

## VIM PERSONAL MAPPINGS

(CC) 2012 Michael Goerz ([www.michaelgoerz.net](http://www.michaelgoerz.net))

<Leader>: ,      <LocalLeader>: \

### Misc

*n*: <Leader>t ..... Show or hide Taglist  
*n*: <Leader>d *i*: ^Ld ..... Insert date stamp  
*n,v*: <Leader>p *i*: ^Lp ..... toggle paste  
*n,v*: <Leader>s ..... Send to b:sendToProgramName  
*i*: ^J ..... Jump to next placeholder  
*n*: <Leader>w ..... Write current file  
:SuperTabHelp ..... Set behavior of Tab key  
:Bclose .... Delete current buffer w/o closing window  
:German ..... activate spell checking in German  
:English ..... activate spell checking in English  
:ManualFolding ..... Activate folding, fdm=manual  
:Errors ..... Open syntastic list of errors  
:SyntasticCheck .... Manually cause syntastic check  
*n*: <Leader>u ..... Toggle Gundo window  
:GundoToggle ..... Toggle Gundo window

### Sudo

:e sudo:/etc/passwd ..... Edit file with sudo perms  
:w sudo:/etc/passwd .... Write file with sudo perms  
:w sudo:% ... Write current file with sudo permissions

### Filters

:ListFilters ..... List filters for filetype  
:’a,’b Filter ..... Filter range  
:Filter ..... Filter entire file  
*n*: <Leader>f *v*: ^Lf ..... Filter entire file/selection

### NerdCommenter

*n*: <Leader>cc ..... comment line or selection  
*n*: <Leader>cn .... comment line or selection (nested)  
*n*: <Leader>c\$ ..... comment to end of line  
*n,v*: <Leader>c<space> ..... toggle line or selection  
*n,v*: <Leader>ci ..... toggle individual lines  
*n,v*: <Leader>cs ..... “sexy” commenting  
*n,v*: <Leader>cu ..... uncomment line or selection  
*i*: ^c ..... Insert comment

### Surround

*n*: ysmr ..... Wrap movement *m* in *r*  
*n*: ysiw" ..... Wrap current word in quotes  
*n*: ysm1 ..... Wrap *m* in latex environment  
*n*: ysmt ..... Wrap *m* in html tag  
*n*: ysmf ..... Wrap *m* in function  
*n*: ysm1 ..... Wrap *m* in latex environment  
*n*: ySmr .... Wrap movement *m* in *r*, new lines, indent  
*n*: yssr ..... Wrap current line in *r*  
*n*: ySSr ..... Wrap current line in *r*, new lines, indent  
*n*: cstr ..... Replace target surrounding *t* with *r*  
*n*: cs'' ..... Replace double quotes with single quotes  
*n*: dst ..... Delete target surrounding  
*v*: sr ..... Wrap selection in *r*  
*v*: s" ..... Wrap selection in quotes  
*v*: S" ..... Wrap selection in quotes, new lines, indent

### Git

:Git [args] ..... Run git  
:GitAdd [afile] ..... Stage changes  
:GitBlame ..... Show blame  
:GitCheckout [args] ..... Checkout branch  
:GitCommit [args] ..... Commit  
:GitDiff [args] ..... Show diff  
:GitLog [args] ..... Show commit logs  
:GitPull [args] ..... Fetch and merge  
:GitPullRebase [args] ..... GitPull –rebase  
:GitPush [args] ..... Push to remote  
:GitStatus ..... Working tree status  
:GitGrep [args] ..... Perform git grep  
:GitVimDiffMerge [args] ..... vimdiff for merge  
:GitVimDiffMergeDone ..... Mark conflicts resolved  
*n*: <Leader>gd ..... GitDiff  
*n*: <Leader>gD ..... GitDiff –cached  
*n*: <Leader>gs ..... GitStatus  
*n*: <Leader>gl ..... GitLog  
*n*: <Leader>gL ..... Git log –all –graph –decorate  
*n*: <Leader>ga ..... GitAdd  
*n*: <Leader>gc ..... GitCommit  
*n*: <Leader>gC ..... GitCommit –a  
*n*: <Leader>gp ..... GitPull  
*n*: <Leader>gb ..... GitBlame  
*n*: <Leader>gA ..... GitCommit –amend

