

configuration

```
username git config --global
user.name "John Doe"
```

```
email git config --global
user.email
johndoe@gmail.com
```

```
default git config --global
editor core.editor emacs
```

```
check git config --list
current
settings
```

```
getting git help <verb>git help
help config
```

For specific projects, just remove the --global option

.gitignore file example

```
# 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
```

More examples here

<https://github.com/github/gitignore>

track projects

```
track git init
existing
```

this should be done inside existing directory

```
clone git clone <url>
project
```

```
clone git clone <url>
project in <dest-url>
dir
```

```
check git status
status
```

```
check git status -s
status
(short)
```

```
track new git add <filename>
file or add
to staging
(before
commit)
```

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)

```
commit git commit
changes
(comment
on editor)
```

```
commit git commit -m "commit
changes message"
(comment
inline)
```

```
commit git commit -am "message"
skipping
staging
```

```
remove & git rm <file>
delete file
```

file will be removed from working AND staging area

```
remove & git rm --cached <file>
keep file
```

file be not tracked but kept in working area

track projects (cont)

```
rename/move git mv file.txt
file newfile.txt
```

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

git rm examples

```
git rm log/*.log
git rm *~
```

check differences

```
working vs staged git diff
```

```
staged vs commit git diff --staged
```

```
check diff external git difftool --
editor tool-help
```

```
use external editor git difftool
```