

Pymacs 初體驗 0.0.1

PyConTW 2013

Superbil <superbil@gmail.com>

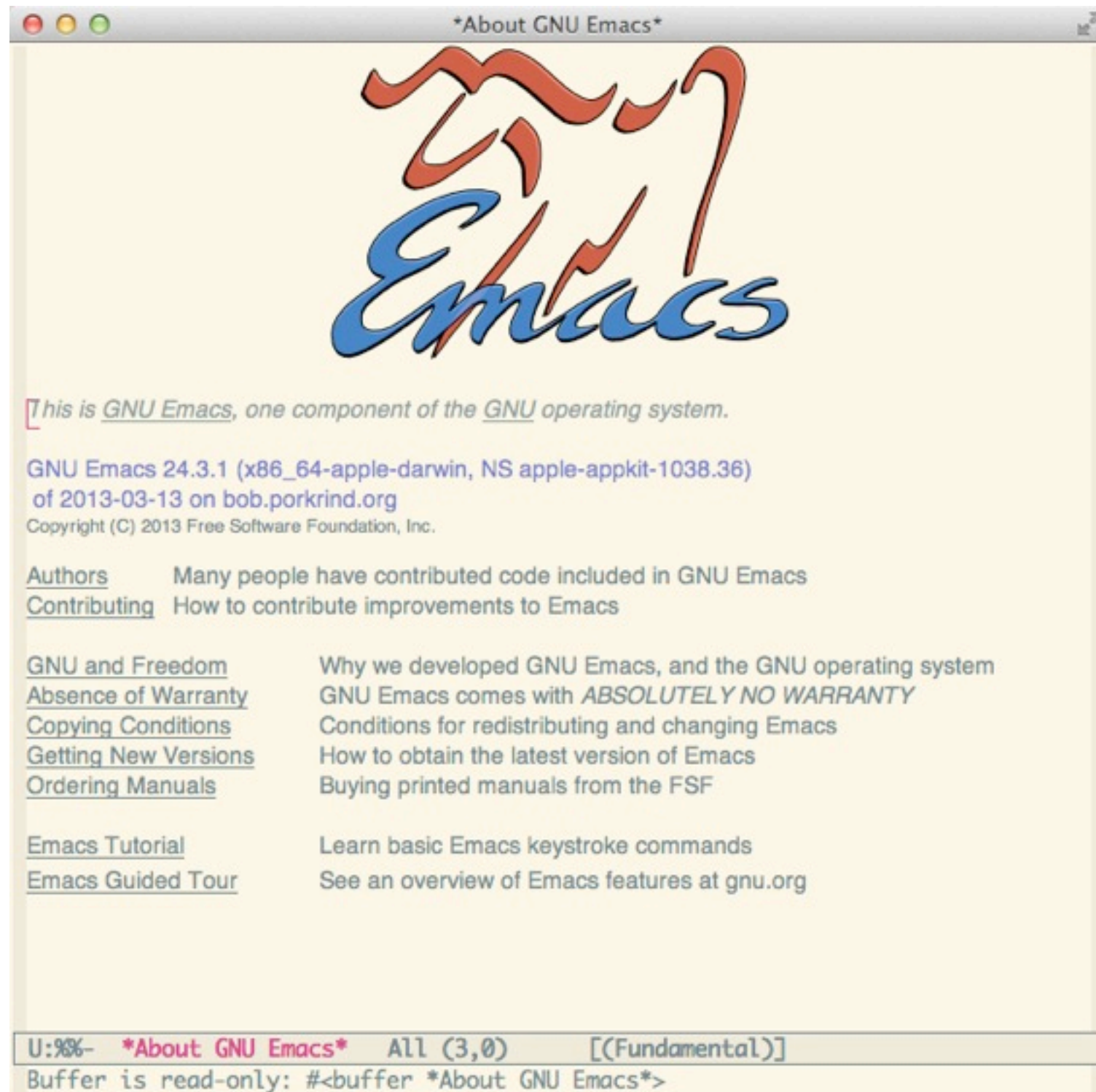
About Me

- 軟體工程師、Mobile App 開發
- Emacs 使用經驗大約 1 年
- Python 使用經驗大約 1 年
- PyHUG 主持人
- about.me/superbil
- freenode #python.tw #emacs.tw
on RPi

