



Signup

Login



@satoshi1335



Gitの基礎

Git

GitHub

bitbucket

入門

基礎

Git使えと言われたけどよく分かんないから調べてみた.

Git関連の記事見ても良く分からない言葉が多かったので用語もまとめてみた.

間違いがあれば指摘して頂けると幸いです.

参考文献

<https://git-scm.com/book/ja/v2>

http://programmerbox.com/2013-07-17_git_summary/

Gitとは？

代表的な「分散型バージョン管理システム」

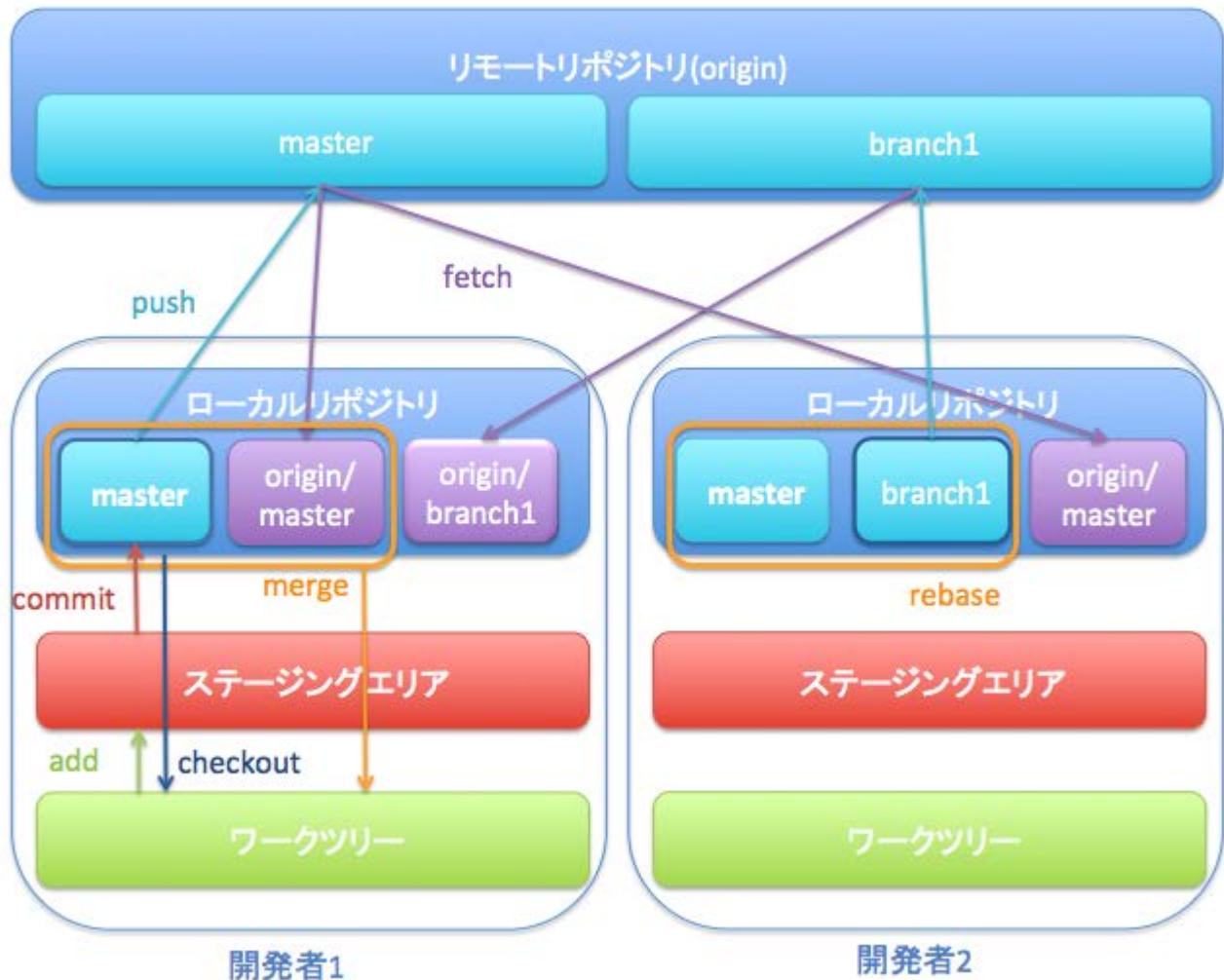
- 分散型： 自分のマシンにリポジトリ（開発過程が保存されるデータベース）を持っている
- バージョン管理システム：（複数人で）開発過程を記録しながら開発を行なうためのシステム

【用語】

- リポジトリ： ファイルやディレクトリの状態や変更履歴（開発過程）が保存されるデータベース

- (Gitでは連想記憶ファイルシステム??)

