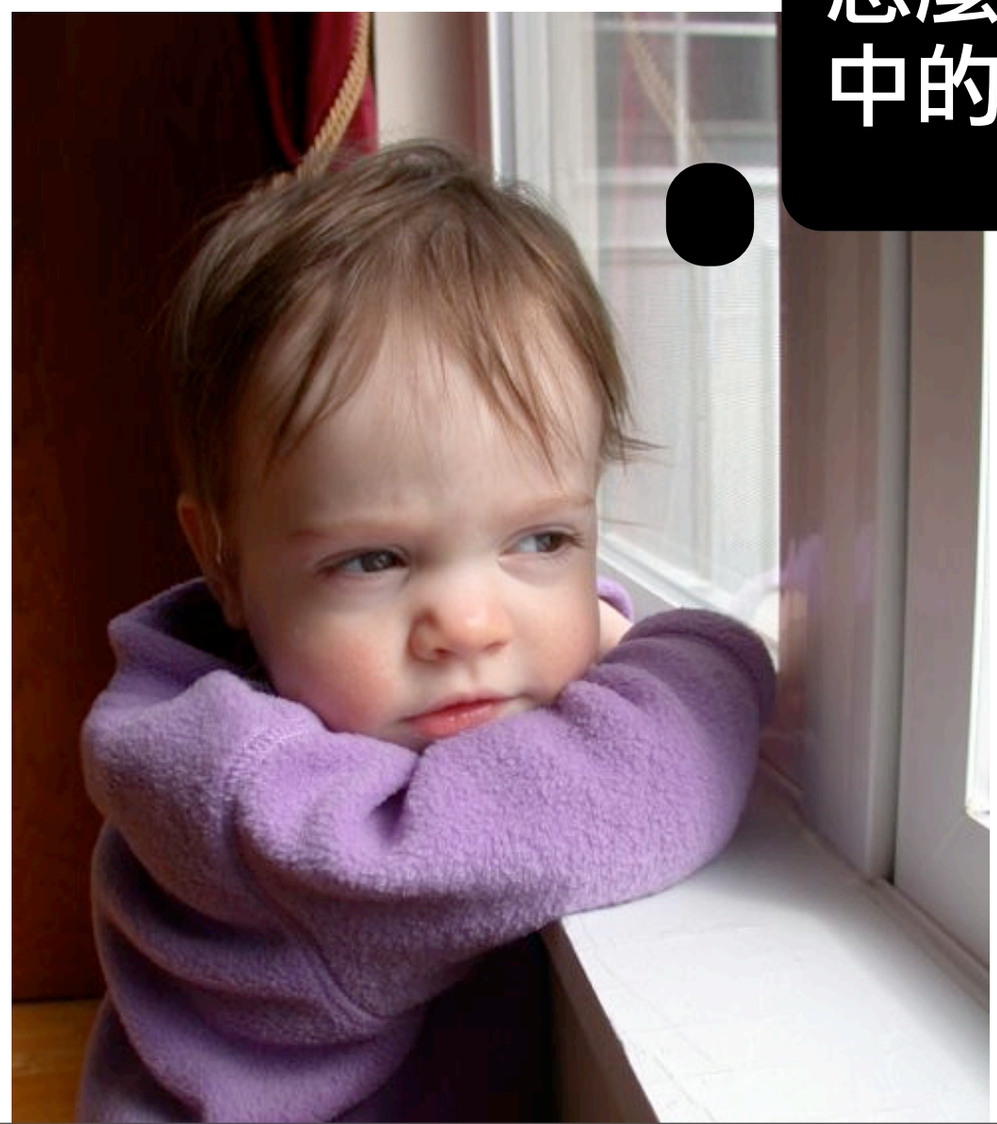
Microsoft Visual Studio

Eclipse

Code Block

Notepad++

太多



怎麼都沒有理想中的編輯器呢？

鬱卒了....

於是....

- 繼續嘗試其他編輯器

- 繼續嘗試其他編輯器
- 花銀子買 IDE , Editor

- 繼續嘗試其他編輯器
- 花銀子買 IDE , Editor
- 又再寫一個 XXX Editor ..

- 繼續嘗試其他編輯器
- 花銀子買 IDE , Editor
- 又再寫一個 XXX Editor ..
- 又或者....

筆記本

自從不用 Microsoft Windows 之後 ...

GEdit

Emacs

TextMate

VIM

最糟糕的

編輯方式

```
#!/usr/bin/env perl
my $hppiness = COS::CUP();
```

```
#!/usr/bin/env perl
my $hppiness = COS::CUP();
```

Oops!

少了“a”

```
#!/usr/bin/env perl
my $hppiness = COS::CUP();
```

x N

```
#!/usr/bin/env perl
my $hppiness = COS::CUP();
```

x N

```
#!/usr/bin/env perl
my $hppiness = COS::CUP();
```

x N

```
#!/usr/bin/env perl
my $hppiness = COS::CUP();
```

x N

```
#!/usr/bin/env perl
my $hppiness = COS::CUP();
```

x N

```
#!/usr/bin/env perl
my $hppiness = COS::CUP();
```

x N

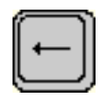
```
#!/usr/bin/env perl
my $hppiness = COS::CUP();
```

x N

```
#!/usr/bin/env perl
my $hppiness = COS::CUP();
```

x N

```
#!/usr/bin/env perl
my $hppiness = COS::CUP();
```



x N

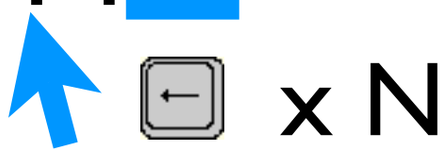
```
#!/usr/bin/env perl
my $hppiness = COS::CUP();
```

x N

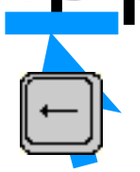
```
#!/usr/bin/env perl
my $hppiness = COS::CUP();
```

x N

```
#!/usr/bin/env perl
my $hppiness = COS::CUP();
```



```
#!/usr/bin/env perl
my $hppiness = COS::CUP();
```



```
#!/usr/bin/env perl
my $hppiness = COS::CUP();
```

```
#!/usr/bin/env perl
my $happiness = COS::CUP();
```

```
#!/usr/bin/env perl
my $happiness = COS::CUP();
```

一百個類似的狀況？

```
#!/usr/bin/env perl
my $happiness = COS::CUP();
```

一百個類似的狀況？

糟

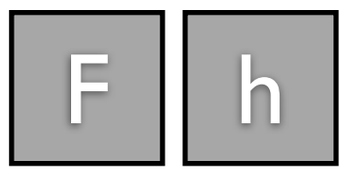
浪費時間

移動游標

The VIM way...

