What are the three questions my audience would ask me?

Coding standards
XDebug
Customize: Color Schemes
How to use: tabs
Plugins: ctrlp
leader-fef

Adding things to your .vimrc.after
http://drupal.org/node/29325
https://github.com/carlhuda/janus/wiki/Customization

Plug-Ins

Snipmate: https://github.com/honza/snipmate-snippets
TAGBAR: `<Leader>rt` to toggle Tagbar.
XDebug:
http://2bits.com/articles/using-vim-and-xdebug-dbgp-for-debugging-drupal-or-any-php-application.html
Fugitive: https://github.com/carlhuda/janus for custom mappings, https://github.com/carlhuda/janus
NerdTree: \n
BufferGator: \b
Syntastic: Create an error, try to save
Supertab: start typing a variable, tab
Snipmate: h_menu(tab) mi(tab)
NerdCommenter: \c<space> (toggle), or \cm (multiline)
Ack: Shift Command f gets you ack, if you have done a brew install ack
Easy Motion: \\f find all occurrences ; repeats in normal for f, , is go backward

Other Really Cool Stuff

In file explorere (:E or :e or :Vex or :Sex) % is new file
d creates a directory

In edit mode, the . repeats, AWESOME!

Do ":help Text Objects" and read up, they are magic.

= auto-indents, so:
G goes to the end of the file, so =G autoindents to end of file
And gg goes to the beginning of the file, so, "gg=G" in Normal mode auto indents the whole doc

:changes shows all changes made to the doc, pretty neat!
:jumps shows everywhere you moved your cursor.

Other debugger option: VImdebugger

Moving Around

gg to the top of the file (or :0)
G to the bottom
ctrl-d and ctrl-u for down and up

Split Window commands (from vimcasts.org)

command	action

`ctrl-w s`	split the current window horizontally, loading the same file in the new window
`ctrl-w v`	split the current window vertically, loading the same file in the new window
`:sp[lit] filename`	split the current window horizontally, loading *filename* in the new window
`:vsp[lit] filename`	split the current window vertically, loading *filename* in the new window

Closing split windows

command	action
`:q[uit]`	close the currently active window
`:on[ly]`	close all windows except the currently active window

Changing focus between windows

command	action
`ctrl-w w`	cycle between the open windows
`ctrl-w h`	focus the window to the left
`ctrl-w j`	focus the window to the down
`ctrl-w k`	focus the window to the up
`ctrl-w l`	focus the window to the right

ctrl-w HJKL moves the current split window in the direction you choose

ctrl-w = equalizes all sizes
ctrl-w _ expands vertically
crtl-w | expands horizontally

:tabedit filename
ctrl-w T moves split window to it's own tab
gt moves forward, gT backward
3gt goes to third tab

g; puts you back where you were last changing the file, it's moving back through the change list
g, goes forward

ctrl-o
ctrl-i move you around

A takes you to the end of the line and puts you in insert
C deletes to the end of the line and puts you in insert
c is like d, deletes a word, but then, places you in insert, so it can be repeated with .

:e! reloads current file from disk, removing local unsaved changes
If you are on a git repo'd file, :Gread does the same thing.

Columnar block editing ctrl-V, shift-I to insert, c to replace
shift-A adds after cursor

change your .vimrc, then re-source it with

```
:source $MYVIMRC
```

Mappings allow you to set up Command # to go to a tab of that number, like firefox

```
" For mac users (using the 'apple' key)
map <D-S-]> gt
map <D-S-[> gT
map <D-1> 1gt
map <D-2> 2gt
map <D-3> 3gt
map <D-4> 4gt
map <D-5> 5gt
map <D-6> 6gt
map <D-7> 7gt
map <D-8> 8gt
map <D-9> 9gt
map <D-0> :tablast<CR>
```