

configuration	track projects	track projects (cont)
username <code>git config --global user.name "John Doe"</code>	track new project <code>git init</code>	rename file <code>git mv file.txt newfile.txt</code>
email <code>git config --global user.email "john.doe@example.com"</code>	clone project <code>git clone <url></code>	git status -s example MM Rakefile = has been staged as modified and then modified again in the working space (and this edit has not been staged yet). _M Readme = modified in working space and not staged yet ?? LICENCE.TXT = not tracked yet A_file.txt = added to staging
default editor <code>git config --global core.editor emacs</code>	clone project in directory <code>git clone <url> <destination></code>	git rm examples <code>git rm log/*.log</code> <code>git rm *</code>
check current settings <code>git config --list</code>	check status <code>git status</code>	check differences working vs staged <code>git diff</code> staged vs commit <code>git diff --staged</code>
getting help <code>git help <verb> git help config</code>	check status (short) <code>git status -s</code>	check diff <code>git diff --tool -help</code>
For specific projects, just remove the --global option	track new file or add to staging (before commit) <code>git add <filename></code>	use external editor <code>git difftool</code>
.gitignore file example	Remember: if you stage a file and then change it again, this will be marked as changed and not staged and will be committed without the latest changes (not yet staged)	
# no .a files *.a # but do track lib.a, even though you're ignoring .a files above !lib.a # only ignore the TODO file in the current directory, not subdir /TODO /TODO # ignore all files in the build/ directory build/ # ignore doc/notes.txt, but not doc/server/arch.txt doc/*.txt # ignore all .pdf files in the doc/ directory doc/*.pdf	commit changes (comment on editor) <code>git commit</code> commit changes (comment inline) <code>git commit -m "commit message"</code> commit skipping staging <code>git commit -am "message"</code>	
More examples here https://github.com/github/gitignore	remove & delete file <code>git rm <file></code> <i>file will be removed from working AND staging area</i> remove & keep file <code>git rm --cached <file></code> <i>file be not tracked but kept in working area</i>	