VIM:

```
#!/usr/bin/env perl
my $hppiness = COS::CUP();
```



VIM:

```
#!/usr/bin/env perl
my $happiness = COS::CUP();
```

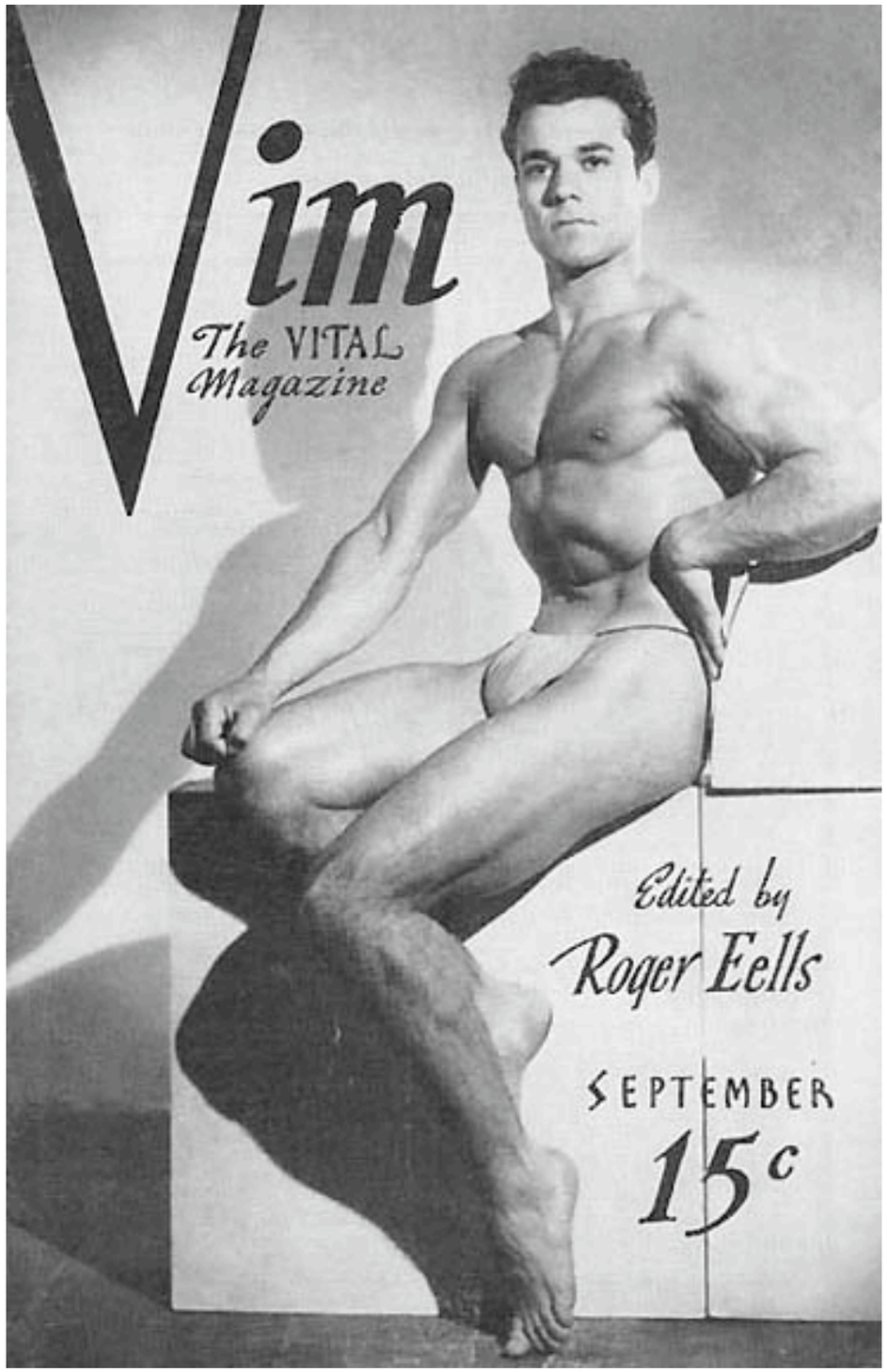
a ppend a


VIM

VIM 不是

IDE

不是…

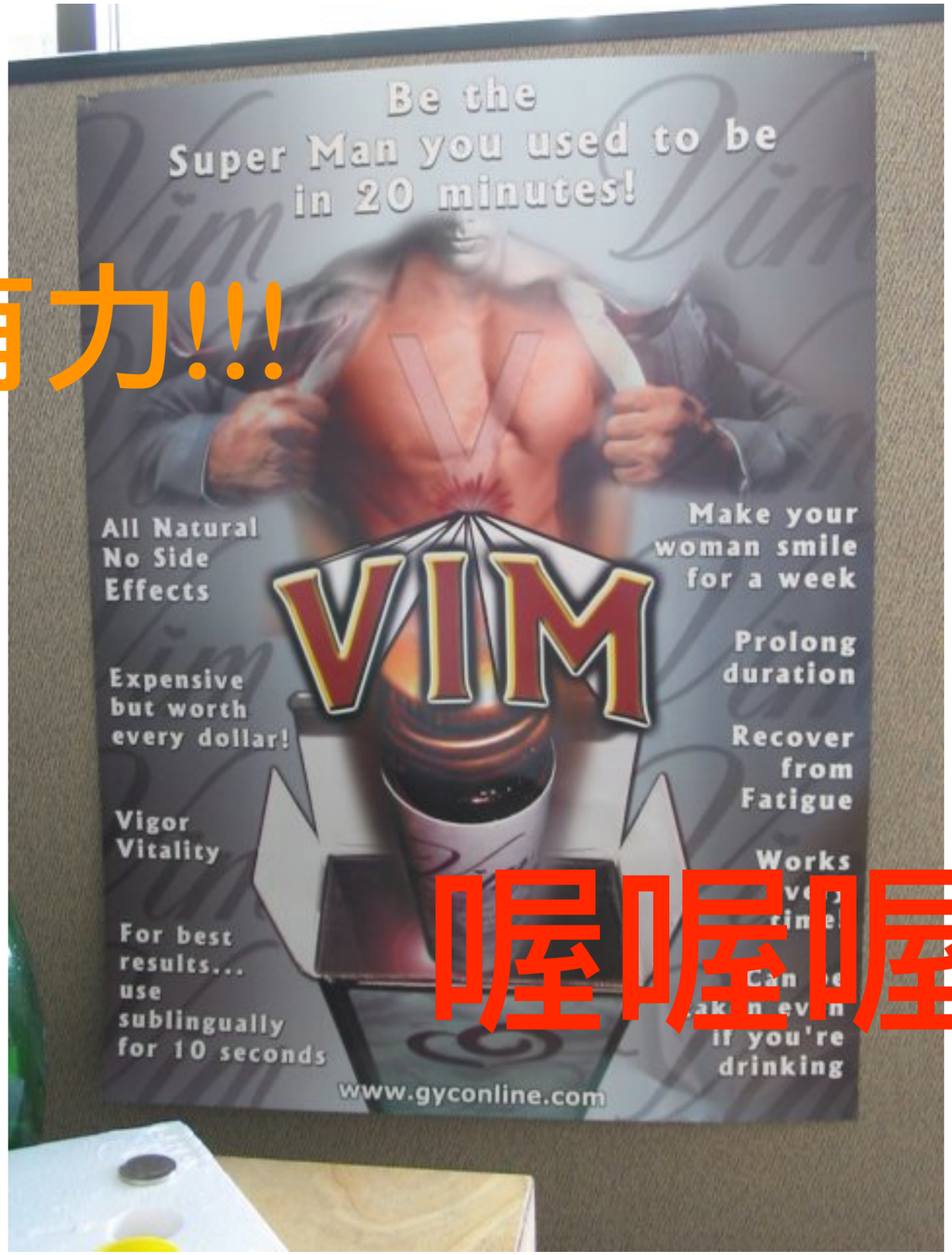

Vim
The VITAL Magazine
Edited by
Roger Eells
SEPTEMBER
15c

也不是...

強而有力!!!


Be the
Super Man you used to be
in 20 minutes!
All Natural
No Side
Effects
Expensive
but worth
every dollar!
Vigor
Vitality
For best
results...
use
sublingually
for 10 seconds
VIM
Make your
woman smile
for a week
Prolong
duration
Recover
from
Fatigue
Works
every
time!
Can be
taken even
if you're
drinking
www.gyconline.com

強而有力!!!


Be the
Super Man you used to be
