

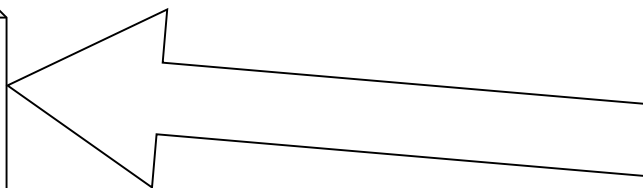






共同開

**開発の branch ② (remote)**













ea



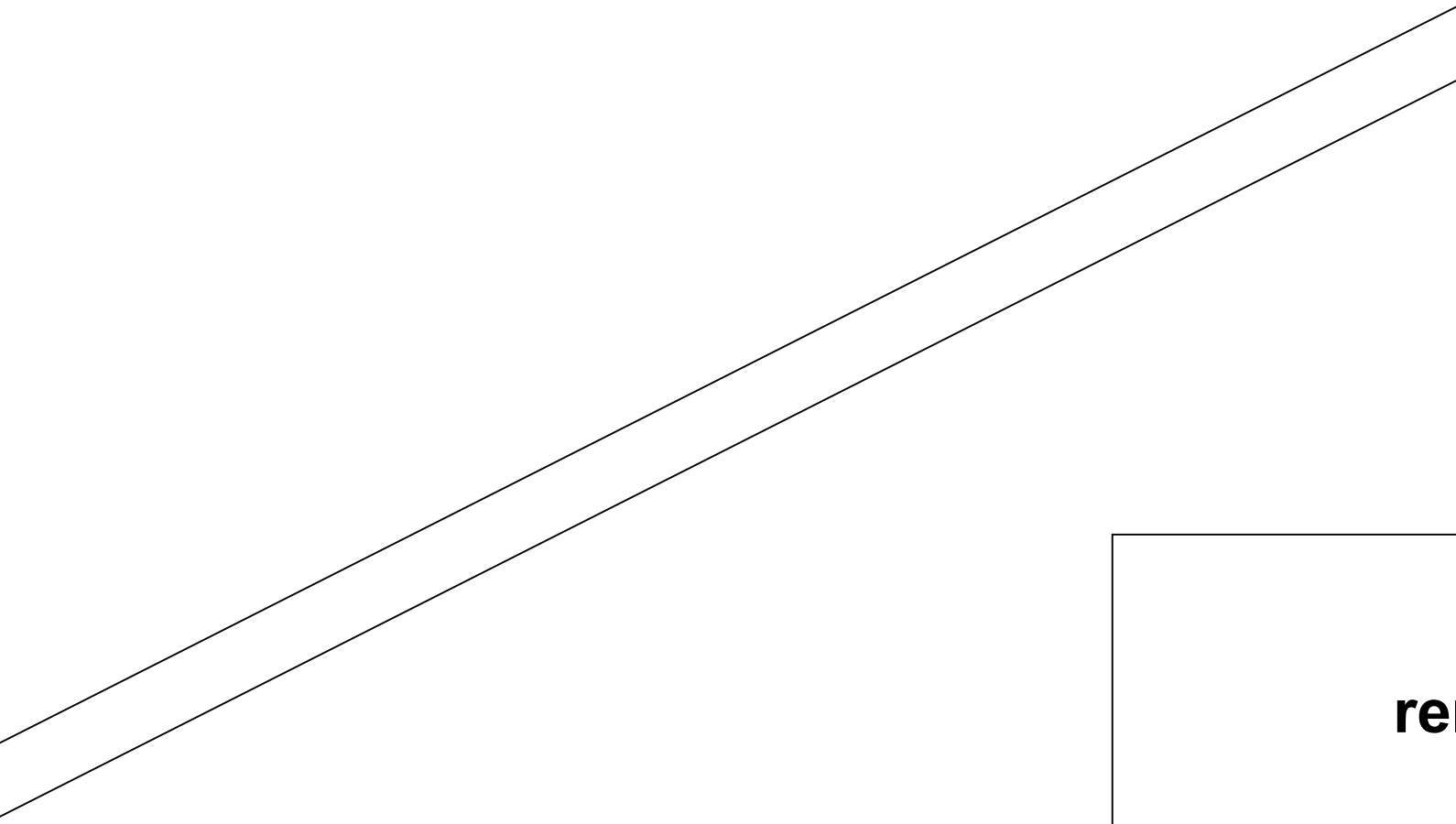
ren  
(jav  
変更

共同開発の branch ② (ea)

note の main  
ajava により  
更新されている)

remote の main

.java.java

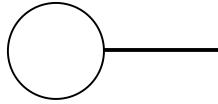


re

**git fetch origin <branch>**  
**+ git merge <branch>**  
**もしくは git pull origin <branch>:**  
remote ブランチの変更を local ブランチに  
統合 (main ブランチ)

**git merge :**

main



ea/home\_page .....

