Vim + Python-mode

Carlos Gustavo Ruiz
@atmantree

Agenda

- IDEs vs Editores
- Vi → Vim
- Vim
- Personalización,
  Scripts y Plugins
- Python-Mode
- Recursos

IDEs vs Editores de Texto

IDEs

- Integrado (todo incluido)
- Alto consumo de recursos
- Uso de Entorno Gráfico es indispensable
- Generalmente definen un estilo de hacer las cosas.

Editores de Texto

- Extensibles
- Bajo consumo de recursos
- Uso de Entorno Gráfico es opcional
- Normalmente no interviene en el estilo del hacer las cosas.

IDEs vs Editores de Texto

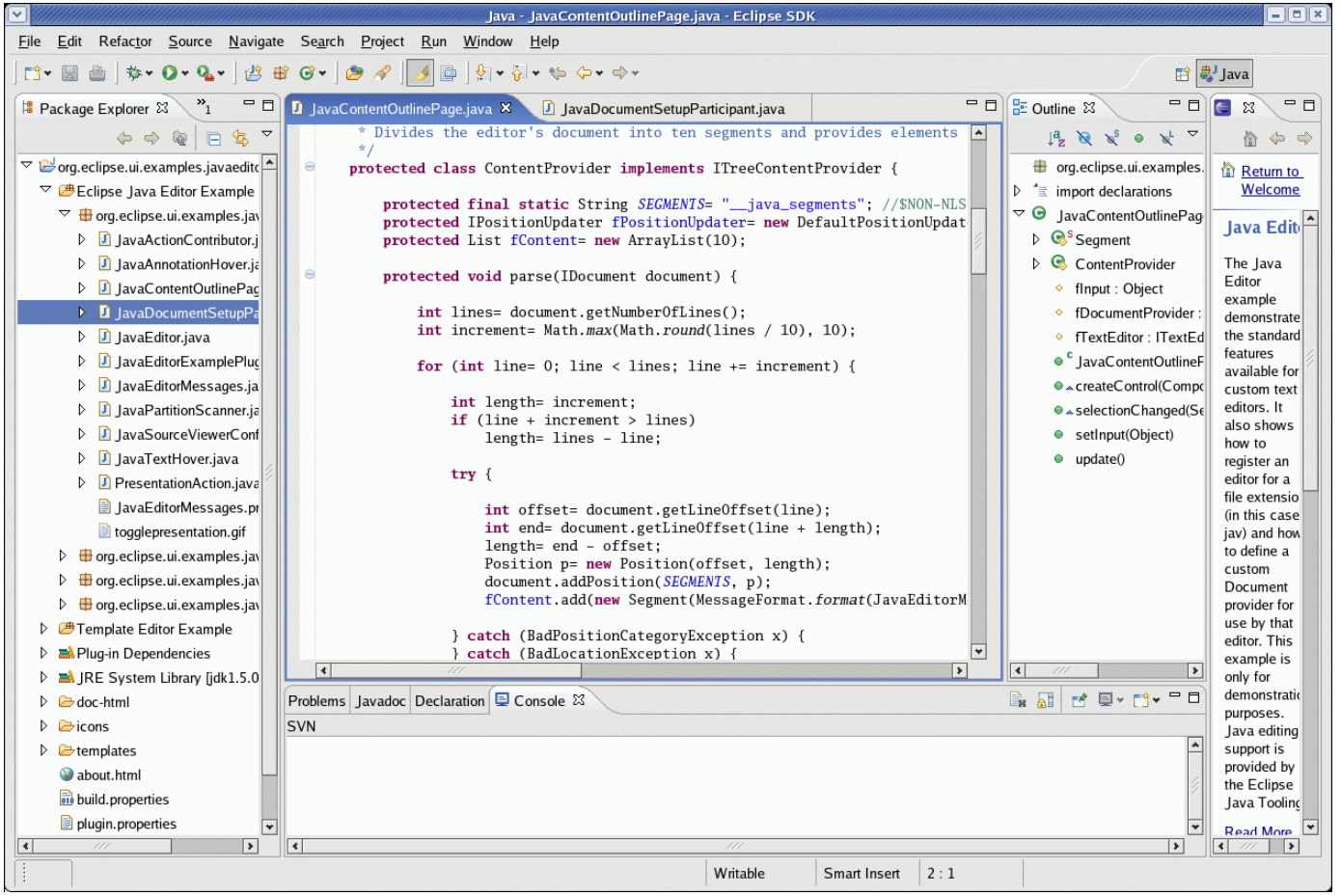
IDEs

- Eclipse
- NetBeans
- VisualStudio
- Xcode
- Qt Creator
- Anjuta

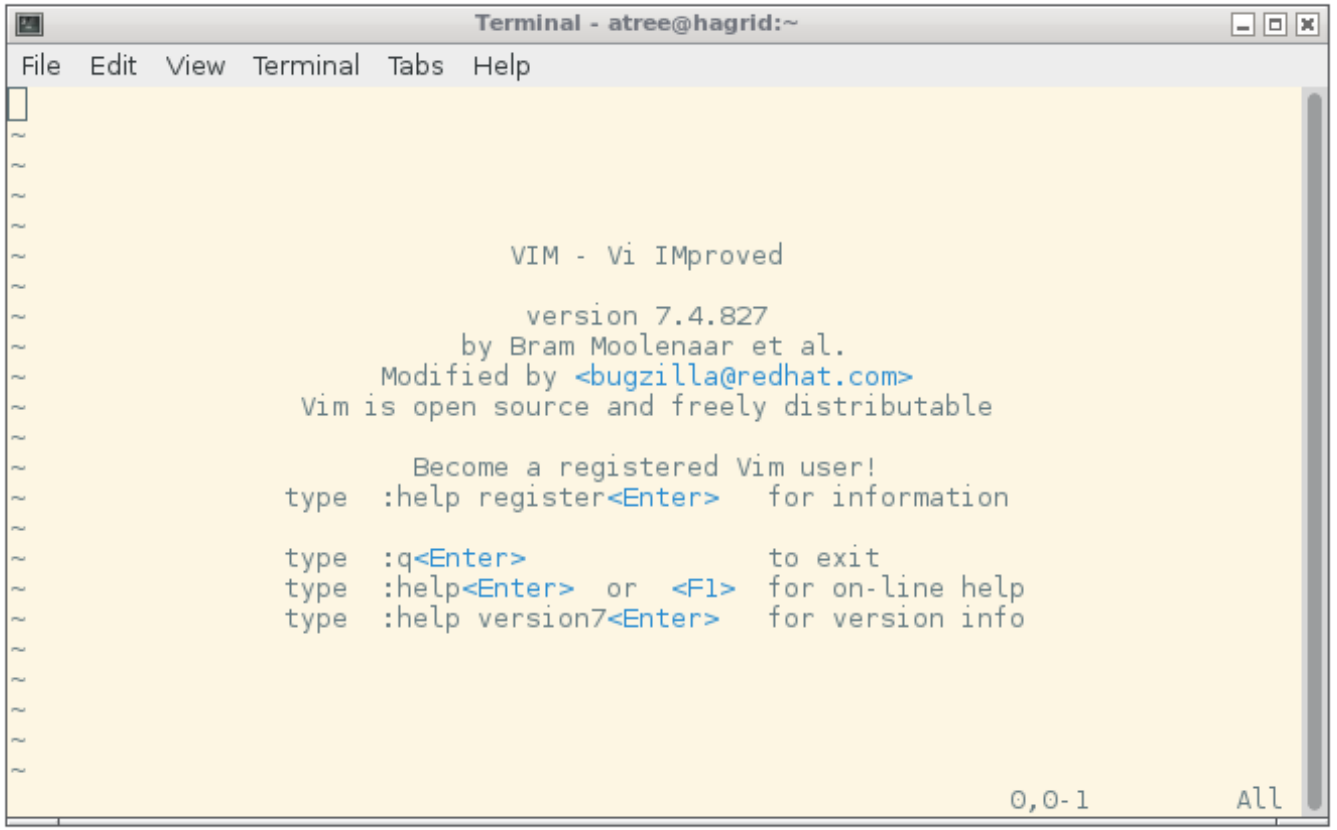
Editores de Texto

- Vim
- Emacs
- Notepad++
- SublimeText
