

すごい git

たのしく学ぼう！





# Part 2

# git あれこれ

# 可視化するために

コミットなどの情報を確認するためのコマンドを覚えておくといい

`git status` :

ワークツリーやインデックスの状態を確認

`git log` :

積み上げて来たコミットの一覧を確認

`git diff` :

コミット間の差分(変更箇所)を確認



# さらに可視化するために

`git log` にはログを綺麗にするオプションがある  
それを駆使すると次のようなのも表示できる。

```
> git log --graph --pretty='format:%C(yellow)%h%Creset %s %Cgreen(%an)%Creset %Cred%d%Creset'
* e29ae56 Part4 をすごい雑に書いた (MATSUBARA Nobutada) (HEAD -> master, origin/master, origin/HEAD)
* 8d1b585 Part3 を雑に書いた (MATSUBARA Nobutada)
* f1d72f5 まとめの段落ミスを修正 (MATSUBARA Nobutada)
* 2bba2f1 Part0 をとりあえず書いた (MATSUBARA Nobutada)
* ba9fbc3 Part2 をとりあえず書いた (MATSUBARA Nobutada)
* afa4cbb part1 にまとめを追記 (MATSUBARA Nobutada)
* 143cb16 part 1 をざっくりと (MATSUBARA Nobutada)
* 9bd1593 テキストのひな型を追加 (MATSUBARA Nobutada)
* 65597e0 草案を書いてみる (MATSUBARA Nobutada)
* 4295ad3 Add gitignore (MATSUBARA Nobutada)
* f3950a5 Initial commit (MATSUBARA Nobutada)
```

別名として設定しておくとなんと呼び出せる  
(`git config --global alias.tr “...”`)



# 歴史を重ねる



# タイムトラベル



# パラレルワールド



# 収束する歴史





# まとめ

- コミットなどを可視化する (`status/log/diff`)
- コミットを作る (`add/commit`)
- 古いコミット・ブロブに行き来できる (`checkout`)
- 複数のコミットを並列管理する (`checkout -b/branch`)
- 2つのコミットを合体させる (`merge`)

というコマンドが git にはある