Gitの概念図



- リモートリポジトリで開発過程を共有する
- ローカルリポジトリを持つ事で、オフラインでの作業が可能に
- 開発過程（状態や変更履歴）のリモートリポジトリへの登録は**3段階**で行なわれる
 1. ワークツリーの変更を加えたファイルをステージングエリアに追加 (add)
 2. ステージングエリアのファイルをローカルリポジトリに登録 (commit)
 3. ローカルリポジトリに登録された情報をリモートリポジトリに追加(push)
- リモートリポジトリからの開発過程情報の取り寄せは後述

【用語】

- ローカルリポジトリ：自分のコンピュータ内にあるリポジトリ
- リモートリポジトリ：自分のコンピュータ外にある（複数人で共有する）リポジトリ
- ステージングエリア：ローカルリポジトリに登録するファイルやディレクトリのリスト
- ワークツリー：バージョン管理対象の（開発過程を記録する）ファイルやディレクトリ
- ディレクトリ：=フォルダ
- ステージング：ワークツリーの変更を加えたファイルをステージングエリアに追加すること (add)
- コミット：ステージングエリアのファイルをローカルリポジトリに登録すること (commit)
- プッシュ：ローカルリポジトリに登録された情報をリモートリポジトリに追加すること (push)

とりあえず**Git**を始めてみる

1. Gitをインストール

- <https://git-scm.com/book/ja/v1/%E4%BD%BF%E3%81%84%E5%A7%8B%E3%82%81%E3%82%8B-Git%E3%81%AE%E3%82%A4%E3%83%B3%E3%82%B9%E3%83%88%E3%83%BC%E3%83%AB>

2. ユーザー情報を設定

- `$ git config --global user.name ユーザー名`
 - `$ git config --global user.email メールアドレス`
 - または ~/.gitconfigを編集
-
- リモートリポジトリが存在しない場合

1. リモートリポジトリを作成

- **Github**を利用
 - <https://github.com/>
- **bitbucket**を利用
 - <https://bitbucket.org/>

- 自ら作成
 - http://qiita.com/nekogeruge_987/items/326ba157d5552f0076be
 - 2. 作業ディレクトリ（バージョン管理したいファイルがあるディレクトリ）に移動
 - `$ mv 作業ディレクトリ`
 - 3. **Git**ディレクトリ（リポジトリに必要なファイルが格納されているディレクトリ）を作成
 - `$ git init`
 - `$ ls -a` → `.git`（Gitディレクトリ）ができている事を確認
 - 4. リモートリポジトリを登録
 - `$ git remote add` 設定するリモートリポジトリ名 リモートリポジトリのurl
 - 5. 作業ディレクトリ内の（全ての）ファイルをステージングエリアにステージング
 - `$ git add ファイル名`（指定したファイルをステージング）
 - `$ git add -all`（全てのファイルをステージング）
 - 6. ローカルリポジトリにコミット（ステージングエリアのファイルをローカルリポジトリに登録）
 - `$ git commit -m “メッセージ”`
 - 最初のcommitで**master**ブランチ(開発の本流)が作成される
 - `$ git branch` → *master となっていることを確認
 - ブランチについては後述
 - 7. リモートリポジトリへプッシュ（ローカルリポジトリに登録された情報をリモートリポジトリに追加）
 - `$ git push` リモートリポジトリ名 リモートブランチ名
- リモートリポジトリが存在する場合
 - リモートリポジトリを自分のコンピュータにクローン（複製）（作業ディレクトリ・Gitディレクトリが自動生成される）
 - `$ git clone` リモートリポジトリのurl
 - ※リモートリポジトリ名はoriginとなる

これでGitを使い始める準備が整う

【用語】

- 作業ディレクトリ：バージョン管理したいファイルやディレクトリ（ワークツリー）があるディレクトリ

- **Gitディレクトリ**：リポジトリに必要なファイルが格納されているディレクトリ(.git)
- **クローン**：リモートリポジトリを自分のコンピュータに複製し，作業ディレクトリとGitディレクトリを自動生成する事