- Atom
- nano

IDEs vs Editores de Texto

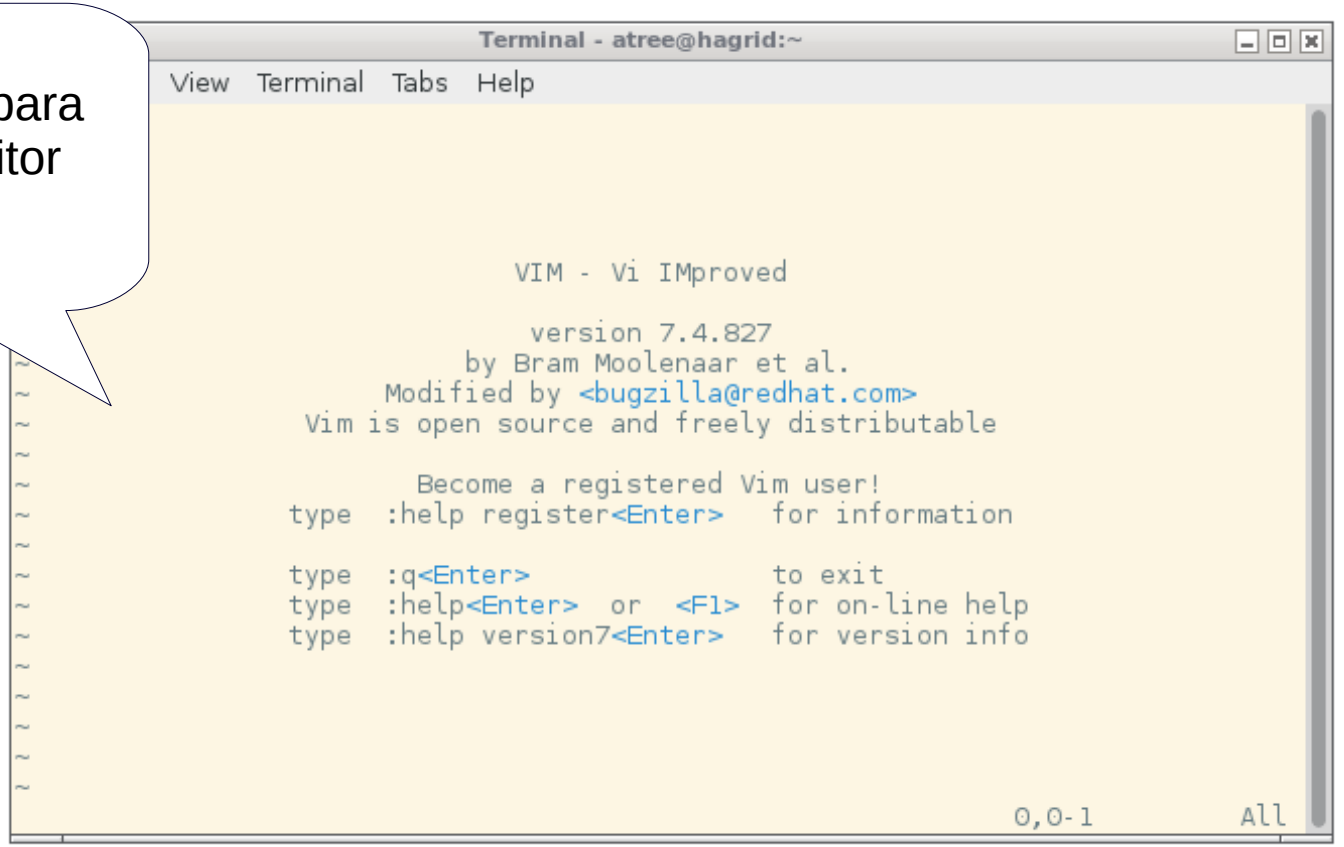


IDEs vs Editores de Texto

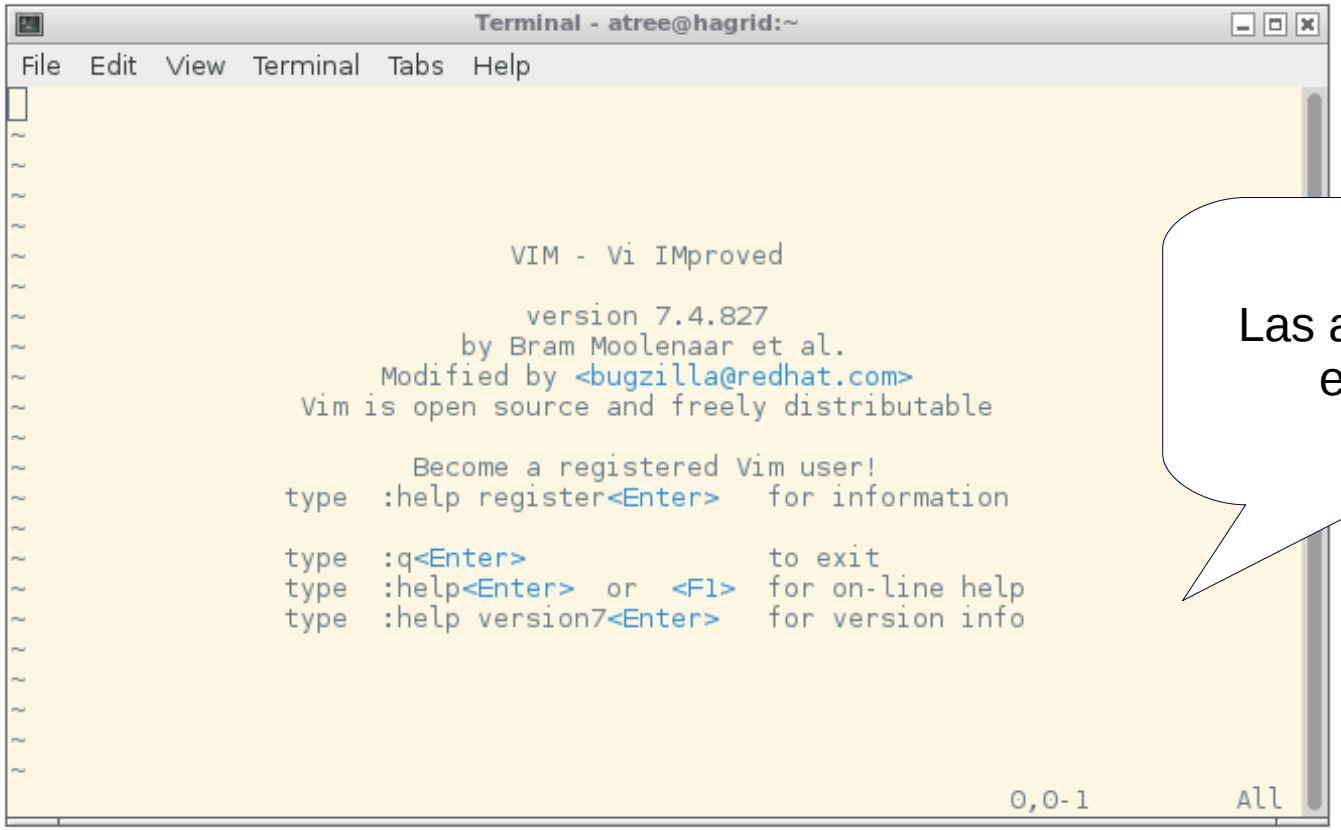


IDEs vs Editores de Texto

¿Una charla para un simple editor de texto?

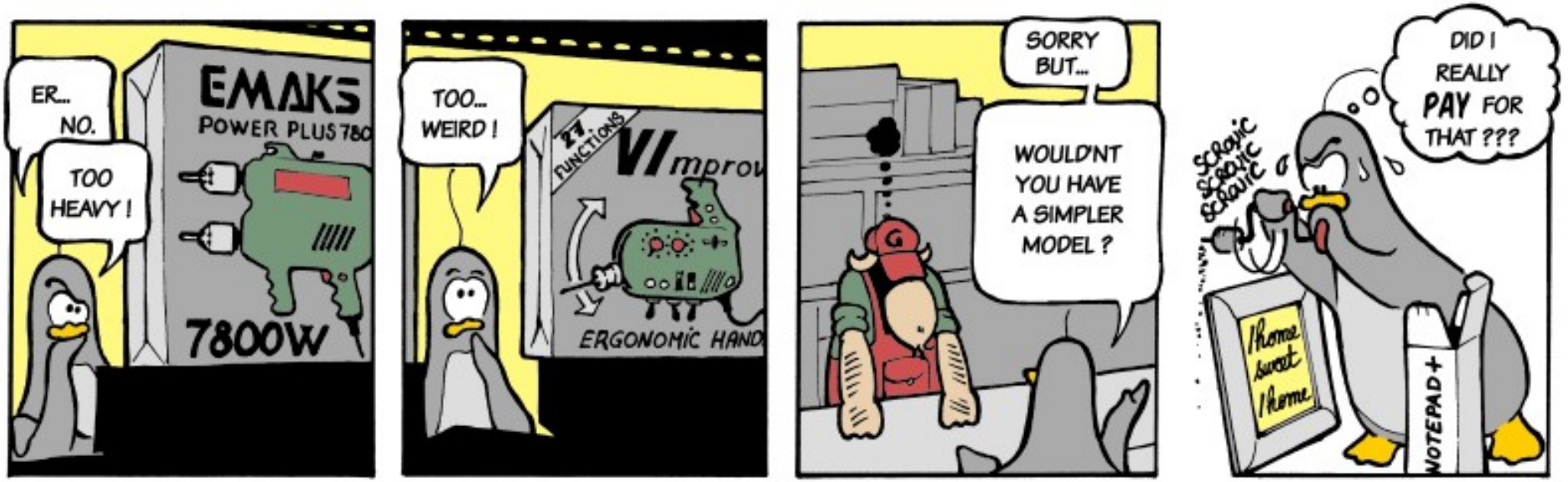


IDEs vs Editores de Texto



Las apariencias engañan

IDEs vs Editores de Texto



¿Por qué Vim?