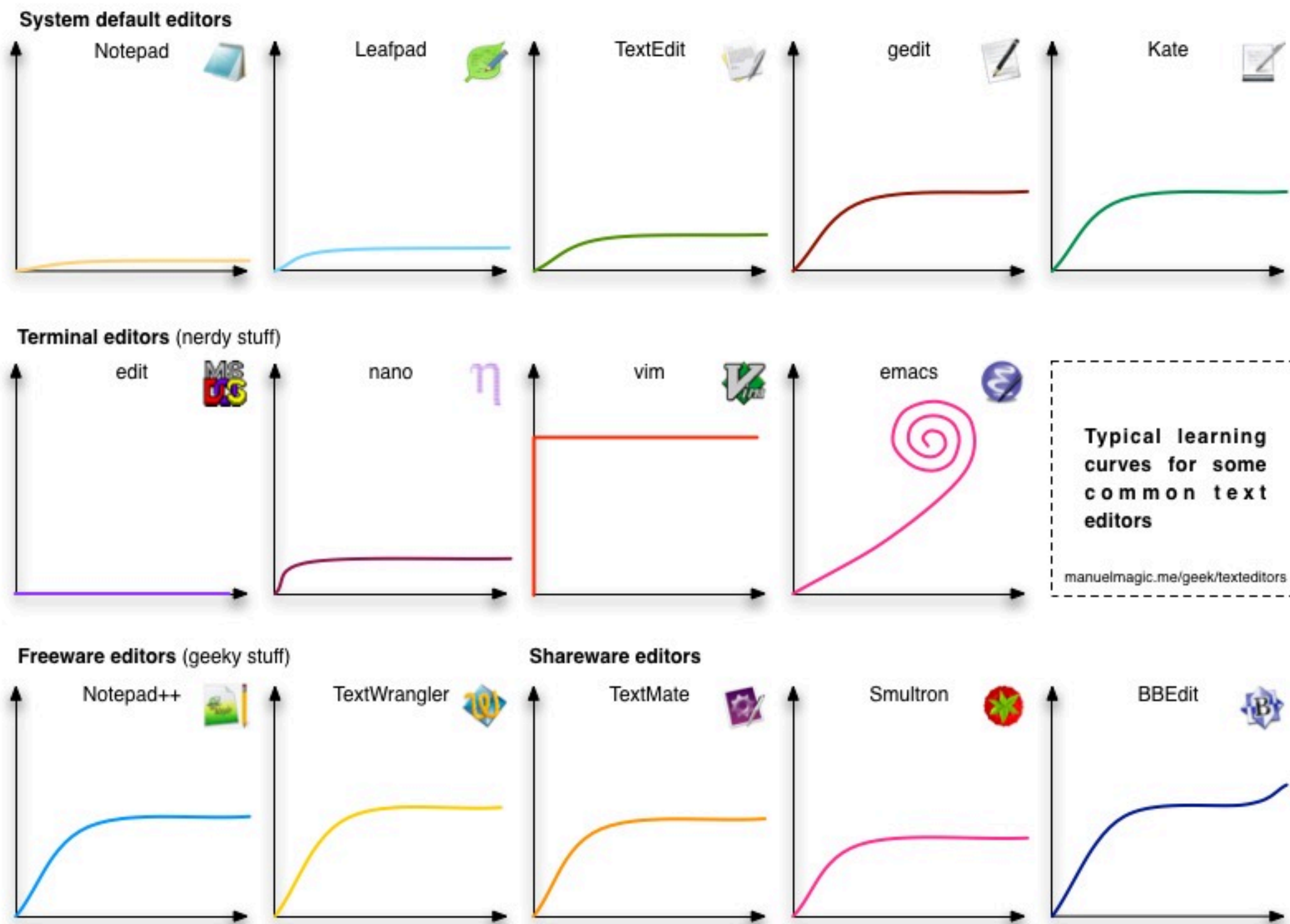
什麼是 Emacs ?



編輯器



編輯器學習曲線



為什麼要用 Emacs ?!

You can hack it !

你不用把手從鍵盤上移開

What is a hacker ?