in 20 minutes!
All Natural
No Side
Effects
Expensive
but worth
every dollar!
Vigor
Vitality
For best
results...
use
sublingually
for 10 seconds
VIM
Make your
woman smile
for a week
Prolong
duration
Recover
from
Fatigue
Works
if you're
drinking
www.gyconline.com

喔喔喔喔!!!

VIM 是

VI Improved

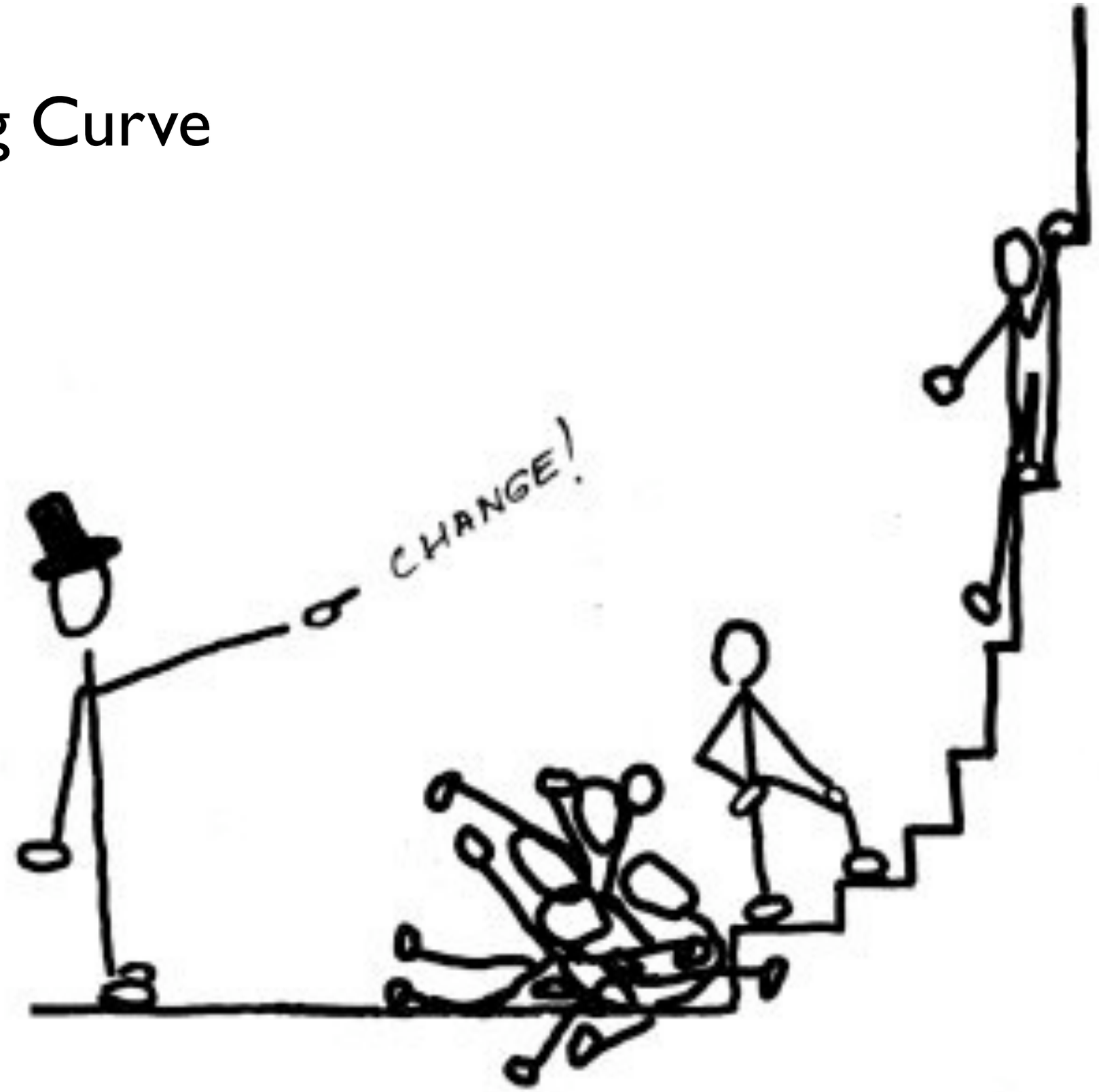
進階

編輯器

提高

文字編輯效率

Learning Curve



各種Vim

各種Vim

- Vim

各種Vim

- Vim
- gVim

各種Vim

- Vim
- gVim
- gVim Easy

各種Vim

- Vim
- gVim
- gVim Easy
- MacVim

各種Vim

- Vim
- gVim
- gVim Easy
- MacVim
- Vi in Emacs

各種Vim

- Vim
- gVim
- gVim Easy
- MacVim
- Vi in Emacs
- Vi in Bash

各種Vim

- Vim
- gVim
- gVim Easy
- MacVim
- Vi in Emacs
- Vi in Bash
- etc ...

