

# **tig**とか**alias**なし生活を送 ってみて改めて**git**を覚え てる話

@vividmuimui

2018/08/01 社内LT

# はじめに

ここ2,3週間ぐらい、Git/Githubの研修資料を作るために、tigやgit aliasを封印してしてます。

その中で今まで知らなかったこと、見たことはあるけど忘れていたものとかを発表するLTです。

tigについては

[https://qiita.com/vivid\\_muimui/items/7e7a740e6537749de0c0](https://qiita.com/vivid_muimui/items/7e7a740e6537749de0c0)

## 注意

難しい話とかトリッキーな話とかではないです。基本的な話しかないです！

人によっては当たり前のことが多いかと思いますが！

優しい目で見てください！

# 目次

- add -u
- reset -mixed, -soft, -hard
- show HEAD~^2~~^~^2
- HEAD, @
- -staged is a synonym of -cached
- diff topic..master/topic...master

# add -u

<https://git-scm.com/docs/git-add>

```
-u, --update          update tracked files
```

対象となるファイルは `commit -a` と一緒です。  
`commit -a` や `add .` しか知らなかったのので地味に便利かなと思っています。  
`commit -am` は少し躊躇するし、`rebase` や `merge` などでもコンフリクトしてコンフリクト解消したときとかにも便利そう(予感)

# reset --mixed, --soft, --hard

<https://git-scm.com/docs/git-reset>

今回改めて覚えるまで、  
デフォルトのオプションがどれかもよくわかってなかったし、--hardしか普段使ってなかったのも、  
--mixedと--softの違いもよくわかってなかったでした。

# reset --mixed, --soft, --hard

- デフォルトは --mixed と --softどちらかわかりますか？
- --mixedと--softの違いを理解できますか？

# reset --mixed, --soft, --hard

<http://d.hatena.ne.jp/murank/20110327/1301224770>

--soft、--mixed(オプションなしと同等)、--hard オプションがあり、影響度の小さい順に以下ようになる。

--soft

HEAD の位置のみを変更する。インデックス、ワーキングツリーには影響なし。

--mixed (またはオプションなし)

HEAD の位置とインデックスを変更する。ワーキングツリーには影響なし。

--hard

HEADの位置、インデックス、ワーキングツリーをすべて変更する。

詳しくは <http://d.hatena.ne.jp/murank/20110327/1301224770> この記事を!



# show HEAD~^2~~^~^2

~ (チルダ)

~世代前のコミットを指定できる。

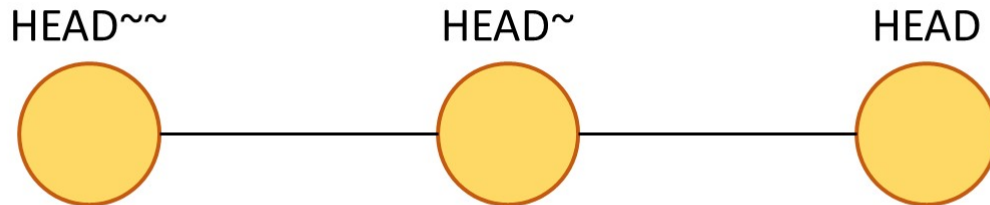
^ (キャレット)