- hacker 是那些勇於挑戰既有體制，改變系統讓世界更進步的玩家，決定一個人是否為 hacker 的關鍵在於思維。hacker 其實是一種思考模式

羅根·拉普蘭特：用駭客思維學習

Logan LaPlante: Hackschooling Makes Me Happy

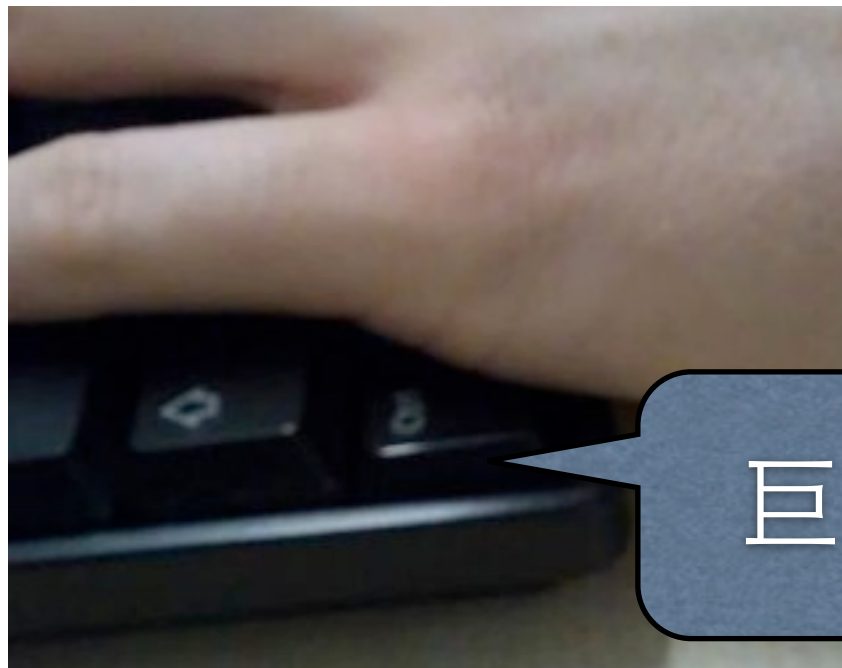
Emacs 基本介紹

- Ctrl + h t // emacs tutorial
- Ctrl + {b,n,p,f} // h j k l
- Ctrl + x b // 切換 buffer
- Ctrl + x Ctrl + e // 執行 () 中的 elisp
- Ctrl + h m // 查詢當下 mode 的文件

Emacs 的熱鍵很難按

更換你的 Ctrl 鍵位置

- 你記得你上次用到 Caps Lock 是什麼時候嗎？
- Ctrl 在下面的位置其實是**非常**難按的



巨人出現了阿!!!

Emacs vs Vim

Emacs vs Vim vs Sublime Text

Emacs vs Vim vs Sublime Text vs More...

Emacs vs Vim

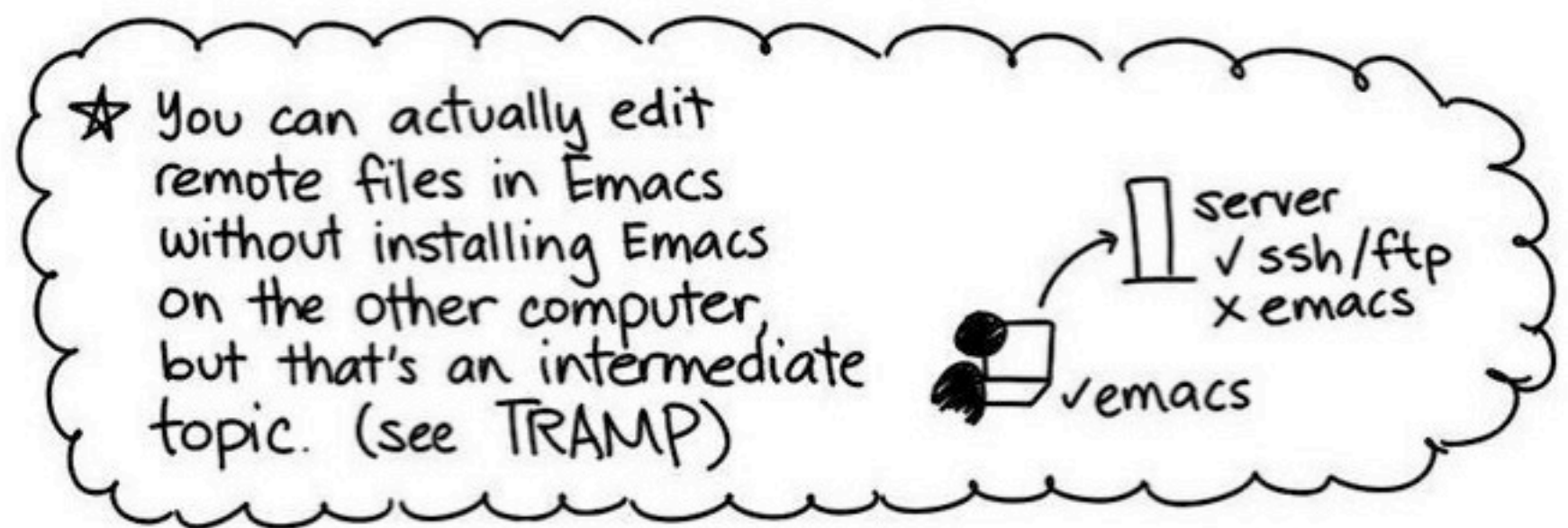
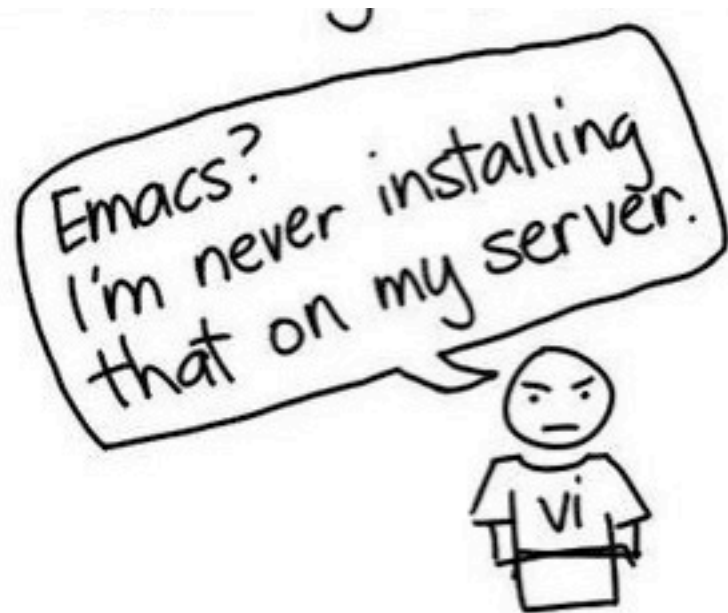
- 其實這是個陳年老問題...
- 沒有好或不好，只有合不合用