Features

1.編輯模式

Mode

四種以上

編輯模式

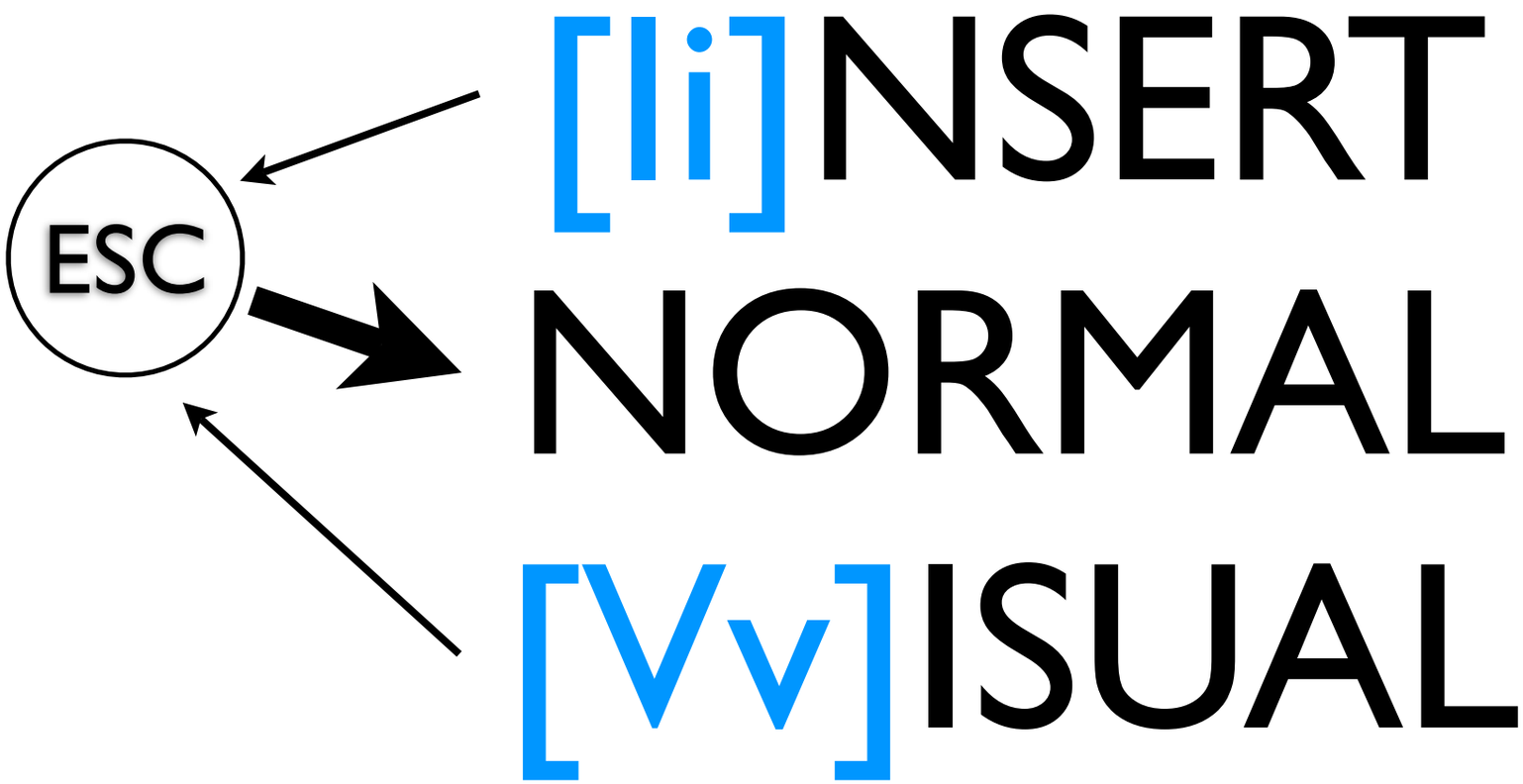
INSERT

NORMAL

VISUAL

SELECT

... et cetera


ESC
[i]NSERT
NORMAL
[Vv]ISUAL

1.1 Normal Mode

Motion

丟掉方向鍵吧

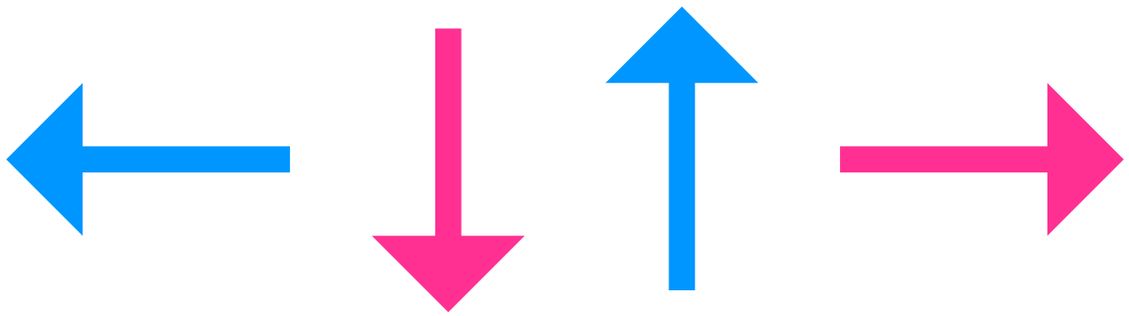
節省

移動

時間

請愛用

HJKL

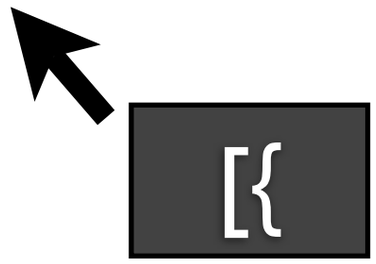


```
# comments ...

foreach my $foo ( @bar ) {

    # 做點事情

    my @outs = grep /pattern/ , @list;

}

sub func1 {


}
```

```
# comments ...

foreach my $foo ( @bar ) {

    # 做點事情

    my @outs = grep /pattern/ , @list;

}

sub func1 {


}
```