ブランチ

Gitには開発の本流から分岐し，本流の開発を邪魔することなく作業を続ける機能がある（ブランチ機能）

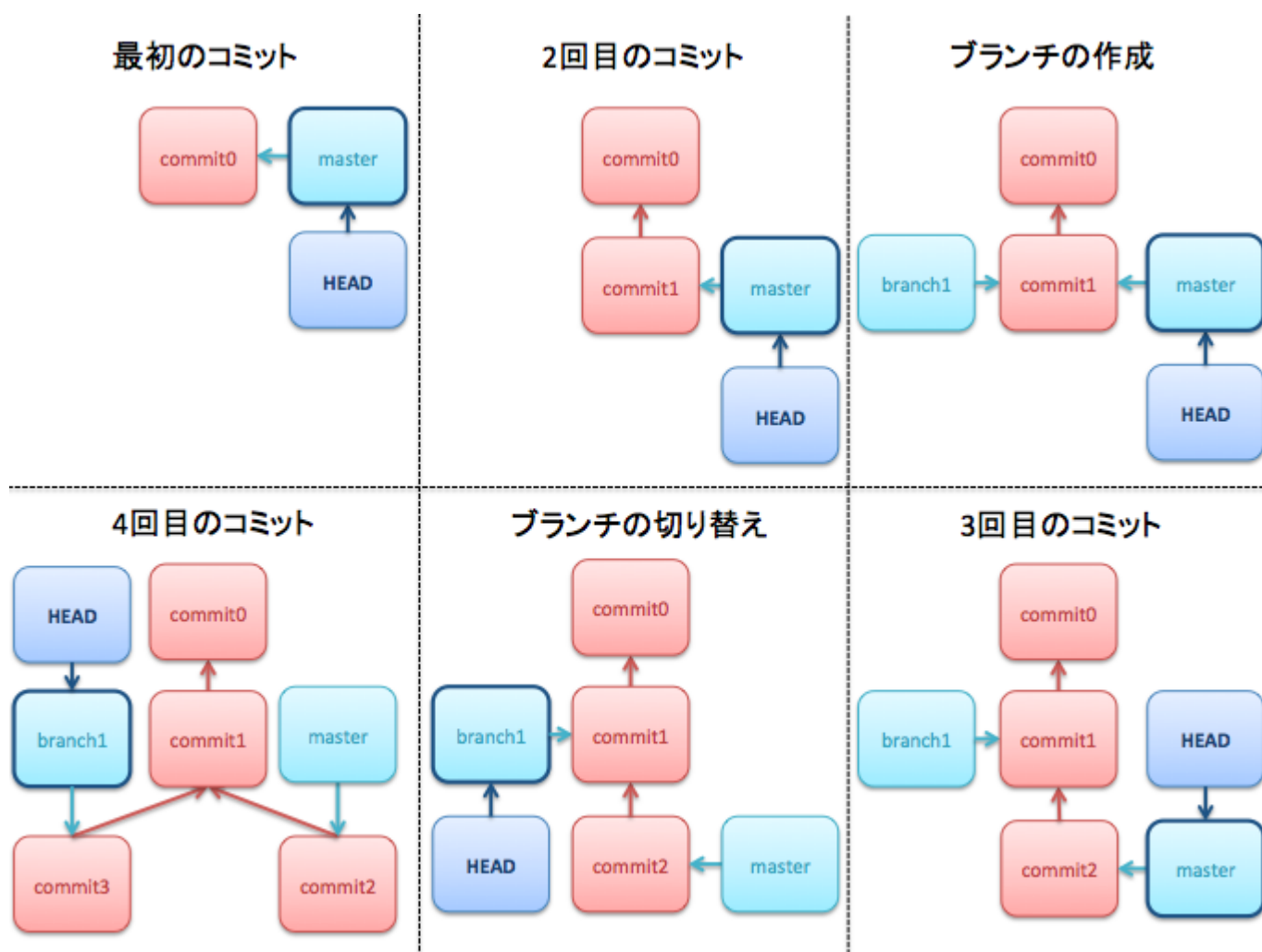
ブランチ機能の為の**3**つのデータ



- **コミットオブジェクト**：コミットに関する以下の要素を持つオブジェクト（データのまとめり）
 - **Key**：リポジトリに保存される名前（ステージングエリアのファイルからハッシュを用いて生成）

- **blob**：リポジトリに登録される、(変更が加わった)ファイルのスナップショット← (差分でない所がGitの特徴らしい)
- **Parent Key**：1つ前のコミットにより生成されたコミットオブジェクトのKey
- **Tree**：blobの集合を表すオブジェクト
- コミットオブジェクト（変更の加わったファイル）を辿って行く事で、過去の作業ディレクトリの状態を復元できる（コミットグラフ）
- **ブランチ**：直近のコミットで生成されたコミットオブジェクトを指すポインタ。複数作ることとで分岐を実現
- **HEAD**：作業中の分岐のブランチを指すポインタ

分岐までの流れ



1. 最初のコミット

- ステージングエリアのファイルからコミットオブジェクト(commit0)が生成され、ローカ

ルリポジトリに保存される

- 最初のコミットオブジェクトはParent Keyを持たない
- masterブランチ（開発の本流を示すブランチ）が生成され、ローカルリポジトリに保存される
- masterブランチが作業中の分岐に設定され、HEADはmasterブランチのKeyを指す

2. 2回目のコミット

- コミットオブジェクト(commit1)が生成され、ローカルリポジトリに保存される
- masterブランチのValueが新しく生成されたコミットオブジェクトのKeyに変わる

3. ブランチの作成

- `$ git branch ブランチ名` で新たなブランチ(branch1)を作成
- 作成されたブランチがHEADが指すブランチ=作業中のブランチ(master)が指しているコミットオブジェクトを指す

4. 3回目のコミット

- コミットオブジェクト(commit2)が生成され、ローカルリポジトリに保存される
- 作業中のブランチ(master)のValueが新しく生成されたコミットオブジェクトのKeyに変わる

5. ブランチの切り替え

- `$ git checkout ブランチ名` で作業するブランチを切り替え
- HEADのValueが指定されたブランチのKeyに変わる
- リポジトリから指定されたブランチが指すコミットオブジェクトまでの状態が読み出され、作業ディレクトリの状態を変更する(チェックアウト)

6. 4回目のコミット

- コミットオブジェクト(commit3)が生成され、ローカルリポジトリに保存される
- 作業中のブランチ(branch1)のValueが新しく生成されたコミットオブジェクトのKeyに変わる

