Creative Software Programming

Lab1: Supplement Material - Vim Advanced Usage

Yoonsang Lee

Fall 2019

Shell settings for convenient vim use - 1

- set completion-ignore-case on
  - bash (shell) has an autocompletion function (by pressing
    \<tab\> key), and it is convenient to make this
    autocompletion case insensitive.

  - Just add "set completion-ignore-case on" to your .inputrc
    file. You can do it like this:

(Shell – home directory)
```
echo 'set completion-ignore-case On' >> ~/.inputrc
```

  - Close and reopen the terminal to see it apply.

Shell settings for convenient vim use - 2

- stty -ixon
  - This setting allows you to use the shortcut keys used with Ctrl in vim.

  - Just add "stty -ixon" to your .bashrc. You can do it like this:

(Shell – home directory)
`echo 'stty -ixon' >> ~/.bashrc`

  - Close and reopen the terminal to see it apply.

Install vim-gtk

- The vim version installed by default on Ubuntu is a vim-tiny version without many useful features.

- Run the following command to install vim-gtk, which contains most of the functionality.

(Shell)
```
sudo apt-get install vim-gtk
```

Vim configuration - .vimrc

- You can change various settings of Vim through **.vimrc** file in your home directory.

- .vimrc is written in Vim's own scripting language called vim script.
  - Comment starts with "
  - For more detail, refer http://learnvimscriptthehardway.stevelosh.com/

- Since there is no .vimrc file at first, let's start editing the .vimrc file with the following command in your home directory:

(Shell)
```
vi .vimrc
```

Recommended minimal settings for .vimrc

```
.vimrc
syntax on    "use syntax highlighting
filetype plugin indent on    "use auto-indentation

set expandtab    "use spaces instead of a tab
set tabstop=4    "number of spaces for a tab
set shiftwidth=4 "number of spaces for each step of indent (e.g. when using '>' or '<')

set nowrap  "stop line breaking
set clipboard=unnamedplus "use system clipboard (e.g. when using 'yy')
set ignorecase  "case-insensitive search
set incsearch    "use incremental search

"disable automatic comment insertion
autocmd FileType * setlocal formatoptions-=c formatoptions-=r formatoptions-=o
```

- After updating .vimrc, enter :so% to apply the updated settings. Try to modify and apply the above file.
- The above file contains minimal items, so feel free to modify / add / remove any settings you want.

Vim Visual Mode

Marking text (visual mode) https://vim.rtorr.com/l

- v – start visual mode, mark lines,
  then do a command (like y–yank)

- V – start linewise visual mode

- Ctrl + v – start visual block mode

- In the .vimrc file, mark a block and copy (y), paste (p), and so on.

Vim Windows

Working with multiple files

https://vim.rtorr.com/

- `Ctrl + ws` – split window
- `Ctrl + ww` – switch windows
- `Ctrl + wq` – quit a window
- `Ctrl + wv` – split window vertically
- `Ctrl + wh` – move cursor to the left window (vertical split)
- `Ctrl + wl` – move cursor to the right window (vertical split)
- `Ctrl + wj` – move cursor to the window below (horizontal split)
- `Ctrl + wk` – move cursor to the window above (horizontal split)

- Split your .vimrc file into multiple windows and move between them.

Vim Plug-ins

- Many plugins are available to extend Vim's functionality.
  - https://vim.sourceforge.io/scripts/script_search_results.php?keywords=&script_type=&order_by=rating&direction=descending&search=search

- You can download and use plug-in files directly, but using "vim plugin managers" is much more convenient to install and manage plug-ins.

Install vim-plug

- One of the most popular vim plugin manager

- https://github.com/junegunn/vim-plug

- To install it, you need to install git first and then configure it initially.

- Replace the gray part below with your information.

(Shell)
```
sudo apt-get install git
git config --global user.name "Your Name"
git config --global user.email "you@example.com"
```

- Install vim-plug following the instructions on the page above.

Using vim-plug

- Let's install two most useful plugins.
  - Add the following lines to .vimrc and save it, and type :so% and :PlugInstall

```
.vimrc
call plug#begin()
Plug 'scrooloose/nerdtree'
Plug 'scrooloose/nerdcommenter'
call plug#end()
```

- To install other plugins, you can add them like Plug '<Github account name> / <Github project name>'.
  - Most Vim plugins are maintained on Github.
  - e.g.) NERDTree https://github.com/scrooloose/nerdtree

NERDTree

- Plugin that allows directory browsing on Vim.