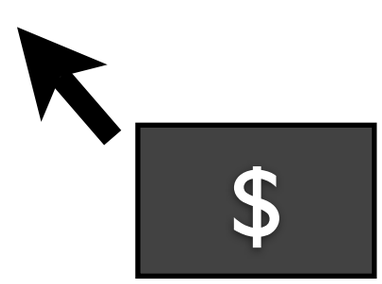
Cursor

```
# comments ...

foreach my $foo ( @bar ) {

    # 做點事情

    my @outs = grep /pattern/ , @list;

}

sub func1 {


}
```

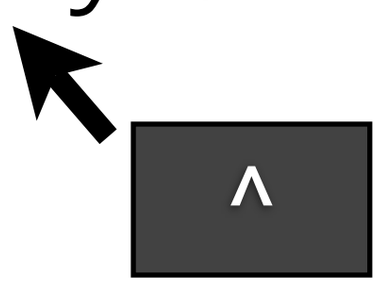


```
# comments ...

foreach my $foo ( @bar ) {

    # 做點事情

    my @outs = grep /pattern/ , @list;

}

sub func1 {


}
```

```
# comments ...

foreach my $foo ( @bar ) {

    # 做點事情

    my @outs = grep /pattern/ , @list;

}

sub func1 {


}
```



```
# comments ...

foreach my $foo ( @bar ) {

    # 做點事情

    my @outs = grep /pattern/ , @list;

}

sub func1 {


}
```



```
# comments ...
```

H

畫面最上方

```
foreach my $foo ( @bar ) {

    # 做點事情

    my @outs = grep /pattern/ , @list;

}

sub func1 {


}
```

```
# comments ...

foreach my $foo ( @bar ) {

    # 做點事情

    my @outs = grep /pattern/ , @list;

}

sub func1 {


}
```

M

畫面中間行

```
# comments ...

foreach my $foo ( @bar ) {

    # 做點事情

    my @outs = grep /pattern/ , @list;

}

sub func1 {


}
```

L

畫面最下方

```
# comments ...

foreach my $foo ( @bar ) {

    # 做點事情

    my @outs = grep /pattern/ , @list;

}

sub func1 {


}
```

修改至行尾

```
# comments ...

foreach my $foo ( @bar ) {

    # 做點事情

    my @outs = grep

}

sub func1 {


}
```

C

並進入 Insert mode

```
# comments ...

foreach my $foo ( @bar ) {

    # 做點事情

    my @outs = grep /pattern/ , @list;

}

sub func1 {


}
```

```
# comments ...

foreach my $foo ( @bar ) {

    # 做點事情

    my @outs = grep /pattern/ , @list;

}

sub func1 {

}
```

:h motion.txt

1.2 Insert Mode

Editing text

Insert Mode

- i : Insert text before the cursor

Insert Mode

- i : Insert text before the cursor
- I : Insert text before the first non-blank in the line

Insert Mode

- i : Insert text before the cursor
- I : Insert text before the first non-blank in the line
- a : Append text after the cursor

Insert Mode

- i : Insert text before the cursor
- I : Insert text before the first non-blank in the line
- a : Append text after the cursor
- A : Append text at the end of the line

1.3 Visual Mode

Select region

Visual Mode

- v : start Visual mode per character.

Visual Mode

- v : start Visual mode per character.
- V : start Visual mode linewise.

Visual Mode

- v : start Visual mode per character.
- V : start Visual mode linewise.
- Ctrl-v : start Visual mode blockwise.

Visual Mode

- v : start Visual mode per character.
- V : start Visual mode linewise.
- Ctrl-v : start Visual mode blockwise.

接上 operator 如 y (yank) , d (delete) , c (change) ... etc

2.語法標記

Syntax Highlight Support

```
/opt/local/share/vim/vim72
$ ls -1 syntax/ | wc -l
     520
```

包含五百多種語法設定檔

可算詞 Syntax

~/.vim/syntax/[filetype].vim

~/.vim/syntax/[filetype].vim

```
syn match [id] [re] [options]
```

~/.vim/syntax/[filetype].vim

```
syn match [id] [re] [options]
syn region [id] start=[re] end=[re]
```

~/.vim/syntax/[filetype].vim

```
syn match [id] [re] [options]
syn region [id] start=[re] end=[re]
syn keyword [id] [keyword] ...
```

~/.vim/syntax/[filetype].vim

```
syn match [id] [re] [options]
syn region [id] start=[re] end=[re]
syn keyword [id] [keyword] ...

hi [id]  guibg=[color] ctermfg=[color]
```

:help syntax.txt

3.編碼支援

ENCODING

```
:set fencs=utf-8,gbk,big5,euc-jp,utf-16le
```

```
:set fenc=utf-8 enc=utf-8 tenc=utf-8
```

設定檔案編碼清單

```
:set fencs=utf-8,gbk,big5,euc-jp,utf-16le
```

```
:set fenc=utf-8 enc=utf-8 tenc=utf-8
```

設定檔案編碼清單

Vim 會從第一組編碼開始讀取檔案，若是失敗則會跳下一組編碼讀取檔案

```
:set fencs=utf-8,gbk,big5,euc-jp,utf-16le
```

```
:set fenc=utf-8 enc=utf-8 tenc=utf-8
```

```
:set fencs=utf-8,gbk,big5,euc-jp,utf-16le
```

```
:set fenc=utf-8 enc=utf-8 tenc=utf-8
```

檔案編碼

```
:set fencs=utf-8,gbk,big5,euc-jp,utf-16le
```

```
:set fenc=utf-8 enc=utf-8 tenc=utf-8
```

內部編碼

```
:set fencs=utf-8,gbk,big5,euc-jp,utf-16le
```

```
:set fenc=utf-8 enc=utf-8 tenc=utf-8
```

Terminal 編碼

:help encoding

4. 快捷鍵

Key Mapping

```
:map
:nmap
:vmap
:imap
:smap
:xmap
```

... more

:map *(all)*

:nmap *(normal mode)*

:vmap *(visual mode)*

:imap *(insert mode)*

:smap *(select mode)*

:xmap *(visual , select mode)*

... more

:map *(all)*

:nmap *(normal mode)* ★ 最常用

:vmap *(visual mode)* ★ 最常用

:imap *(insert mode)* ★ 最常用

:smap *(select mode)*

:xmap *(visual , select mode)*

... more

```
:nmap <C-c><C-c> :!gcc -Wall % -o %:r.out<CR>
```

Ctrl C　　Ctrl C

呼叫 GCC 編譯現在編輯的檔案
並將輸出檔名命名為現在檔案名稱並取代副檔名為 .out

Normal Mode 時，可用 Tab 及 Shift-Tab 做縮排

```
nmap        <tab>    v>
nmap        <s-tab>  v<
```

```
vmap        <tab>    >gv
vmap        <s-tab>  <gv
```

```
nmap        <tab>   v>
nmap        <s-tab> v<

vmap        <tab>   >gv
vmap        <s-tab> <gv
```

Visual/Select Mode 時，也可用 Tab 及 Shift-Tab 做縮排

```
nmap        <tab>   v>
nmap        <s-tab> v<

vmap        <tab
vmap        <s-t
```

When mappings and menus are defined with the |:vmap| or |:vmenu| command they work both in Visual mode and in Select mode. When these are used in Select mode Vim automatically switches to Visual mode, so that the same behavior as in Visual mode is effective.