為何不同時使用 Emacs 和 Vim ?

同時用 **Vim** 和
Emacs，这人早
被烧死了

by 程序员时髦值指南

Install Emacs on Server ?



為什麼要用 Pymacs ?

因為沒有人會寫 **lisp**
emacs lisp

因為沒有人會寫 lisp

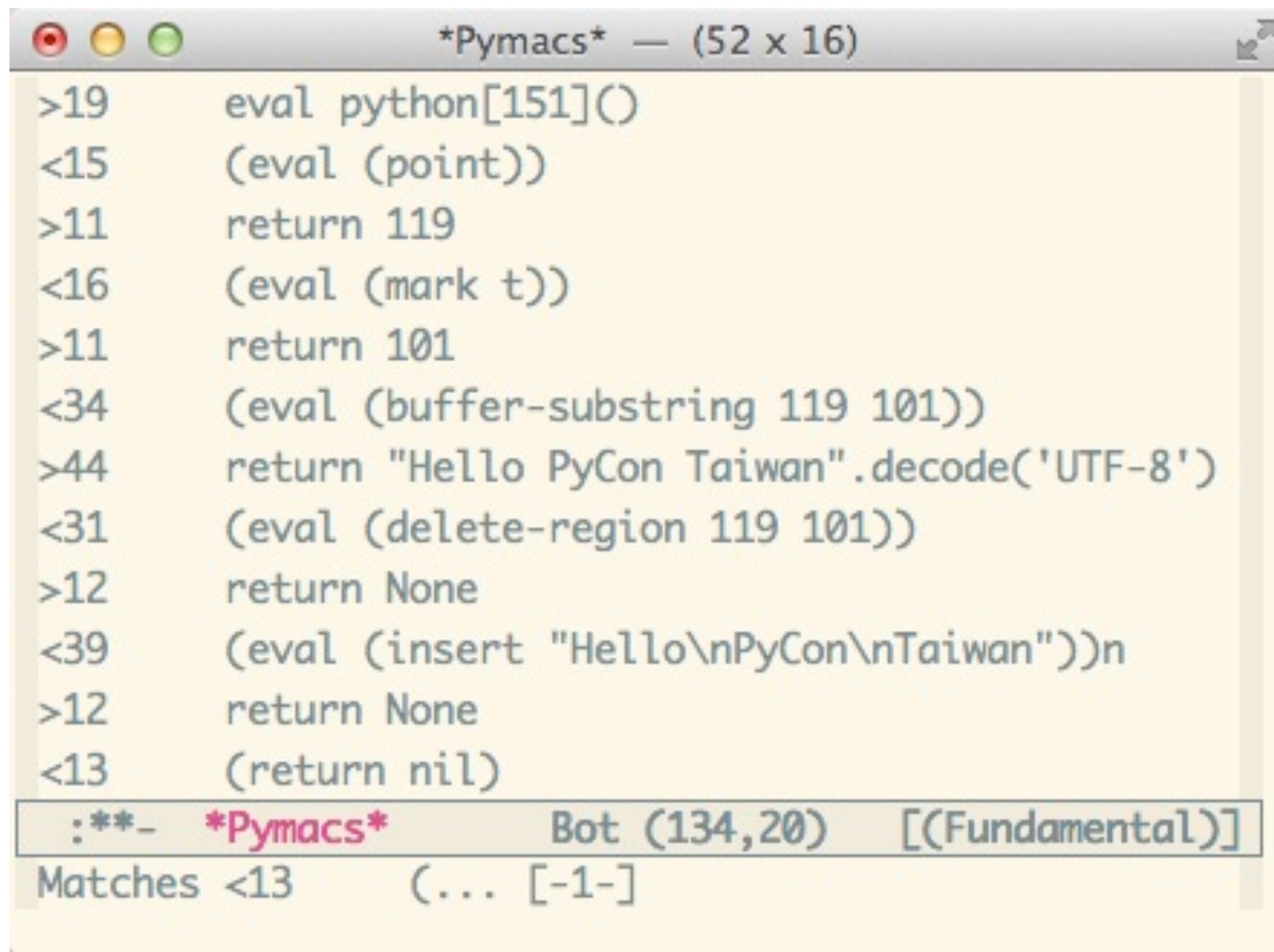
- 想要 hack Emacs 又不想學 lisp
你就可以用 Pymacs

How to install Pymacs

- Python side
 - make install
- Emacs side
 - M-x package-install RET pymacs RET

Emacs 24.x +

Pymacs Debug



```
*Pymacs* — (52 x 16)

>19      eval python[151]()
<15      (eval (point))
>11      return 119
<16      (eval (mark t))
>11      return 101
<34      (eval (buffer-substring 119 101))
>44      return "Hello PyCon Taiwan".decode('UTF-8')
<31      (eval (delete-region 119 101))
>12      return None
<39      (eval (insert "Hello\nPyCon\nTaiwan"))n
>12      return None
<13      (return nil)

:**- *Pymacs*      Bot (134,20)  [(Fundamental)]
Matches <13      (... [-1-])
```

要切換到 ***Pymacs*** 這個 Buffer

Pymacs 運作原理

- 建立 server process 跑到 loop 等待
- 當接收到指令之後執行，並回傳到 emacs
- 實際上是遞迴

Pymacs 的簡單應用

- `pymacs-exec`
 - `(pymacs-exec "print 'Hello PyConTw'")`
 - 在 Python 的執行環境裡面執行程式碼
執行 python 的 statement，回傳值永遠都是 nil

Hello World

Python side

```
from Pymacs import lisp
interactions = {}

def hello_world():
    "Hello world' from Python."
    lisp.insert("Hello from Python!")
interactions[hello_world] = ''
```

Hello World

Emacs side

```