- Vim no solo es un editor, es una herramienta.
- Es rápido.
- Muy flexible y altamente personalizable.
- Se ejecuta prácticamente en cualquier plataforma.
- Trabaja con muchísimos lenguajes de programación.
- Extensible (Macros, Scripts y Plugins).
- Es Código Abierto (Charityware)

Vi → Vim

- **vi** → 1976 (modo visual para editor ex, incluye el concepto de edición modal)
- **STEVIE** → 1987 (primer clon de vi, ofrecía un grupo limitado de opciones de vi, diseñado para Atari ST, luego portado a OS/2) *10% compatible*
- **Elvis** → 1997 (corregía errores de STEVIE, incluyo resaltado de sintaxis, multiples ventanas y una gui, era el vi estándar de Slackware y Minix en el 2003) *80% compatible*
- **nvi** → 2001 (Derivado de Elvis, multiples buffers, historial ilimitado, expresiones regulares, scripting con Perl y Tcl/tk, distribuido con los Unix BSD) *95% compatible*
- **Vim** → 1991 (Edición de multiples archivos en multiples buffers, ventanas y pestañas, lenguaje de scripting, soporte de Perl y Python, resaltado de código de más de 200 lenguajes de programación, historial ilimitado, expresiones regulares, completado de palabras de acuerdo al contexto, integración con multiples compiladores) *99% compatible*
- **Vile** → 1990 (Intento por crear un editor con lo mejor de vi y emacs, edición modal, lenguaje procedimental, soporte para perl – experimental, funciones pueden mapearse a teclas) *10% compatible*

Vim

- Edición Modal
  - Modo Normal (desplazamiento y manipulación de texto)
  - Modo Insert (modifica el texto)
  - Modo Visual (resalta una porción del archivo y la modifica en bloque)
  - Modo Comandos (ejecuta acciones en forma de comandos)
  - Modo Ex (comandos)

Vim

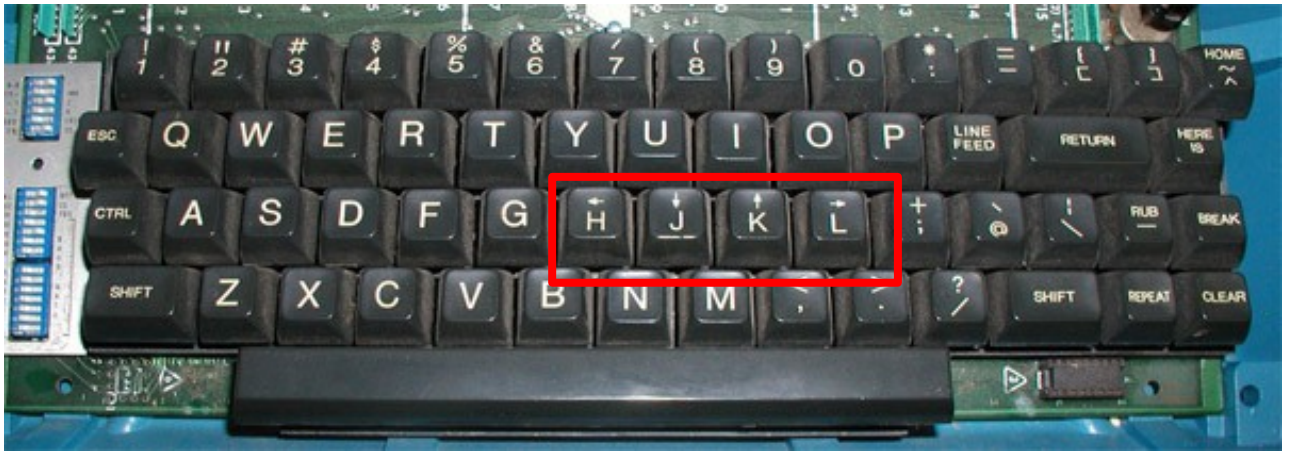
Modo normal

- Desplazamiento (h, j, k, l) (izquierda, arriba, abajo, derecha)

Vim

Modo normal

- Desplazamiento (h, j, k, l) (izquierda, abajo, arriba, derecha)



https://en.wikipedia.org/wiki/ADM-3A (un estándar de 1974)
http://xahlee.info/kbd/keyboard_hardware_and_key_choices.html