Visual/Select Mode 時，也可用 Tab 及 Shift-Tab 做縮排

```
imap <F2> <C-R>=strftime("%c")<CR>
```

在 Insert Mode 時，按下 F2 會插入時間戳記

```
cmap    <c-a>    <home>
cmap    <c-e>    <end>
cnoremap  <c-b>       <left>
cnoremap  <c-d>       <del>
cnoremap  <c-f>       <right>
cnoremap  <c-n>       <down>
cnoremap  <c-p>       <up>

cnoremap  <esc><c-b> <s-left>
cnoremap  <esc><c-f> <s-right>
```

讓 Command Mode 也有 Bash 的 Key Binding

:h map.txt

5.文字物件

TEXT OBJECT

Text Object

- word
- string
- paragraph
- block

⟹ action
(yank,delete,change ...etc)

Operator Mapping

v | c | d i | a { | [| (| “ | ‘

visual
change
delete

Operator

Inner Object
An Object

Region
{ }
[ ]
( )
“ “
‘ ‘

va{

```
char **aliases;
int n_aliases;
int j;
PangoFont *font;

pango_lookup_aliases (family,
          &aliases,
          &n_aliases);

if (n_aliases)
  {
    for (j = 0; j < n_aliases; j++)
  {
    pango_font_description_set_family_static (desc, aliases[j]);
    font = pango_font_map_load_font (fontmap, context, desc);
    if (font)
      pango_fontset_simple_append (fonts, font);
  }
  }
else
  {
    pango_font_description_set_family_static (desc, family);
    font = pango_font_map_load_font (fontmap, context, desc);
    if (font)
   pango_fontset_simple_append (fonts, font);
  }
}

static PangoFontset *
pango_font_map_real_load_fontset (PangoFontMap               *fontmap,
                  PangoContext               *context,
                  const PangoFontDescription *desc,
                  PangoLanguage              *language)
{
  PangoFontDescription *tmp_desc = pango_font_description_copy_static (desc);
  const char *family;
```

```
char **aliases;
int n_aliases;
int j;
PangoFont *font;

pango_lookup_aliases (family,
          &aliases,
          &n_aliases);

if (n_aliases)
  {
    for (j = 0; j < n_aliases; j++)
  {
    pango_font_description_set_family_static (desc, aliases[j]);
    font = pango_font_map_load_font (fontmap, context, desc);
    if (font)
      pango_fontset_simple_append (fonts, font);
  }
    }
  else
    {
      pango_font_description_set_family_static (desc, family);
      font = pango_font_map_load_font (fontmap, context, desc);
      if (font)
    pango_fontset_simple_append (fonts, font);
    }
}

static PangoFontset *
pango_font_map_real_load_fontset (PangoFontMap               *fontmap,
                  PangoContext               *context,
                  const PangoFontDescription *desc,
                  PangoLanguage              *language)
{
  PangoFontDescription *tmp_desc = pango_font_description_copy_static (desc);
```

function (blah , blah)

ci(

function ( )

function (new_args)

“Hello World”

di”

“”

“Hello New World”

6.分頁支援

TAB PAPGES

l/Jifty.pm l/J/M/Tutorial_zhtw.pod l/J/M/Actions.pod

```
=encoding utf8

=head1 NAME

Jifty::Manual::Tutorial - Jifty 從零開始

=head1 DESCRIPTION

這份教學文件將提供建構第一個 jifty 應用程式所需要的技巧。

=cut

=head1 HOW TO

=head2 The requirements
```

:tabnew

```
:tabnew
:tabedit path/to/file
```

```
:tabnew
:tabedit path/to/file
:tabfind path/to/file
```

:help tabpage.txt

7.程式碼折疊

FOLDS

FOLDIS

```
346
347 +---  5 lines: =head2 web----------------------------------------------
352
353 +---  3 lines: sub web {-------------------------------------------------
356
357 +---  5 lines: =head2 subs----------------------------------------------
362
363 +---  3 lines: sub subs {------------------------------------------------
366
367 +---  5 lines: =head2 bus-----------------------------------------------
372
373 sub bus {
374     my $class = shift;
375     my %args = ( connect => 1, @_ );
376     if (not $PUB_SUB and $args{connect}) {
377         my @args;
378
379         my $backend = Jifty->config->framework('PubSub')->{'Backend'};
380         if ( $backend eq 'Memcached' ) {
381             require IO::Socket::INET;
382
383             # If there's a running memcached on the default port. this
384             if ( IO::Socket::INET->new('127.0.0.1:11211') ) {
385                 @args = ( Jifty->app_instance_id );
386             } else {
387                 $backend = 'JiftyDBI';
388             }
389         }
```

Fold Methods

Fold Methods

Syntax Fold

Fold Methods

Syntax Fold

```
:set foldmethod=syntax
```

以程式語言語法做為折疊規則。
某些特定的設定可以參考
$VIMRUNTIME/syntax/*.vim

以 Perl 為例:

/opt/local/share/vim/vim72/syntax/perl.vim

```
if exists("perl_want_scope_in_variables")
“  .....
if exists("perl_extended_vars")
“  .....
  if exists("perl_fold")
“  .....
```

便可在 .vimrc 內設定啟用這些特定的語法標記或折疊 (Fold)

```
let perl_fold = 1
let perl_extended_vars = 1
“ .... etc
```

Fold Methods

- Syntax Fold
- Marker Fold

Fold Methods

Syntax Fold
Marker Fold

```
:set foldmethod=marker
```

以特定標記作為折疊規則，
預設是“{{{“ 以及 “}}}”

```
#  fold this {{{
function do_something {

    echo “COSCUP”;

}
#  }}}
```

Fold Methods

- Syntax Fold
- Marker Fold
- Indent Fold

Fold Methods

Syntax Fold
Marker Fold
Indent Fold

```
:set foldmethod=indent
```

以縮排作為折疊規則

Fold Methods

- Syntax Fold
- Marker Fold
- Indent Fold
- Manual Fold

```
:set foldmethod=manual
```