(pymacs-load "helloEmacs" "hm-")
```

- 讀取一個名為 helloEmacs 的套件，並使用 hm- 來當作前綴詞

產生出 lisp function

- (hm-hello-world)

hm-hello-world is an interactive Lisp function.

(hm-hello-world &rest ARGUMENTS)

It interfaces to a Python function.

Hello world' from Python.

M-x hm-hello-world

(hm-hello-world) Hello from Python!

Sample - winkler

```
from Pymacs import lisp
interactions = {}

def break_on_whitespace():
    start, end = lisp.point(), lisp.mark(True)
    words = lisp.buffer_substring(start,
end).split()
    lisp.delete_region(start, end)
    lisp.insert('\n'.join(words))

interactions[break_on_whitespace] = ''
```

Sample - winkler result

Hello PyCon Taiwan

Marked

M-x winkler-break-on-whitespace

Hello
PyCon
Taiwan

```
from Pymacs import lisp
interactions = {}

def break_on_whitespace():
    start, end = lisp.point(), lisp.mark(True)
    words = lisp.buffer_substring(start, end).split()
    lisp.delete_region(start, end)
    lisp.insert('\n'.join(words))

interactions[break_on_whitespace] = ''
```

Live Demo

到那邊問問題?

- [Pymacs 0.25](#)
- [Emacswiki.org](#)
- [stackoverflow.com](#)
- [Emacs on Google Plus](#)
- [freenode#emacs.tw](#)
- [planet.emacsen.org](#)



笑吧你。

現在每人都可以使用 Emacs 而減少開發時間。

下載 Emacs

更貼心。 完善的文件。
更精彩。 當個時髦的程序員。
更隨心。 模式切換 任我遨遊。

M-x

Happy Hacking

Keyboard

Thank you