branch1でコミットする事で、本流(master)の開発を邪魔せずに作業ができる

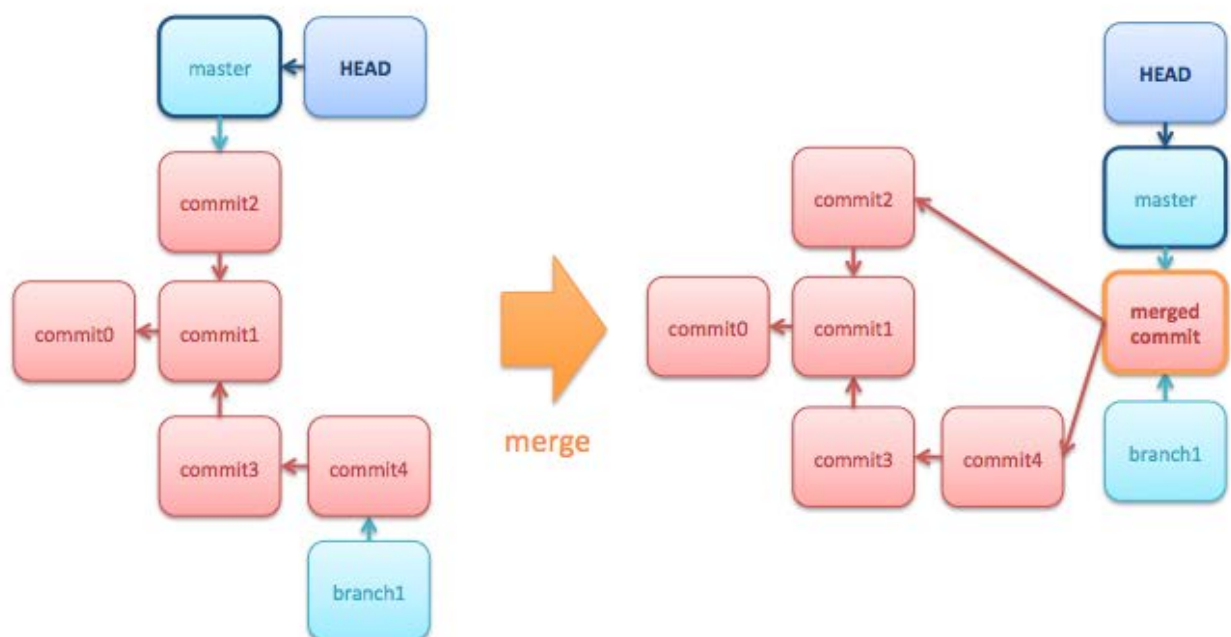
【用語】

- オブジェクト：データのまとまり
- スナップショット：変更が加えられるファイルのある時点抜き取ったもの
- コミットオブジェクト：コミットに関するオブジェクト(変更したファイル・開発過程)
- ブランチ：直近のコミットで生成されたコミットオブジェクトを指すポインタ
- **HEAD**：作業中の分岐のブランチを指すポインタ
- チェックアウト：リポジトリから作業ディレクトリにファイルやディレクトリをコピーすること
- コミットグラフ：コミットオブジェクトをグラフ形式で表現したグラフ（赤色のグラフ）

