

# はじめてのGit

藤井 祥平 @ompugao

# Gitのしくみ

だいじなことは2つ

- commit / commit graph
- ファイル(tree/blob)の3状態

# commit / commit graph

## commitとは

- revision(SHA-1 hash)
- author(コミットをつくった人)
- committer(コミットを適用した人)
- ファイル(tree)のsnapshot
- 一つ前のrevision(SHA-1 Hash)

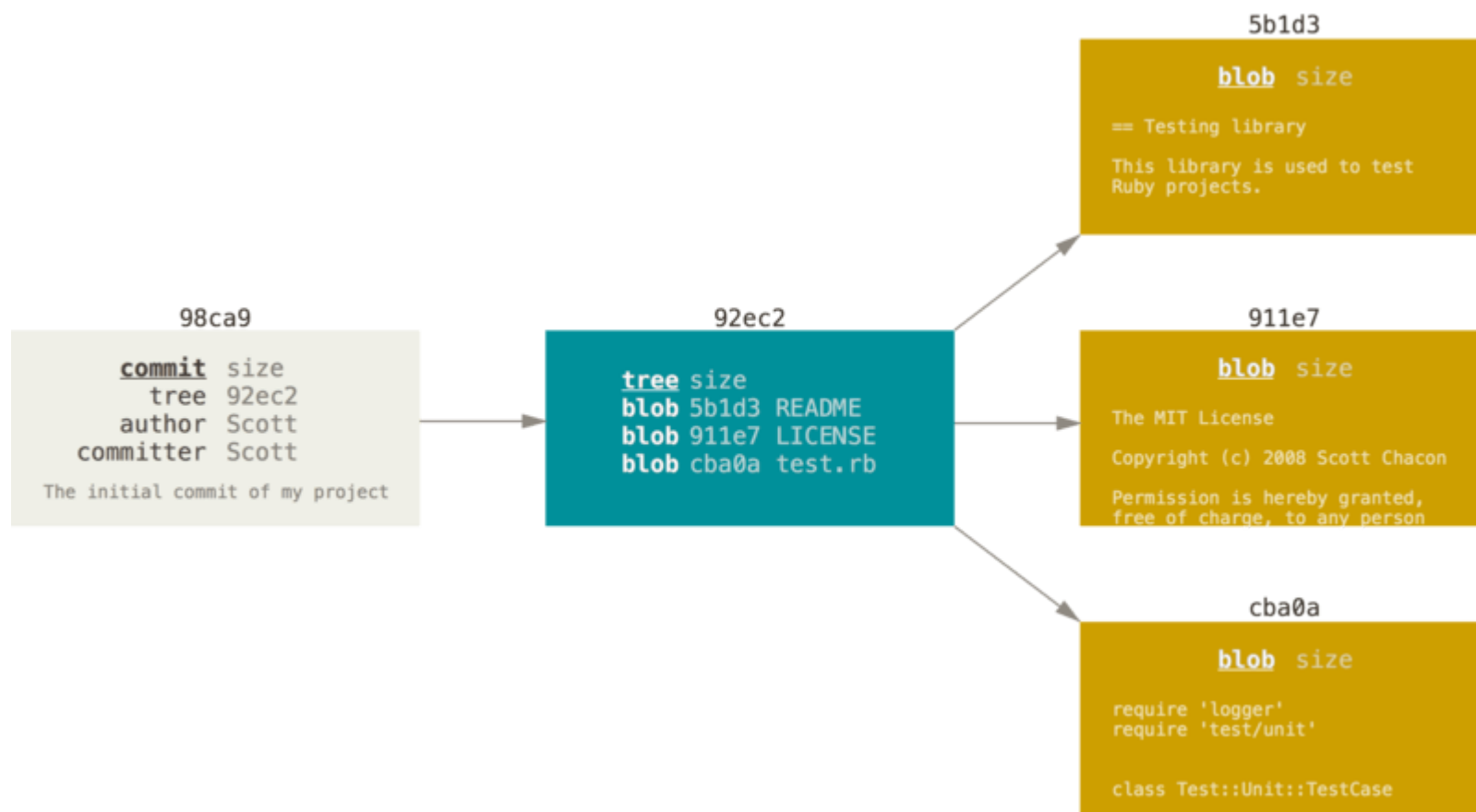
# commit / commit graph

## commit graphとは

- commitでできたツリー構造
  - commitに含まれていた "一つ前のrevision" をつかって木構造をつくっている

# commit / commit graph

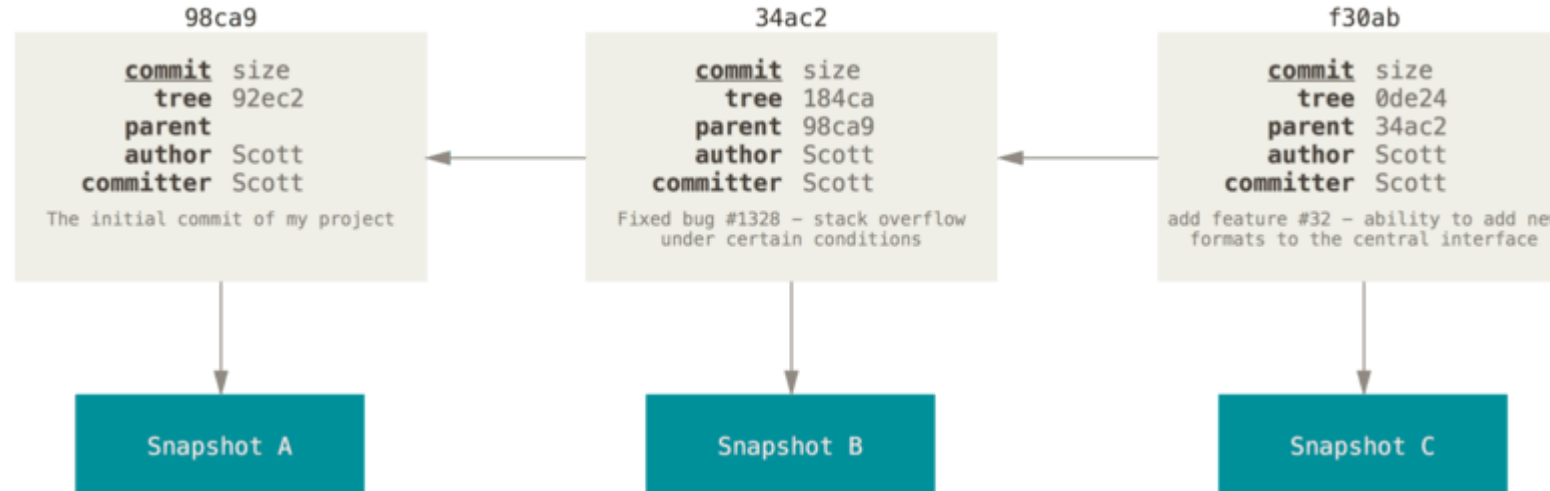
- commitにはrevision(hash)とディレクトリやファイルのスナップショットが入っている