手動建立折疊區塊，
以 z f 鍵建立。

```
autocmd  BufWinLeave *.*  silent mkview
autocmd  BufWinEnter *.*  silent loadview
```

利用 autocmd 加上 mkview , loadview 來儲存手動建立的折疊區塊，儲存的折疊會被放在 ~/.vim/view/ 裡頭。

Fold Methods

- Syntax Fold
- Marker Fold
- Indent Fold
- Manual Fold
- Expr Fold *(Custom Fold Function)*

```
:set foldexpr=MyFoldLevel(v:lnum)
```

呼叫自訂的折疊函式

Fold Methods

- Syntax Fold
- Marker Fold
- Indent Fold
- Manual Fold
- Expr Fold *(Custom Fold Function)*
- Diff Fold

8.Modeline & FileType

ModeLine

```
" vim:fdm=marker:sw=2:ts=2:et:fdl=0:
"============================================
" Author:   Cornelius <
cornelius.howl{at}gmail{dot}com >
```

```
" vim:fdm=marker:sw=2:ts=2:et:fdl=0:
"============================================
" Author:   Cornelius <
cornelius.howl{at}gmail{dot}com >
```

開啟此檔時，使用 Marker 作為折疊，
縮排為兩個空隔，將 Tab 延展為空白，
折疊所有的 Fold (Level 0 以上)

File Type

~/.vim/ftplugin/ruby.vim

針對某種檔案類型的設定
不一定都要寫在 vimrc 內

~/.vim/filetype.vim

可在 filetype.vim 內定義新的檔案類型。

9.格式化

FORMATTING

程式碼很亂，怎麼辦？

要手動自己排嗎？

```
:set equalprg=perltidy
```

```
:set equalprg=perltidy
```

可針對特定區塊重新排列程式碼

選擇文字區塊後，按下 “=” 鍵，可使用外部的 perltidy 程式來做格式化處理。

```
autocmd Filetype perl :set equalprg=perltidy
```

可使用 autocmd 針對檔案類型做
equalprg 設定

```
autocmd Filetype c :set equalprg=indent
```

開啟 C 程式碼時，使用外部的
GNU Ident 程式來做格式化處理。

SQL

```
$ cat bin/sql-beautify.pl
#!/usr/bin/env perl
use warnings;
use strict;
use SQL::Beautify;
local $/;
my $sql = SQL::Beautify->new( query => <STDIN> , spaces =>
4 , break => "\n" );
print $sql->beautify;
```

```
autocmd Filetype sql :set equalprg=sql-beautify.pl
```

開啟 SQL 程式碼時，使用外部的 Filter 來做格式化處理。

10. QuickFix

:grep & :make

```
:grep [pattern] [filepath]
```

:grep [pattern] [filepath]

呼叫 grepprg (預設 vimgrep) 執行，並將 grep 的結果彙整至 QuickFix

```
59
60      /* Don't overwrite an existing error. This preserves the first
61       * error, which is the most significant. */
62      _cairo_status_set_error (&font_face->status, status);
63
64      return _cairo_error (status);
65  }
66
67  void
68  _cairo_font_face_init (cairo_font_face_t               *font_face,
69                         const cairo_font_face_backend_t *backend)
70  {
71      CAIRO_MUTEX_INITIALIZE ();
72
73      font_face->status = CAIRO_STATUS_SUCCESS;
74      CAIRO_REFERENCE_COUNT_INIT (&font_face->ref_count, 1);
75      font_face->backend = backend;
```

```
cairo-font-face.c                                                    59,0-1
 5 cairo-font-face.c|69| const cairo_font_face_backend_t *backend)
 6 cairo-font-face.c|81| * cairo_font_face_reference:
 7 cairo-font-face.c|82| * @font_face: a #cairo_font_face_t, (may be %NULL in which case this
 8 cairo-font-face.c|87| * cairo_font_face_destroy() is made.
 9 cairo-font-face.c|89| * The number of references to a #cairo_font_face_t can be get using
10 cairo-font-face.c|90| * cairo_font_face_get_reference_count().
11 cairo-font-face.c|92| * Return value: the referenced #cairo_font_face_t.
12 cairo-font-face.c|94| cairo_font_face_t *
13 cairo-font-face.c|95| cairo_font_face_reference (cairo_font_face_t *font_face)
14 cairo-font-face.c|109| slim_hidden_def (cairo_font_face_r
[Quickfix List]
```

QuickFix Window

```
:set grepprg=/path/to/grep
```

設置 grepprg (預設 VIM 內建 grep)

:make

:make

呼叫 makeprg (預設 make) 執行 Makefile. 並以該語言的 compiler output parser 彙整結果。

```
:set makeprg=gmake
```

設置 makeprg (預設 make)

Result ⇛ QuickFix Window

:copen

開啟 QuickFix Window

:cclose

關閉 QuickFix Window

:cnext

下一筆 Result

:cprevious

上一筆 Result

QuickFix Window Toggle

```
com! -bang -nargs=? QFix cal QFixToggle(<bang>0)
fu! QFixToggle(forced)
  if exists("g:qfix_win") && a:forced == 0
    cclose
    unlet g:qfix_win
  else
    copen 10
    let g:qfix_win = bufnr("$")
  en
endf
nn      <leader>q :QFix<cr>
```

QuickFix Window Toggle

```
com! -bang -nargs=? QFix cal QFixToggle(<bang>0)
fu! QFixToggle(forced)
  if exists("g:qfix_win") && a:forced == 0
    cclose
    unlet g:qfix_win
  else
    copen 10
    let g:qfix_win = bufnr("$")
  en
endf
nn       <leader>q :QFix<cr>
```

nmap 至 “\q” , <leader> 預設為 “\” 鍵

11. 插件

Plugin

SnipMate

提供類似 TextMate 編輯器的
程式碼片段完成功能。

```
snippet cla class .. initialize .. end
    class ${1:`substitute(Filename(), '^.', '\u&', '')`}
        def initialize(${2:args})
            ${3}
        end

end
```

自訂常用程式碼樣板

DBExt.vim

資料庫輔助插件，支援
Oracle, Sybase, MSSQL ,
MySQL, DBI 等等

xml.vim

提供 XML 相關輔助功能，
如自動補完標籤等等。

FuzzyFinder.vim

Fuzzy/Partial pattern explorer

The_NERD_TREE.vim

樹狀目錄檔案瀏覽

The NERD Commenter

註解輔助工具

taglist.vim

透過 ctags 工具產生程式碼標籤
並可將 macro , function , variable 等資料整理出來

cscope

C 程式碼追蹤輔助

autocomplpop.vim

MRU

most recently used

bufexplorer.vim

Buffer Explorer

Rails.vim

git-vim

那麼，怎麼安裝呢？

最原始的作法

http://www.vim.org/index.php

nerd_commenter

CPAN Linux NewLinks PasteBin Work Tmp Do MailBox CG-WLBARGLV3 Plk MyExp

SPONSOR Vim development VOTE for features

the editor

BUY the Vim book HELP Uganda LEARN Vim

not logged in (login)

Google™ Custom Search

Search

Home
Advanced search

About Vim
Community
News
Sponsoring
Trivia
Documentation
Download

Scripts
Tips
My Account

Site Help

What is Vim?