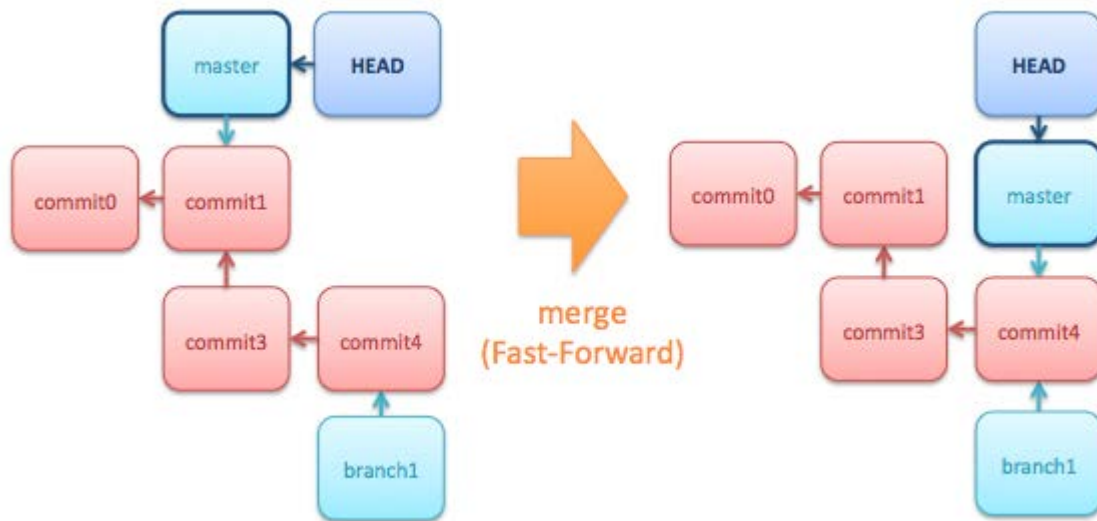
ブランチの統合

2つのブランチ（独立した変更）を1つの状態に統合する方法は2つある

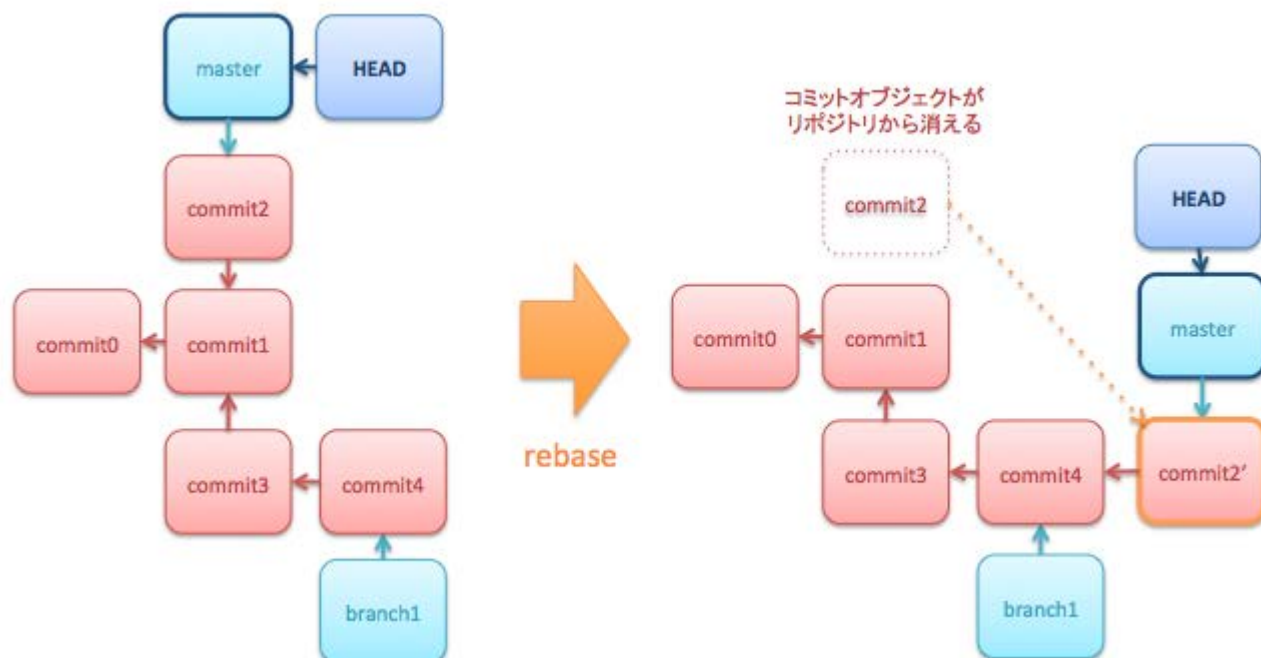
- マージ：両ブランチで変更されたファイルのスナップショットを再作成して、新しいコミットオブジェクトを自動生成する
 - `$ git merge ブランチ名` (現在のブランチと指定したブランチをマージする. 同時にチェックアウトされる.)



- **Fast-Forward**：統合するブランチに親を辿ってたどり着ける場合のマージ。上流にあるブランチが下流のブランチを指すようにすれば良い



- 異なるブランチで同じファイルを編集した場合コンフリクトが生じる（新しいコミットオブジェクトを自動作成できない）
 - ファイルを開いて修正し、そのファイルをステージング(add)すると、Gitはコンフリクトが解消されたと認識する
 - その後コミットすることで新しいコミットオブジェクトが生成される
- リベース：一方のブランチでの変更をもう一方のブランチで再現する
 - **\$ git rebase ブランチ名** 現在のブランチの変更を、もう一方の指定したブランチで再現する
 - コミットオブジェクトがリポジトリから消えるため、注意が必要



どちらの統合方法がいいかは場合によるらしい

【用語】

マージ：両ブランチで変更されたファイルのスナップショットを再作成して、新しいコミットオブジェクトを自動生成するブランチの統合方法

Fast-Forward：統合するブランチに親を辿ってたどり着ける場合のマージ（ブランチのValueを子にずらすだけ）

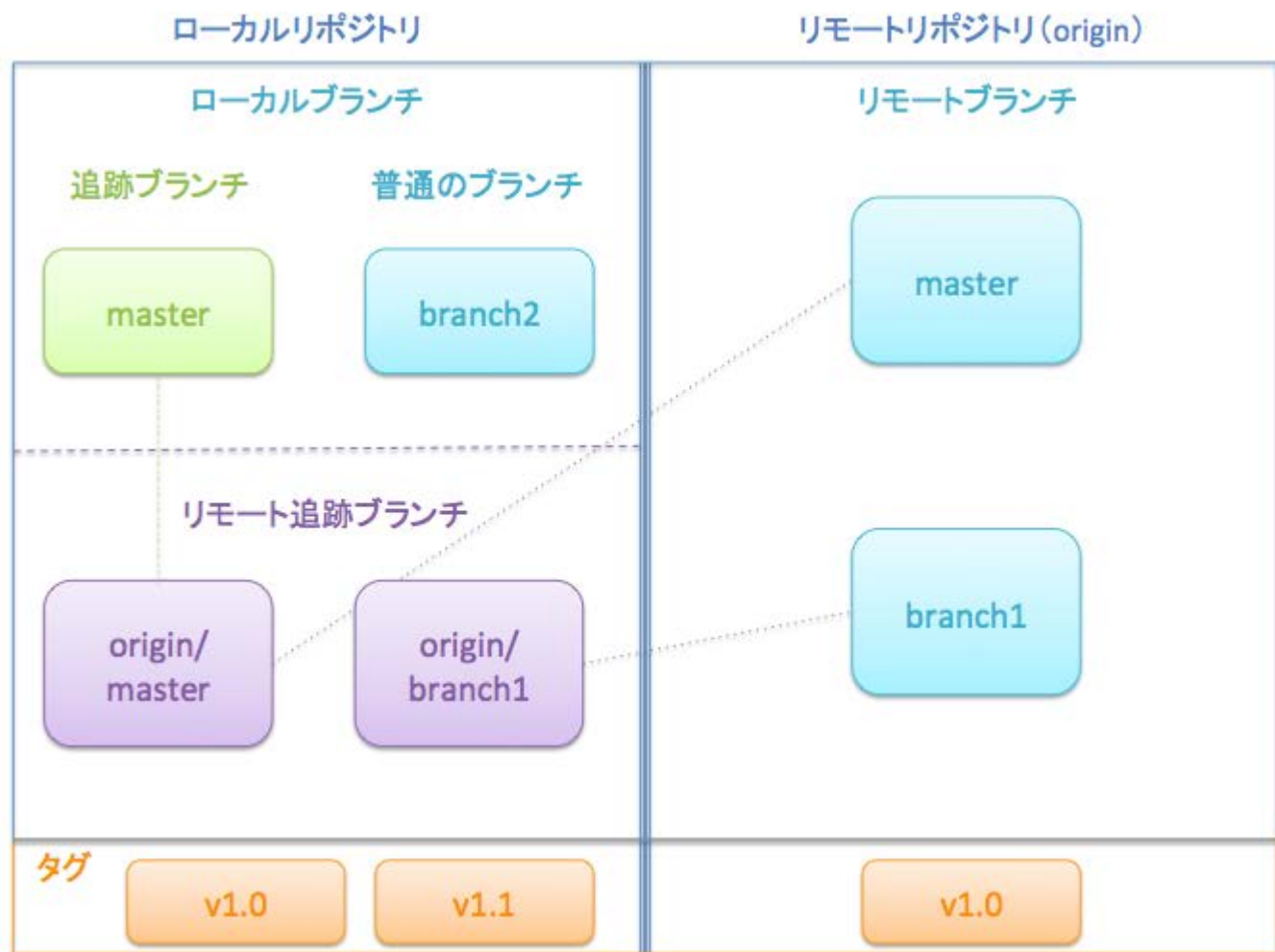
コンフリクト：異なるブランチで同じファイルを編集した場合、新しいコミットを自動作成できないこと