quoted from: <https://git-scm.com/book/ja/v2/Git-のブランチ機能-ブランチとは>

# commit / commit graph

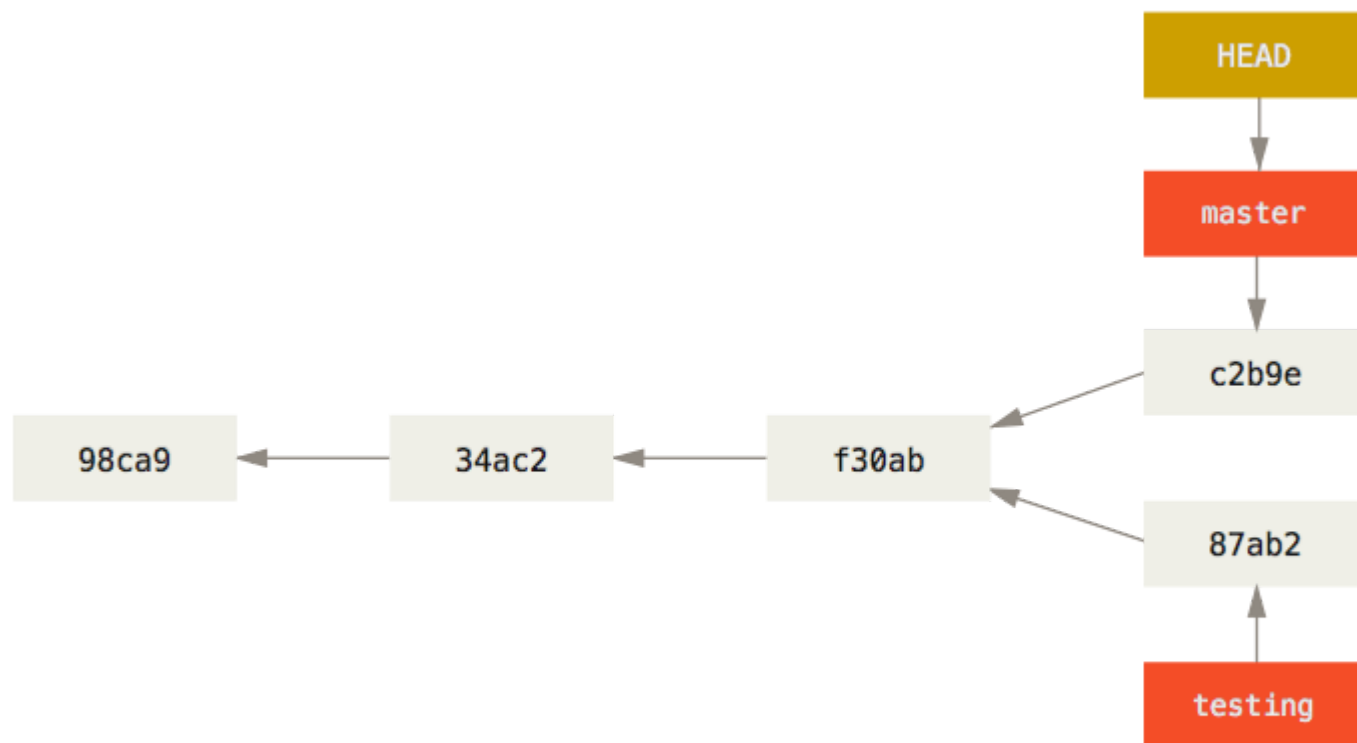
- 3commitくらいいつくると、commitが数珠繋ぎになる



quoted from: <https://git-scm.com/book/ja/v2/Git-のブランチ機能-ブランチとは>

## commit / commit graph

- branchを切るとこうなる



quoted from: <https://git-scm.com/book/ja/v2/Git-のブランチ機能-ブランチとは>

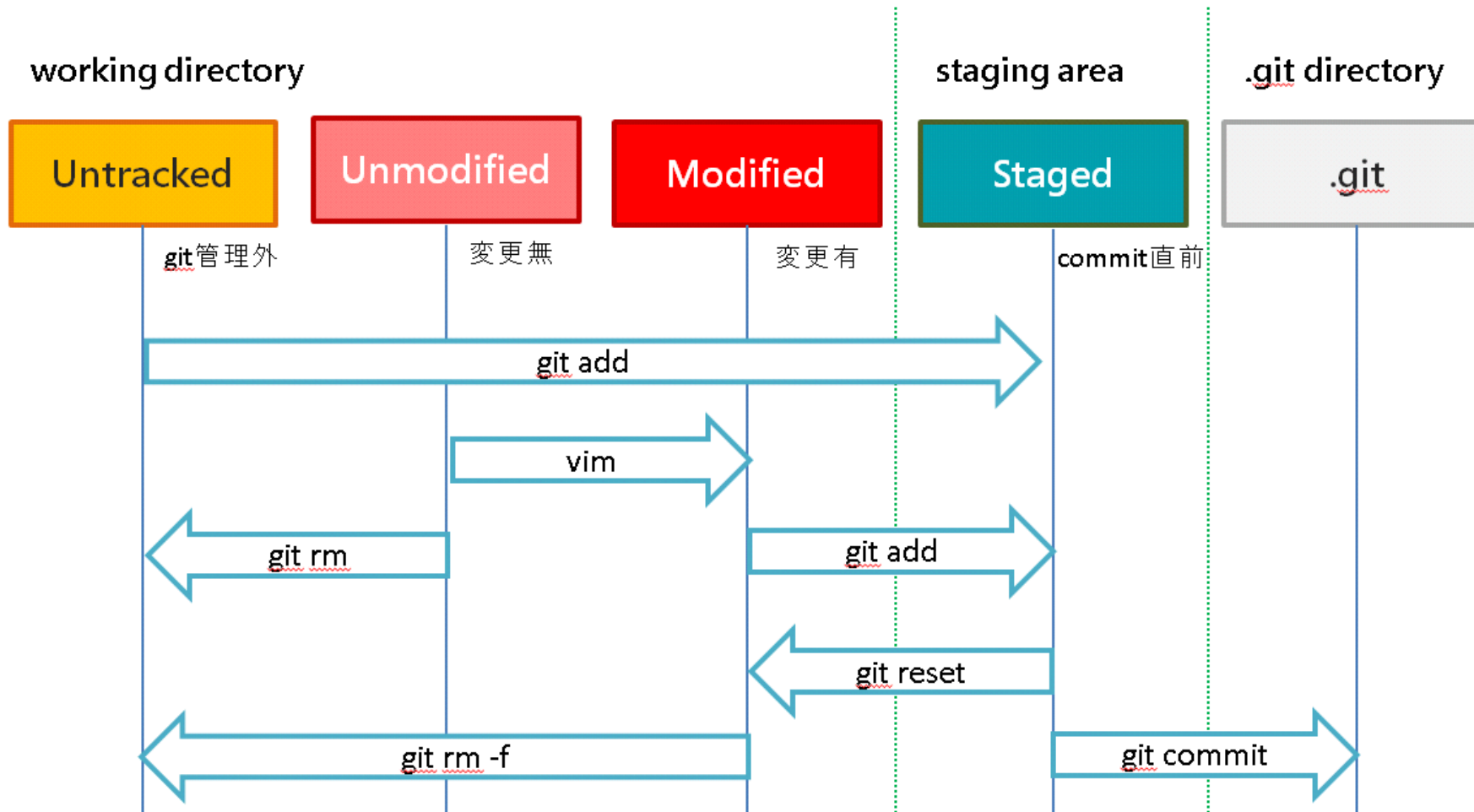
# ファイルの3状態

- 作業ディレクトリ(Working Directory)
  - 作業するところ
  - ↓ `$ git add`
- ステージング・エリア(Staging Area)
  - commitする準備段階
  - ↓ `$ git commit`
- Gitディレクトリ
  - commit graphがあるところ



# ファイルの3状態

- ファイルを編集するときのライフサイクルはこんな感じ



- まぎらわしいので書いてませんが、stagedなfileをgit commitするとunmodifiedになります

## 以上を念頭に

- try gitを試してみましょう
- <https://try.github.io/levels/1/challenges/1>

# Appendix

「こわくない Git」というスライドを発表しました - kotas.tech

<http://kotas.hatenablog.jp/entry/2012/11/22/000046>

見えないチカラ:【翻訳】Gitをボトムアップから理解する

<http://keijinsonyaban.blogspot.jp/2011/05/git.html>