複数ある親コミットのなかから指定できる

<https://qiita.com/chihiro/items/d551c14cb9764454e0b9>

# show HEAD<sup>~</sup>^2<sup>~</sup>^<sup>~</sup>^2

~ (チルダ)  
~世代前のコミットを指定できる。

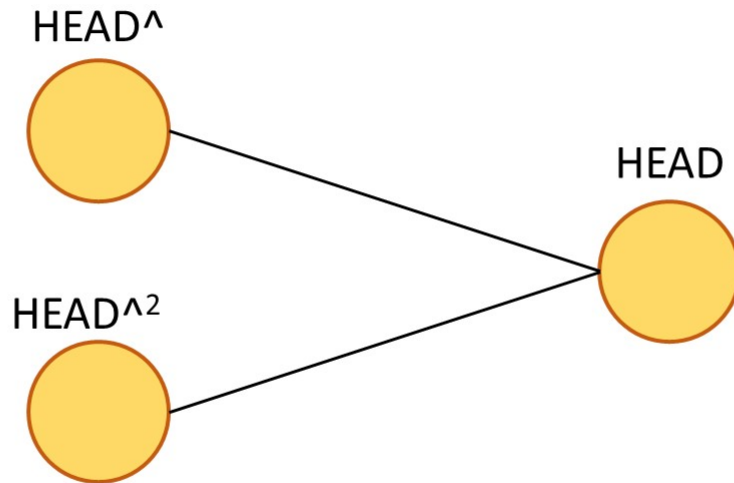


<https://qiita.com/chihiro/items/d551c14cb9764454e0b9>

# show HEAD<sup>~^2</sup>...<sup>~^2</sup>

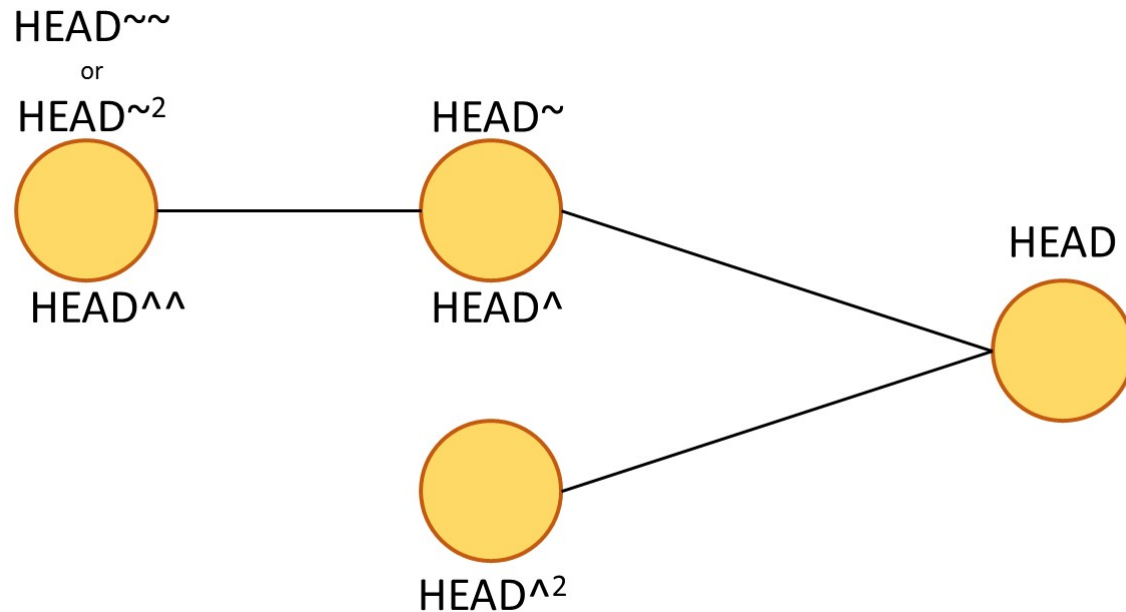
<sup>^</sup> (キャレット)

複数ある親コミットのなかから指定できる



<https://qiita.com/chihiro/items/d551c14cb9764454e0b9>

# show $\text{HEAD} \sim \wedge 2 \sim \sim \wedge \sim \wedge 2$



<https://qiita.com/chihiro/items/d551c14cb9764454e0b9>

# HEAD, @

gitのv1.8.5からは、大文字「HEAD」の4文字を打たなくて済むよう「@」というエイリアスが用意された

<http://tech.aainc.co.jp/archives/6740>

- *Instead of typing four capital letters "HEAD", you can say "@" now, e.g. "git log @".*

<https://raw.githubusercontent.com/git/git/master/Documentation/RelNotes>  
(ちなみに、1.8.5は2013年年末ごろにリリースされてます。)

# HEAD, @

git rb -i @~2とか書くことができて便利 😊  
(rb = rebase)

# –staged is a synonym of –cached

<https://git-scm.com/docs/git-diff>

```
git diff --cached  
git diff --staged
```

--cachedよりも--stagedのほうが絶対理解しやすいはず。  
(なんでcachedという名前なのか、は理解してないです 🤔)

# diff topic..master/topic...master

```
git diff topic master  
git diff topic..master  
git diff topic...master
```

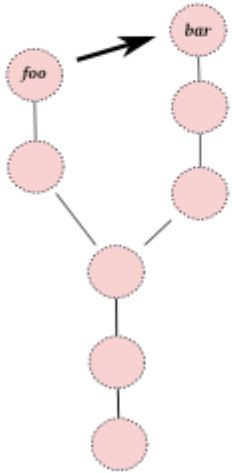
ちがいわかりますか？



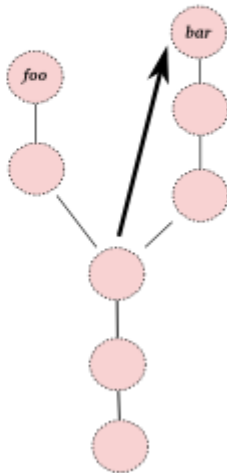
# diff topic..master/topic..master

*git diff*

```
foo..bar  
foo bar
```



```
foo...bar  
$(git merge-base foo bar) bar
```

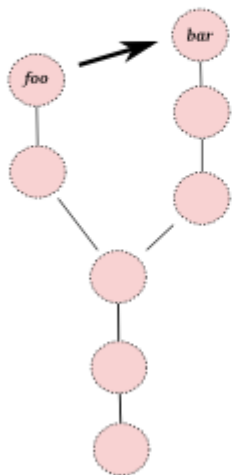


<https://yakst.com/ja/posts/4116>

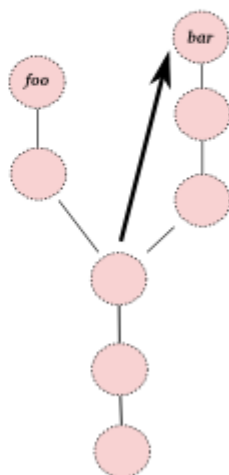
`git diff foo..bar` は `git diff foo bar` と完全に同じです。  
どちらも2つのブランチ `foo` と `bar` の最新の変更同士の違いを表示します。  
一方で、`git diff foo...bar` は、  
2つのブランチの「マージベース」と `bar` の最新の変更との違いを表示します。  
「マージベース」とは通常、2つのブランチ間で共通な最後のコミットのことです

### *git diff*

`foo..bar`  
`foo bar`



`foo...bar`  
`$(git merge-base foo bar) bar`



# diff topic..master/topic...master

githubのPRをだすときのURLが

```
https://github.com/foo/bar/compare/branchA...branchB
```

のように branchA...branchB であることがこれで納得いきますね 💡

# 終わり

tigや各種エディタ拡張とかとても便利だし、普段の開発ではaliasやpeco(history)で特定のコマンドしか使うことないと思います。

素のgitを触ることあまりないとは思いますが、  
たまーに素のgitを触ってみると新しい発見があるかもしれません！

(**Reference manual**読んだほうが早くね！とかそういう系の正論は 🤖)