Vim

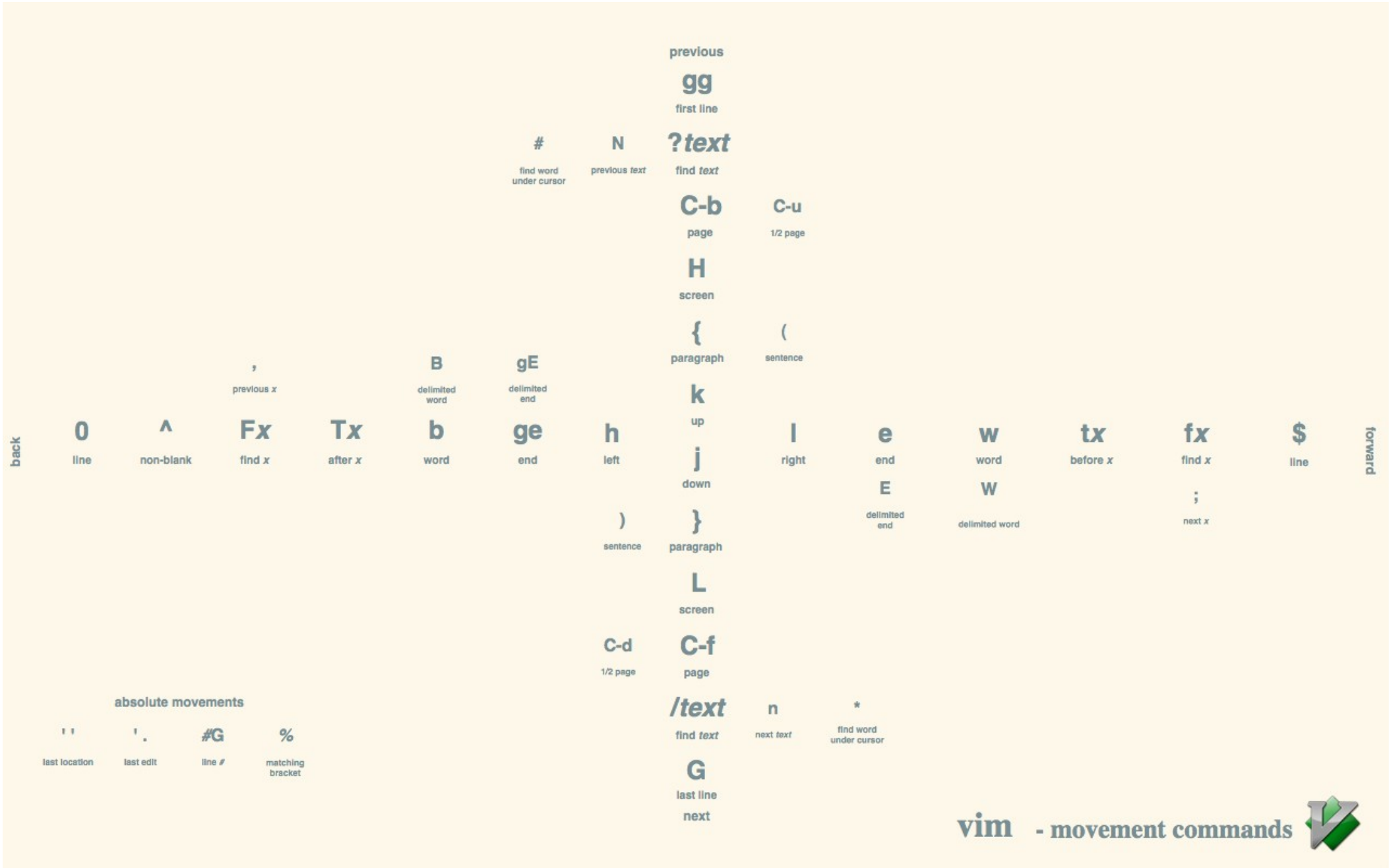
Modo normal

- Otros desplazamientos (algunos de los más comunes)
  - w → Avanzar palabra
  - W → Avanzar hasta el próximo carácter en blanco
  - b → Retroceder al inicio de la palabra
  - B → Retroceder hasta el carácter en blanco previo
  - e → Avanzar hasta el final de palabra
  - ^ → Inicio de línea
  - $ → Final de línea
  - /texto → ir a la vez que aparezca *texto*
  - J → Quitar salto de línea

Vim

Modo normal

- Otros desplazamientos (algunos de los más comunes)
  - ^E → Mover la ventana hacia abajo
  - ^Y → Mover la ventana hacia arriba
  - ^F → Bajar una página
  - ^B → Subir una página
  - H → Mover el cursor al tope de la ventana
  - M → Mover el cursor al medio de la ventana
  - L → Mover el cursor al fondo de la ventana
  - gg → Ir al inicio del archivo
  - G → Ir al final del archivo



http://www.sitepoint.com/getting-started-vim/

Vim

Objetos de texto

- w → palabras
- s → sentencias (oraciones)
- p → parrafos
- t → tags (etiquetas – xml/html)

Movimientos

- a → all (todos)
- i → in (dentro)
- t → until (hasta)
- f → buscar hacia adelante
- F → buscar en reversa

Comandos

- d → delete (borrar y/o cortar)
- c → change (borrar y cambiar a modo insert)
- y → yank (copiar)
- v → selección visual

Vim

Ahora juntemos lo anterior con esta formula

- [numero]{comando}{objeto o movimiento}

Ejemplos de acciones:

- diw → delete in word (borrar en palabra)
- ciw → change in word (igual que el anterior pero te cambia a modo insert)
- caw → cambiar toda la palabra
- yi) → copiar el contenido dentro de paréntesis
- di) → borrar el contenido dentro del paréntesis
- da) → igual que el anterior pero elimina el paréntesis inclusive

Vim

- Otros comandos:
  - dd→ borrar línea
  - yy → copiar línea
  - p→ pegar en la línea siguiente
  - P → pegar en la línea previa
  - D → borrar hasta el final de la línea
  - C → cambiar hasta el final de la línea
  - o → insertar linea después de la actual
  - O → insertar linea después de la actual
  - I → mover al inicio y pasar a modo edición
  - A → mover al final y pasar a modo edición
- Para que averiguar:
  - ¿De lo visto que hace *va”*?
  - ¿Qué hace el comando . (punto)?

Vim

Modo Insert

- Se usa para modificar texto como haría con un editor común y corriente

Modo visual

- Permite seleccionar un bloque y aplicar comandos y modificaciones por lote

Modo Comando

- Es todo lo que escribimos en los comandos con la fórmula :*comando*.
- Ejemplo:
  - :help
  - :vimtutor

Modo ex

- Parecido al modo comando, permite ejecutar instrucciones del comando ex:

```
vim -E -s Makefile <<-EOF
  :%substitute/CFLAGS = -g$/CFLAGS =-fPIC -DPIC -g/
  :%substitute/CFLAGS =$/CFLAGS =-fPIC -DPIC/
  :%substitute/ADAFLAGS =$/ADAFLAGS =-fPIC -DPIC/
  :update
  :quit
EOF
```

Personalización, Scripts y Plugins

- Una vez que se ha personalizado Vim, este queda como su chaqueta o el jean favorito.
  - Para personalizar Vim principalmente nos valemos de modificar el archivo *.vimrc*
  - En el se hace mapeado de atajos de teclado, cambiamos de esquema de colores, se corre rutinas y se define condiciones y sintaxis para diferentes tipos de archivos y un montón de cosas más.