リベース：一方のブランチでの変更をもう一方のブランチで再現するブランチの統合方法

リモートリポジトリとのやり取り

ブランチの分類

リモートリポジトリ・ローカルリポジトリを考えたときに、ブランチは以下の様に分類できる



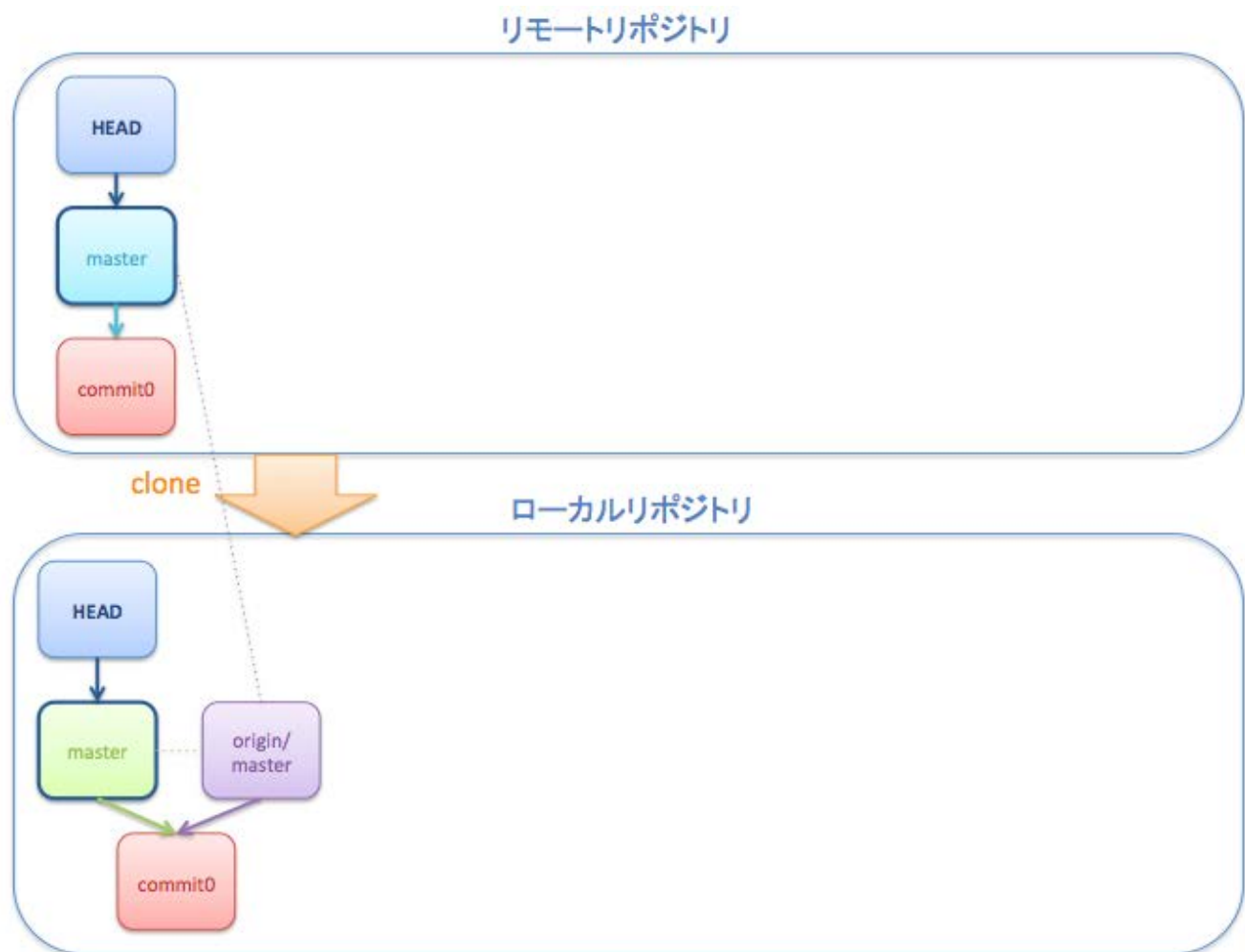
- ローカルブランチ：ローカルリポジトリのブランチ
- リモートブランチ：リモートリポジトリのブランチ
- リモート追跡ブランチ：リモートリポジトリとの接続時にリモートブランチが指していたコミットオブジェクトを指すローカルリポジトリのポインタ
 - ブランチ名は「リモートリポジトリ名/リモートブランチ名」となる ex) **origin/master**
 - リモート追跡ブランチに切り替える（リモート追跡ブランチのKeyをHEADのValueにする）事はできない
- 追跡ブランチ：プル（後述）の為にリモート追跡ブランチと対応づいたローカルブランチ
- タグ：ブックマークしたいのコミットオブジェクトを指すポインタ

