Bash Functions

Function	Use
`vack <regex>`	Find all files that do contain a specific regex pattern, and open them in vim.
`vnack <regex>`	Find all files that do not contain a specific regex pattern, and open them in vim.
`taila <glob>`	Tail the access log of sites starting with a glob.
`taile <glob>`	Tail the error log of sites starting with a glob.
`v <glob>`	Change to a vhosts directory starting with a glob.
`g`	The g script.

Standard Scripts

Script	Use
`get_supervisor_email <name>`	Returns the email address of the supervisor for the given name.
`git freeze <branch>`	Freezes an xERP branch in git, this includes tagging, changing JIRA, and deleting the branch.
`git merge-feature <branch>`	Merges an xERP feature branch into the current versioned branch, this includes tagging, changing JIRA, and deleting the branch.
`git report <object>`	Generates a report for a given commit, branch, or tag in a repo.
`git restore <regex>`	Finds and restores deleted or otherwise missing git files.

Vim Mappings

Key Combination	Use
`<C-w>h`, `<C-w>j`, `<C-w>k`, `<C-w>l`	Move in split windows with `ctrl+hjkl`
`+`, `-`	Intuitive increment/decrement.

Vim Mappings (cont)

`:w!!`	Save a file that you accidentally opened up without sudo.
`<F12>`	Toggle pastemode.

Vim Plugins - Easy Align

Delimiter	Alignment
`s`	Aligns sql statements by keyword.
`?`	Align on question mark.
`:`	Align on colon.
`[,]`	Align on close square bracket.
`(,)`	Align on close parenthesis.
`{,}`	Align on close curly brace.
	Align on space.
`,`	Align on comma.
`=`	Align on equal sign.
`<`	Align on left angle bracket.
`>`	Align on right angle bracket.

How to use: Select text using visual mode (`v`, `shift-v`, or `ctrl-v`), then press `<Enter>`. Now you have started easy align. Then press one of the delimiters defined above to align the text to that character. You can also specify a number before the delimiter to state which character to align on. Therefore the sequence `<Enter>3,` aligns on the third comma. Negatives work from the end. And `*` means align on all occurrences.

Vim Plugins - Ultisnips

How to use: In insert mode press `ctrl-t`, this brings up the ultisnip list. Then you can type or navigate through the list to find the snippet you want. You can also start by typing all or part of a snippet name before pressing `ctrl-t` to limit your search initially. Press `<Enter>` to select a snippet, then use `<Tab>` and `shift-<Tab>` to navigate through the snippet selection areas filling them out as you go.

Zsh Functions

Function	Use
`ack <regex>`	Search for text inside a directory matching the regex.
`mdless <filepath>`	Process a markdown file through pandoc and open it in a pager.

Zsh Functions (cont)

`zsh_stats`	Find out how much you use commands.
`which_alias <alias>`	Get the value of an alias.
`take <filepath>`	Make a directory and go to it all in one step.
`vack <regex>`	Find all files that do contain a specific regex pattern, and open them in vim.
`vnack <regex>`	Find all files that do not contain a specific regex pattern, and open them in vim.
`taila <glob>`	Tail the access log of sites starting with a glob.
`taile <glob>`	Tail the error log of sites starting with a glob.
`v <glob>`	Change to a vhosts directory starting with a glob.
`g`	The g script.

Zsh Aliases

Alias	Mapping
`rm`	`rm -i`
`mv`	`mv -i`
`cp`	`cp -i`
`ll`	`ls -alF`
`la`	`ls -A`
`l`	`ls`
`cl`	`clear && ls`
`..`	`cd ..`
`cd..`	`cd ..`
`cd...`	`cd ../..` (and so on...)
`md`	`mkdir -p`
`rd`	`rmdir`
`d`	`dirs -v | head 10`
`1`	`cd -`
`2`	`cd -2`
`3`	`cd -3` (and so on...)

Vim Plugins - Endwise

How to use: In a shell like document this plugin automatically adds the closing `fi` or `endifs` and the like for you.

Vim Plugins - Fugitive

Command	Use
`:Gstatus`	Bring up the current status in a buffer.
`:Gcommit`	Starts the commit process in vim.
`:Gpush`	Pushes the repo.
`:Gblame`	Opens a blame window for the current file.

How to use: See vim-fugitive for details.

Vim Plugins - Supertab

How to use: Tab now autocompletes.

Vim Plugins - Syntastic

Mapping	Use
`<F7>`	Start syntax and style checking.
`<F8>`	Toggle syntax and style checking.
`<Home>`	Move up through the errors list.
`<End>`	Move down through the errors list.

How to use: When you're in a file press`<F7>` to start syntastic. Then navigate through the document and fix the errors/warnings. On each save the checkers will run again and show the output. You're goal should be to get a clean file with no output.