### Snippets

:ListSnippets ..... List snippets for filetype  
:AppendSnippets ..... Append snippet after cursor  
:InsertSnippets ..... Insert snippet before cursor  
*n*: <Leader>ni *v*: ^Lni ..... :InsertSnippet  
*n*: <Leader>na *v*: ^Lna ..... :AppendSnippet

### TeX

:LatexFold ..... Fold up LaTeX (fdm=manual)  
*n,v*: <LocalLeader>e *i*: ^Le .insert or wrap in \emph  
*n,v*: <LocalLeader>o *i*: ^Lo ... insert or wrap in \Op  
*i*: ^Lk ..... insert \Ket  
*i*: ^L2 ..... insert \sqrt  
*i*: ^Lv ..... insert \vec  
*i*: ^Lh ..... insert \hat  
*n*: <LocalLeader>i *i*: ^Li .insert env/math function  
*i*: ^Lt ..... \text  
:let b:leftquote = ’ ’ ..... German left quotes  
:let b:rightquote=“ ” ..... German right quotes  
:setlocal makeprg=... ..... change compiler  
*i*: ‘a ... ‘o .....  $\alpha \dots \omega$   
*i*: ‘c .....  $\chi$   
*i*: ‘f .....  $\phi$   
*i*: ‘q .....  $\theta$   
*i*: ‘w .....  $\omega$   
*i*: ‘v .....  $\vee$   
*i*: ‘& .....  $\wedge$   
*i*: ‘D’F’G’L’O’P’Q’U’X’Y’P’S .....  $\Delta\Phi\Gamma\Lambda\Omega\Pi\Theta\Upsilon\Xi\Psi\Sigma$   
*i*: ‘I .....  $\int_a^b$   
*i*: ‘N .....  $\nabla$   
*i*: ‘E .....  $\epsilon$  (var- $\epsilon$ )  
*i*: ‘H .....  $\varphi$  (var- $\phi$ )  
*i*: ‘R .....  $\rho$  (var- $\rho$ )  
*i*: ‘T .....  $\vartheta$  (var- $\theta$ )  
*i*: ‘Z .....  $\sum_a^b$   
*i*: ‘1 ..... \unity  
*i*: ‘6 ..... \difquo  
*i*: ‘8 .....  $\infty$   
*i*: ‘/ or ‘% ..... \frac{}{}\br/>*i*: ‘@ .....  $\circ$   
*i*: ‘| ..... “|  
*i*: ‘| ..... \Big\|\br/>*i*: ‘= .....  $\equiv$   
*i*: ‘\ ..... \

*i*: ‘. ....\cdot  
*i*: ‘\* .....×  
*i*: ‘− ..... $\cap$   
*i*: ‘+ ..... $\cup$   
*i*: ‘( .....\left( \right)  
*i*: ‘[ .....\left[ \right]  
*i*: ‘< .....≤  
*i*: ‘> .....≥  
*i*: ‘, .....\nonumber  
*i*: ‘: .....\cdots  
*i*: ‘~ .....\tilde{}  
*i*: ‘; .....\dot{}  
*i*: ‘\_ .....\bar{}  
*i*: ‘^E .....\exp()  
*i*: ‘^L .....\lim-{}  
*i*: <Up> .....↑  
*i*: <Down> .....↓  
*i*: <Left> .....←  
*i*: <Right> .....→  
*i*: ^F .....→

## C

*n,v*: <LocalLeader>cc ..... Open reference TOC  
*n,v*: <LocalLeader>cr .... View reference for keyword  
*n,v*: <LocalLeader>cw . View reference (ask keyword)

## Perl

*n*: K ..... Look up ‘perldoc -f’ for word  
*:make* ..... Check for syntax errors

## Python

*n*: <LocalLeader>pw ..... View reference for keyword  
*n,v*: K ..... Look up documentation for word  
*:Pydoc name* ..... Look up name in documentation  
*:make* ..... Run through pylint

## Fortran

*n*: <LocalLeader>i *i*: ^Li ..... Complete construct  
*:FortranFold* .. Fold up Fortran code (fdm>manual)

## XML/HTML

*n*: <LocalLeader>i *i*: ^Li ..... Insert/convert to tag