

Customizing Git

Aliases

Commands can be composed and renamed in Git to better suit your working style. For example, I've renamed `git status` to `git s` and I've composed `git log --oneline --decorate` to `git logod`.

The configuration lives in a file in your home directory called `.gitconfig`.

Alias can be set up in section like the following in the `.gitconfig` file:

```
[alias]
  br = branch
  bra = branch -a
  s = status
  cl = clone
  ci = commit
  co = checkout
  d = diff
  dw = diff --word-diff
  dh = diff HEAD
  dc = diff --cached
  who = shortlog -s --
  ph = push
  pl = pull
  lg = log -p
  lgod = log --oneline --decorate
```

Let's discuss some of these shortcuts and try them out in our own.

Aliases can also be added via the `config` command, but find this to be less efficient than editing the `~/.gitconfig` file directly.

```
git config {-global alias.who shortlog -s --
```

Aliases only allow for Git commands by default, but you can coax it to run shell commands by prefixing the alias with an exclamation `!`.

```
dm = !git diff | mate
dv = !git diff | vim
publish = !git checkout master && git pull && git checkout dev && git rebase master && git c
```

The `publish` command comes from the ideas of Justin French.