4つのやり取り

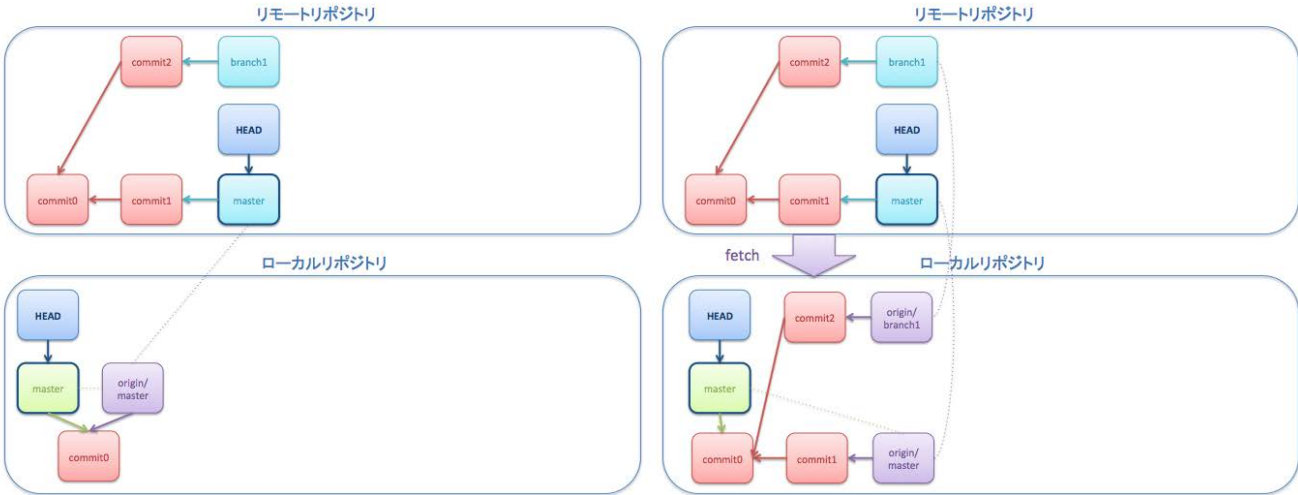
具体的なりモートリポジトリとの接続（やり取り）は以下の4つ

- クローン：リモートリポジトリを自分のコンピュータに複製し、作業ディレクトリとGitディ

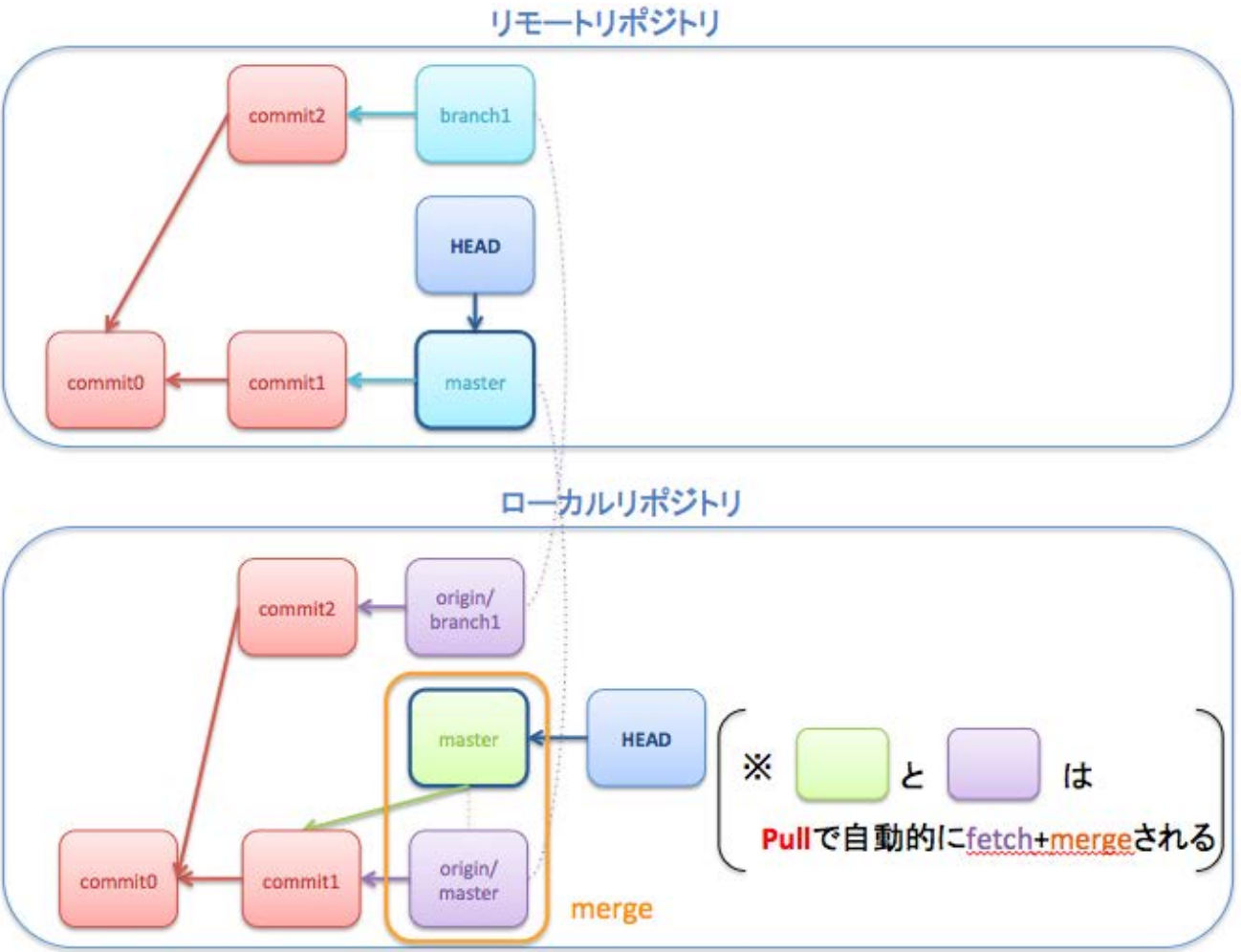
レクトリを自動生成する



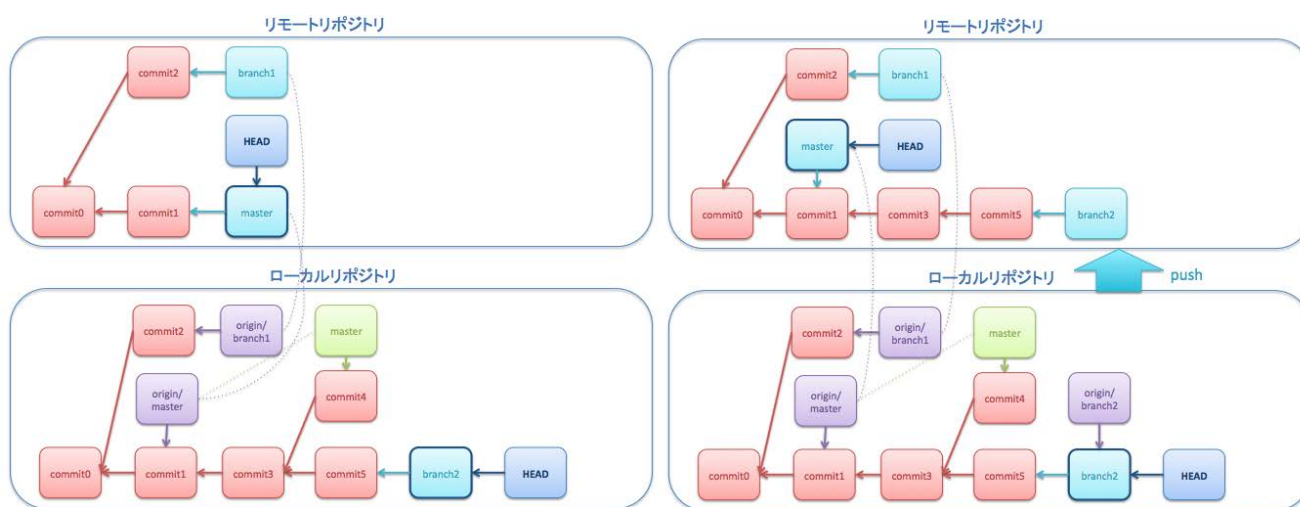
- クローン時に、リモート追跡ブランチ（リモートブランチが指していたコミットオブジェクトを指すローカルブランチ）が生成される
- 複製されたブランチは、リモート追跡ブランチの追跡ブランチとなる
- フェッチ：ローカルリポジトリにないリモートリポジトリのコミットオブジェクトをローカルリポジトリに追加する



- リモートリポジトリに新たなブランチがある場合、リモート追跡ブランチが追加される
 - リモート追跡ブランチのValueが、リモートリポジトリのブランチが指すコミットオブジェクトのKeyに変わる
 - フェッチした後、リモート追跡ブランチをマージする事で、作業ディレクトリがリモートリポジトリの最新の状態に置き換わる
- プル：フェッチ + 追跡ブランチと対応しているリモート追跡ブランチをマージする



- プッシュ：ローカルリポジトリの現在のブランチのコミットオブジェクトを指定したリモートリポジトリの指定したブランチ名で追加する



- プッシュすると、プッシュしたブランチと対応するリモート追跡ブランチが生成される
- リモート追跡ブランチの指すコミットオブジェクトと、対応するリモートリポジトリの指すコミットオブジェクトが異なる場合、プッシュできない
- その場合は、一度フェッチ+マージしてからリモートディレクトリに追加する

【用語】

- プッシュ：ローカルリポジトリのコミットオブジェクトをリモートリポジトリに追加すること
- フェッチ：リモートリポジトリのコミットオブジェクトをローカルリポジトリに追加すること
- プル：フェッチ + 追跡ブランチと対応しているリモート追跡ブランチをマージすること
- ローカルブランチ：ローカルリポジトリのブランチ
- リモートブランチ：リモートリポジトリのブランチ
- リモート追跡ブランチ：リモートブランチとの接続時にリモートブランチが指していたコミットオブジェクトを指すローカルリポジトリのポインタ
- 追跡ブランチ：リモート追跡ブランチと対応づいたローカルブランチ(プル用)
- タグ：ブックマークしたいのコミットオブジェクトを指すポインタ

 Like 187



satoshi1335