

VIM PERSONAL MAPPINGS

Misc

n: \t Show or hide Taglist
n,v: \s Send to b:sendToProgramName
i: ^J Jump to next placeholder
:lipsum 1 par 5 words add lorem ipsum text
n: \ni *v:* ^Lni :InsertSnippet
n: \na *v:* ^Lna :AppendSnippet
:ListSnippets List snippets for filetype
:SuperTabHelp Set behavior of Tab key

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
:GitVimDiffMerge [args] vimdiff for merge
:GitVimDiffMergeDone Mark conflicts resolved
n: \gd GitDiff
n: \gD GitDiff -cached
n: \gs GitStatus
n: \gl GitLog
n: \ga *or* \gA GitAdd
n: \gc GitCommit
n: \gp GitPullRebase

NerdCommenter

n: ,cc comment line or selection
n: ,cn comment line or selection (nested)
n: ,c\$ comment to end of line
n,v: ,c<space> toggle line or selection
n,v: ,ci toggle individual lines
n,v: ,cs “sexy” commenting
n,v: ,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: ysm\ Wrap *m* in latex environment
n: ysmt Wrap *m* in html tag
n: ysmf Wrap *m* in function
n: ysm\ 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

C

n,v: \cc Open C reference TOC
n,v: \cr View C reference for keyword
n,v: \cw Ask for keyword, view C reference

TeX

n: \lr Check References
n: \lc Check Citations
n: \lg Look at Log File
n: \i *i:* ^Li Insert environment or math function
:let b:leftquote = ’"’ German left quotes
:let b:rightquote = "\" German right quotes
:setlocal makeprg=... change compiler
i: ‘a ... ‘o $\alpha \dots \omega$
i: ‘c χ
i: ‘f ϕ
i: ‘q θ
i: ‘v \vee
i: ‘w *or* ’& \wedge
i: ‘D’F’G’L’O’P’Q’U’X’Y’P’S $\Delta\Phi\Gamma\Lambda\Omega\Pi\Theta\Upsilon\Psi\Sigma$
i: ‘I \int_a^b
i: ‘N ∇
i: ‘E ε (var- ϵ)
i: ‘H φ (var- ϕ)
i: ‘R ϱ (var- ρ)

i: ‘T ϑ (var- θ)
i: ‘Z \sum_a^b
i: ‘1 \left
i: ‘2 \right
i: ‘3 \Big
i: ‘6 \difquo
i: ‘8 ∞
i: ‘/ *or* ‘% \frac{}{}
i: ‘@ \circ
i: ‘| “|
i: ‘| \Big\|
i: ‘= \equiv
i: ‘\ \backslash
i: ‘. \cdot
i: ‘* \times
i: ‘- \cap
i: ‘+ \cup
i: ‘(..... \left(\right)
i: ‘[..... \left[\right]
i: ‘< \leq
i: ‘> \geq
i: ‘, \nonumber
i: ‘: \cdots
i: ‘~ \tilde{}
i: ‘; \dot{}
i: ‘^ \op{}
i: ‘_ \bar{}
i: ‘E \exp()
i: ‘L \lim{}
i: <Up> \uparrow
i: <Down> \downarrow
i: <Left> \leftarrow
i: <Right> \rightarrow
i: ‘F \rightarrow

XML/HTML

n: \i *i:* ^Li Insert Tag / Convert word to tag
:%tidy -w 80 .Tidy html fragment with linewidth 80

Fortran

n: \i *i:* ^Li Complete Fortran construct
:FortranFold Fold up Fortran code

Python

n: \pw leaderbspW View Python reference for keyword

n,v: K Look up documentation for word

:Pydoc *name* Look up name in documentation

:make Run through pylint

Perl

n: K Look up 'perldoc -f' for word

:make Check for syntax errors