- How to run: Enter :NERDTree in Vim and press Enter

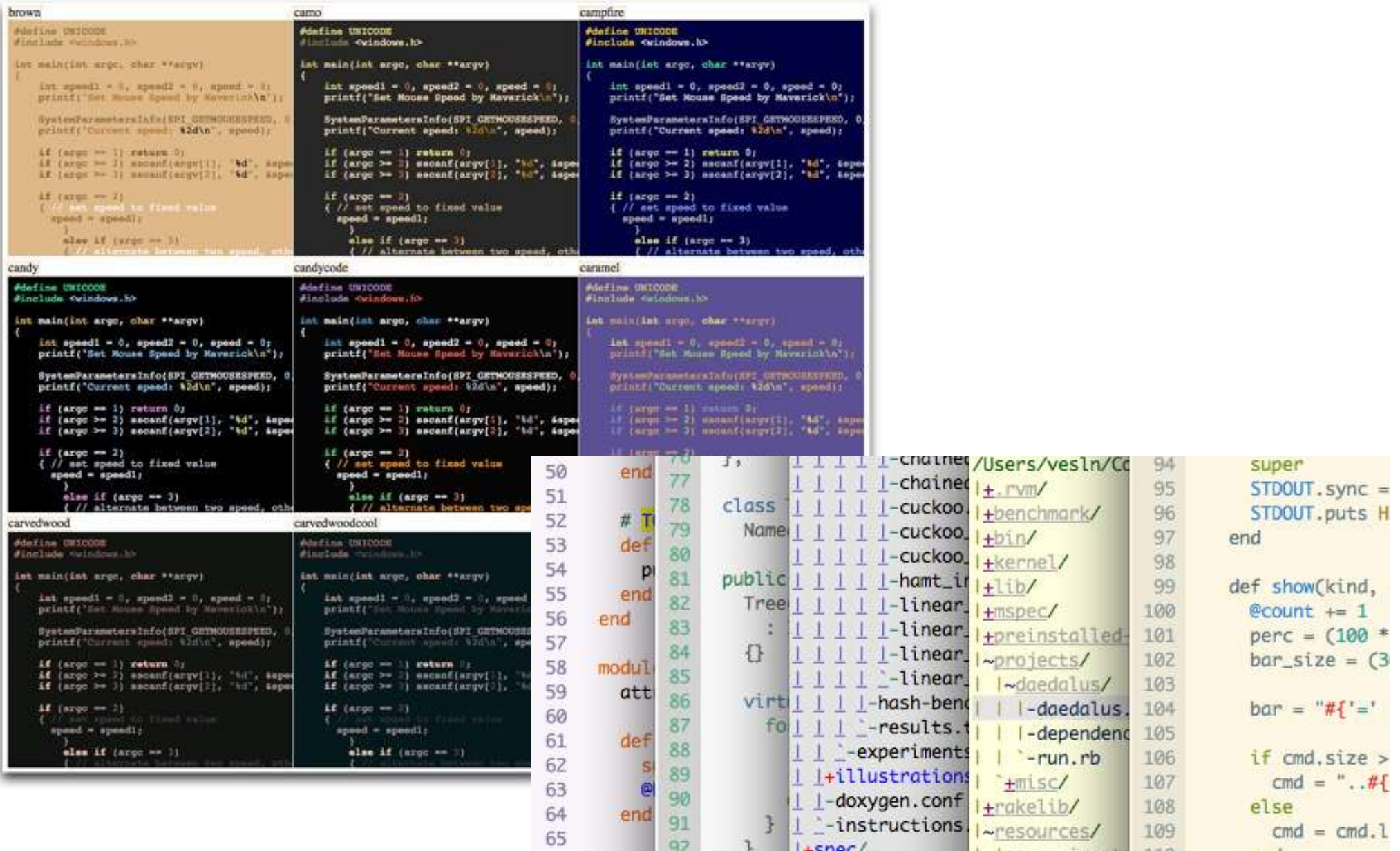- Go to the NERDTree window and press ? to test its functions.

NERD Commenter

- Plugin that provides comment / uncomment function for various programming languages.
- Let's add a new shortcut by adding the following to .vimrc:

.vimrc

```
let mapleader=","                   " change <leader> key
let NERDCreateDefaultMappings = 0      "disable default mapping
let NERDCommentWholeLinesInVMode = 1 "always comment whole line
map <Leader>c <plug>NERDCommenterComment
map <Leader>x <plug>NERDCommenterUncomment
```

- After reloading .vimrc by typing :so%,
- ,c : comment
- ,x : uncomment

Vim colorscheme

Install Vim colorscheme

- A colorscheme can be installed like a plugin

- Let's install two most popular colorschemes.
  - Add the following bold lines to .vimrc and save it, and type :so% and :PlugInstall

---

.vimrc

```
call plug#begin()
Plug 'scrooloose/nerdtree'
Plug 'scrooloose/nerdcommenter'
Plug 'vim-scripts/xoria256.vim'
Plug 'vim-scripts/peaksea'
call plug#end()
```

Apply Vim colorscheme

- Add the following lines to .vimrc to apply an installed colorscheme.
  - Add the following lines to .vimrc and save it, and type :so%
  - Apply another colorscheme by changing commented / uncommented lines.

.vimrc
``` colorscheme xoria256 "colorscheme peaksea ```

More Colorschemes?

- http://vimcolors.com/

- https://vim.sourceforge.io/scripts/script_search_results.php?keywords=&script_type=color+scheme&order_by=rating&direction=descending&search=search

- Googling 'vim colorscheme'

- Find the address of a colorscheme's Github project and add it between plug#begin() and plug#end().