Vim is a highly configurable text editor built to enable efficient text editing. It is an improved version of the vi editor distributed with most UNIX systems. Vim is distributed free as charityware. If you find Vim a useful addition to your life please consider helping needy children in Uganda.

News

Vim 7.2.245 is the current version

Visit to Kibaale

[2009-05-12] Vim comes for free. I do ask you to consider helping Vim's charity, please read on. In April I have visited the Kibaale Childrens Centre. You can read my report, with lots of pictures, here. The new clinic has become very popular. Every day many patients come here for medical help that would otherwise not be available to them. The Kibaale clinic saves lives! This popularity comes at a price. We struggle to pay for all the medicine, salaries of the medical staff, maintenance of equipment, etc. Therefore I have set up this page for sponsoring a room in the clinic. We need your help! (*Bram Moolenaar*)

Vim 2009 desktop calendar

[2008-12-21] The usual Vim desktop calendar is now available for download and printing: http://www.moolenaar.net/#Calendar. If you print it on thick paper you can fold it so that it stands on your desk. One side contains a useful 12-month calendar. On the other side there is brief information about ICCF-Holland, Vim and A-A-P. Happy Vimming in 2009! (*Bram Moolenaar*)

more news... Get a Vim poster DVD and video about Vim's charity project

Recent Script Updates

2,721 scripts, 3,524,650 downloads

[2009-08-13] textobj-entire : Text objects for entire buffer
(0.0.1) - Fix a typo on the name of a variable. - *Kana Natsuno*
[2009-08-12] WuYe : A dark background color scheme
(1.2.1) The final version - *SAM Lee*
[2009-08-12] Screen (vim + gnu screen) : Start gnu screen w/ your vim session and a split shell + send commands to the sh
(0.4) - fixed usage in cygwin, which the previous version broke - fixed handling of large amounts of text sent through :ScreenSend (thanks to Tobias Wolf for

Search for Vim Scripts

on the web and the Vim website:

Google Code Search

You can use a regexp pattern, e.g., go{2}gle, h[ea]llo. (more about regexp patterns)

on the Vim website only

keywords nerd

type -- all --

sort by Rating Descending search cancel

Search Results

Searched scripts for "**nerd**"

Showing **1** to **3** of **3** results

Script	Type	Rating	Down loads	Summary
The NERD tree	utility	2095	24700	A tree explorer plugin for navigating the filesystem
Source Explorer (srcexpl.vim)	utility	903	4145	A Source code Explorer based on tags works like context window in Source Insight
trinity.vim	utility	98	1480	Build the trinity of srcexpl, taglist, NERD_tree to be a good IDE

prev | next

Showing **1** to **3** of **3** results

The NERD tree : A tree explorer plugin for navigating the filesystem

script karma	Rating **2107/596**, Downloaded by 24786

created by
Marty Grenfell

script type
utility

description
Grab the latest dev version from github: https://github.com/scrooloose/nerdtree.

What is this "NERD tree"??

Check out this demo http://www.flickr.com/photos/30496122@N07/2862367534/sizes

The NERD tree allows you to explore your filesystem and to open files and directories. It presents the filesystem to you in the form of a tree which you

package	script version	date	Vim version	user	release notes
NERD_tree.zip	**3.1.1**	*2009-06-07*	7.0	*Marty Grenfell*	- fix a bug where a non-listed no-name buffer was getting created every time the tree window was created, thanks to Derek Wyatt and owen1 - make <CR> behave the same as the 'o' mapping - some helptag fixes in the doc, thanks strull - fix an error when using :set nohidden and opening a file where the previous buf was modified. Thanks iElectric

下載套件

install details
NOTE: In version 2.0.0 the script file and help file were renamed to NERD_commenter.vim and NERD_commenter.txt.
If you are upgrading from version < 2.0.0 to version >= 2.0.0 then you must delete the old files NERD_comments.vim and
NERD_comments.txt.

Stick the NERD_comments.vim in ~/.vim/plugin and NERD_comments.txt in ~/.vim/doc.

Run :helptags ~/.vim/doc.

Restart vim.

Go :help NERD_commenter.txt to read the help file.

閱讀安裝步驟

install details
NOTE: In version 2.0.0 the script file and help file were renamed to NERD_commenter.vim and NERD_commenter.txt.
If you are upgrading from version < 2.0.0 to version >= 2.0.0 then you must delete the old files NERD_comments.vim and
NERD_comments.txt.

Stick the NERD_comments.vim in ~/.vim/plugin and NERD_comments.txt in ~/.vim/doc.

Run :helptags ~/.vim/doc.

Restart vim.

Go :help NERD_commenter.txt to read the help file.

閱讀安裝步驟

太繁瑣!

太繁瑣!

Vimana

Vim script Manager

- Vimball
- Archive File (zip , rar)
- .vim file

POWERED BY

Perl

使用 CPAN 安裝 Vimana

```
$ cpan Vimana
```

搜尋

```
$ vimana search xml
```

```
$ vimana search xml
rrd.vim                - Edit RRD data with Vim.
qt.vim                 - tiny tool for the uic used in Qt from
Trolltech
syntax-for-xul         - Highlighting for XML User interface Language.
maven2.vim             - Compiler plugin for maven2
 .... skip
```

查詢 script 資訊

```
$ vimana info xml.vim
```

安裝 xml.vim 外掛

```
$ vimana install xml.vim
```

```
$ vimana install xml.vim
$ vimana install rails.vim
$ vimana install the-nerd-tree.vim
$ vimana install taglist.vim
$ vimana install snipmate
$ vimana install fuzzyfinder.vim
etc ...
```

ALL Works

Git Repository

http://github.com/c9s/Vimana/tree/master

ENJOY

Thank You

Demo

如果有時間....

Q & A