Personalización, Scripts y Plugins

Vim es extensible vía Macros, Scripts y Plugins

- Las macros permiten simplificar rutinas simples durante la edición
- Los scripts en VimL permiten automatizar tareas más complejas y eventualmente escribir..
- Plugins, que es de lo más genial una vez que se comprenden los conceptos básicos.

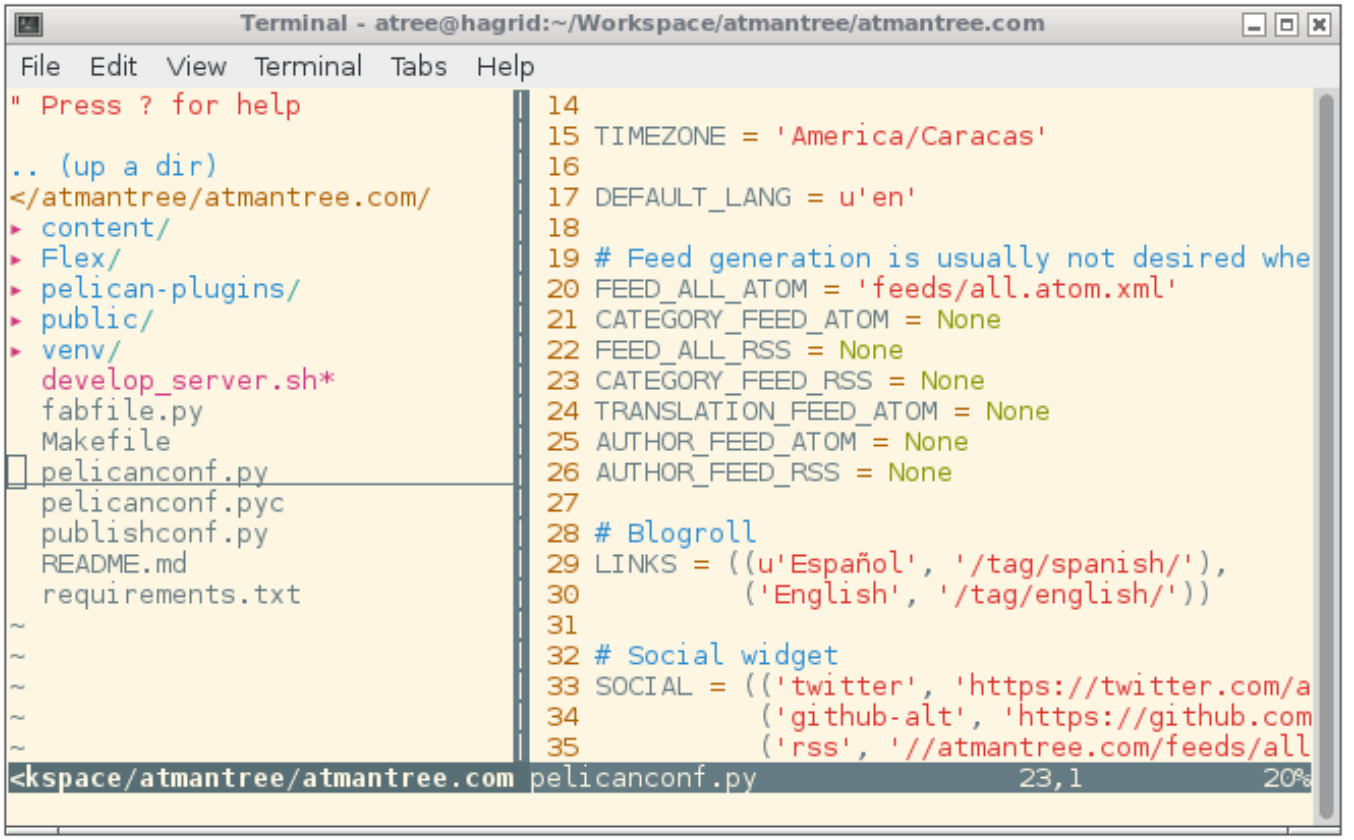
Personalización, Scripts y Plugins

Plugins

- Instalación:
  - Instalarlos a mano en la carpeta *.vim/* con algo de trabajo sobre el archivo *.vimrc* (ejemplo http://stackoverflow.com/a/1639654)
  - Usar gestor de plugins:
    - Vundle
    - Pathogen
    - ~~Neobundle~~ Dein

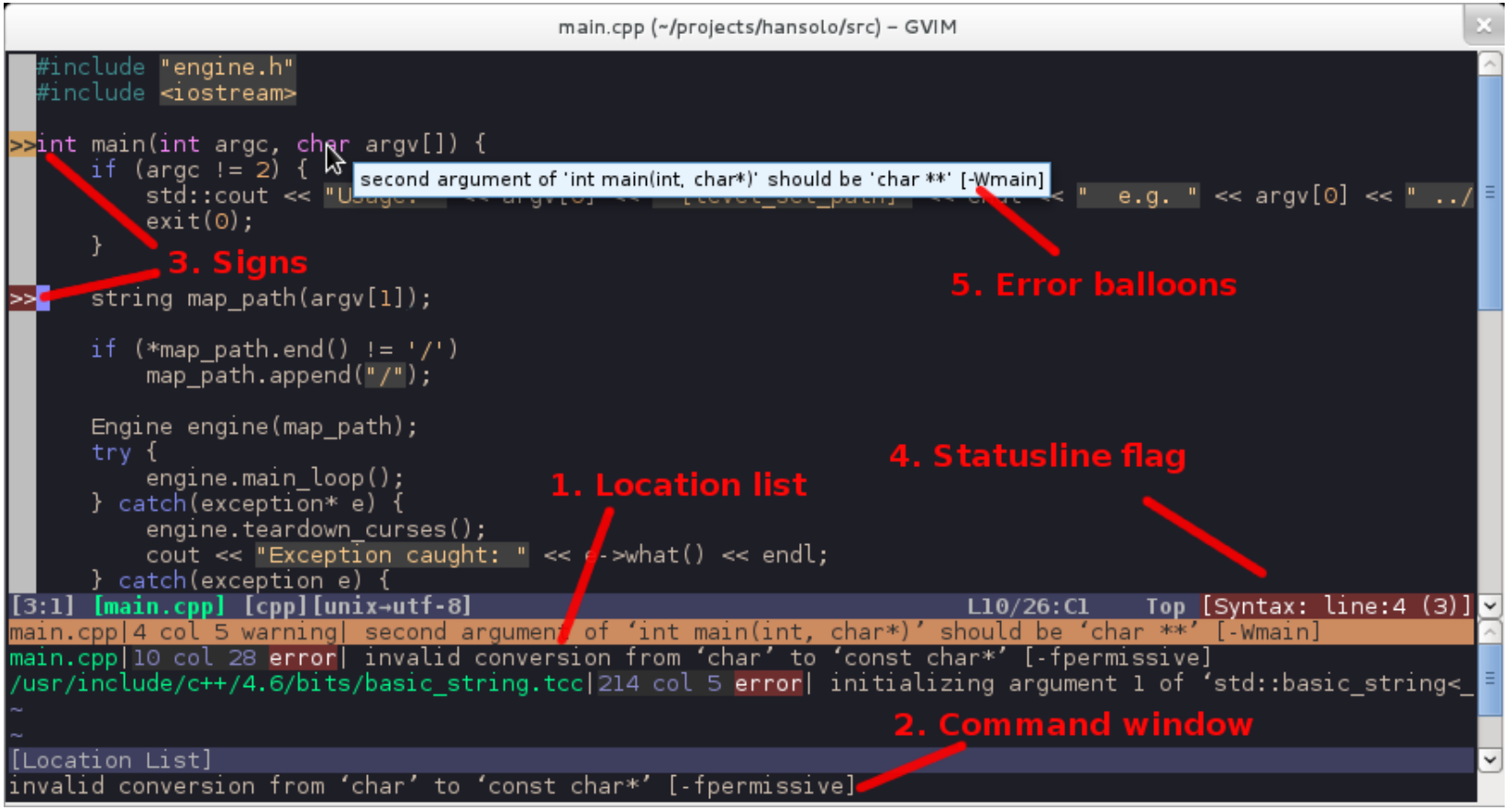
Plugins Útiles

- NerdTree



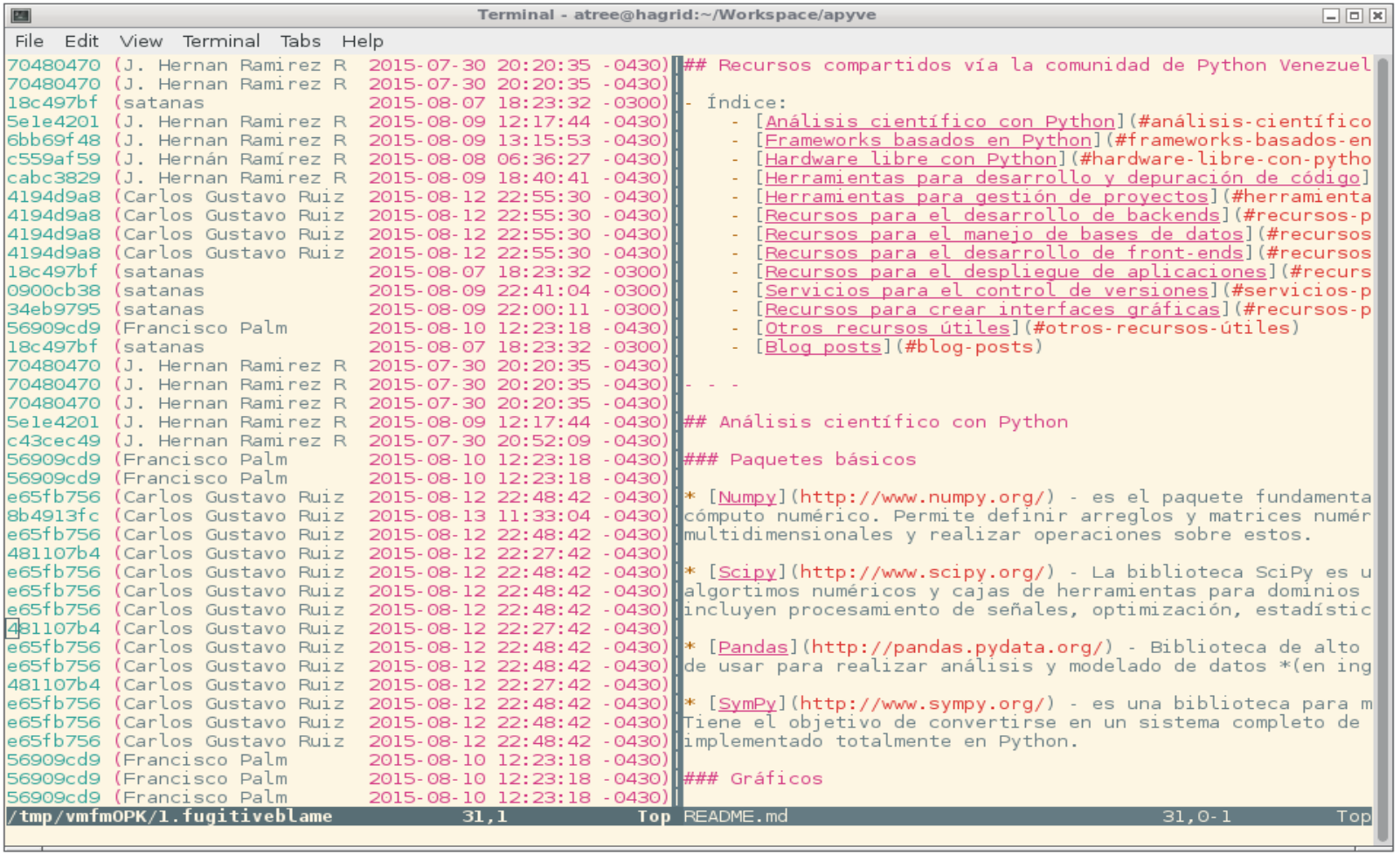
Plugins Útiles

- Syntastic



Plugins Útiles

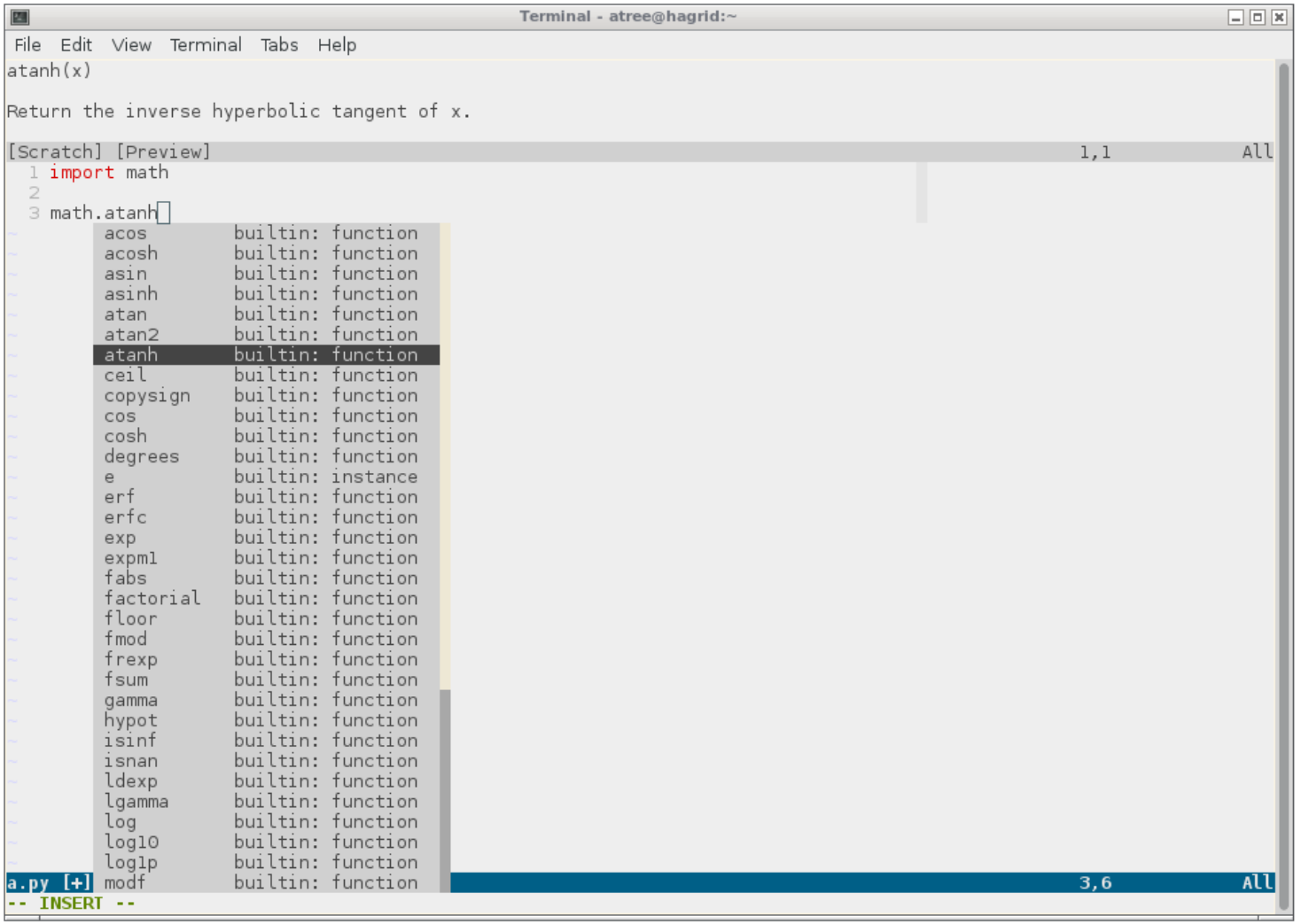
- Figitive



https://github.com/pyve/awesome-pyve

Plugins Útiles

- YouCompleteMe



Python Mode

Python Mode es un plugin para facilitar el trabajo con Python dentro de VIM, incluye un grupo de herramientas entre las que se cuentan:

- Pylint → Analizador de código fuente
- Rope → Herramienta de refactorización. Ayuda a buscar las definiciones de objetos y funciones.
- PyDoc → Herramienta para obtener la documentación de los módulos utilizados.
- PyFlakes →Validaciones dentro del código.
- Pep8 → Para estándares de codificación.
- Pep257 → Alerta de Docstrings faltantes.
- mccabe → Análisis de complejidad ciclomática

Python Mode

Entre las bondades de python-mode se cuentan:

- Soporte para Python 2 y Python 3
- Resaltado de sintaxis mejorado
- Soporte para virtualenv
- Ejecutar código python
- Agregar y quitar puntos de parada (breakpoints)
- Indentación mejorada
- Folding (compactación)
- Movimientos definidos para Python
- Revisión de código
- Autoarreglo de errores de PEP 8
- Busqueda en documentación de Python
- Ir a definición
- Y más..

Python Mode

Demo

Recursos

- http://www.vim.org/
- Para aprender
  - vimtutor
  - http://www.openvim.com/
  - http://vim-adventures.com/
  - http://www.vimgenius.com/
- Para Plugins
  - http://vimawesome.com/

Recursos

- Para consultas rápidas
  - :help
  - http://vim.wikia.com/
- Configuraciones “llave en mano”
  - http://vim-bootstrap.com/
  - http://vim.spf13.com/
- Comunidad Vim de Venezuela
  - https://telegram.me/vimvnzla

Referencias

- https://en.wikibooks.org/wiki/Learning_the_vi_Editor/Vim
- https://youtu.be/5r6yzFEXajQ
- https://youtu.be/YhqsjUUHj6g
- http://www.vim.org/6k/features.es.txt
- http://vim.wikia.com/wiki/Vim_Tips_Wiki