## remote と local の関係

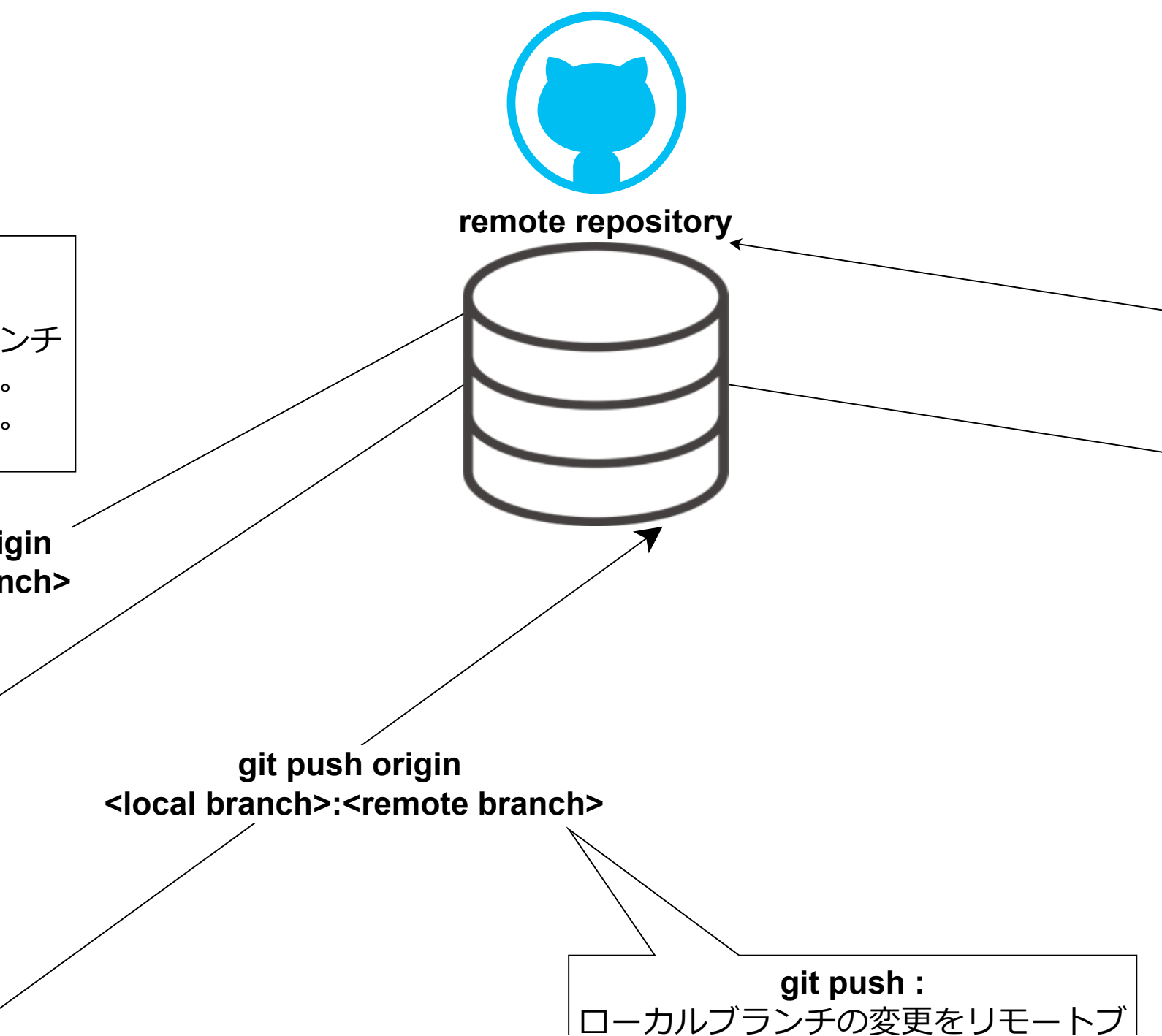
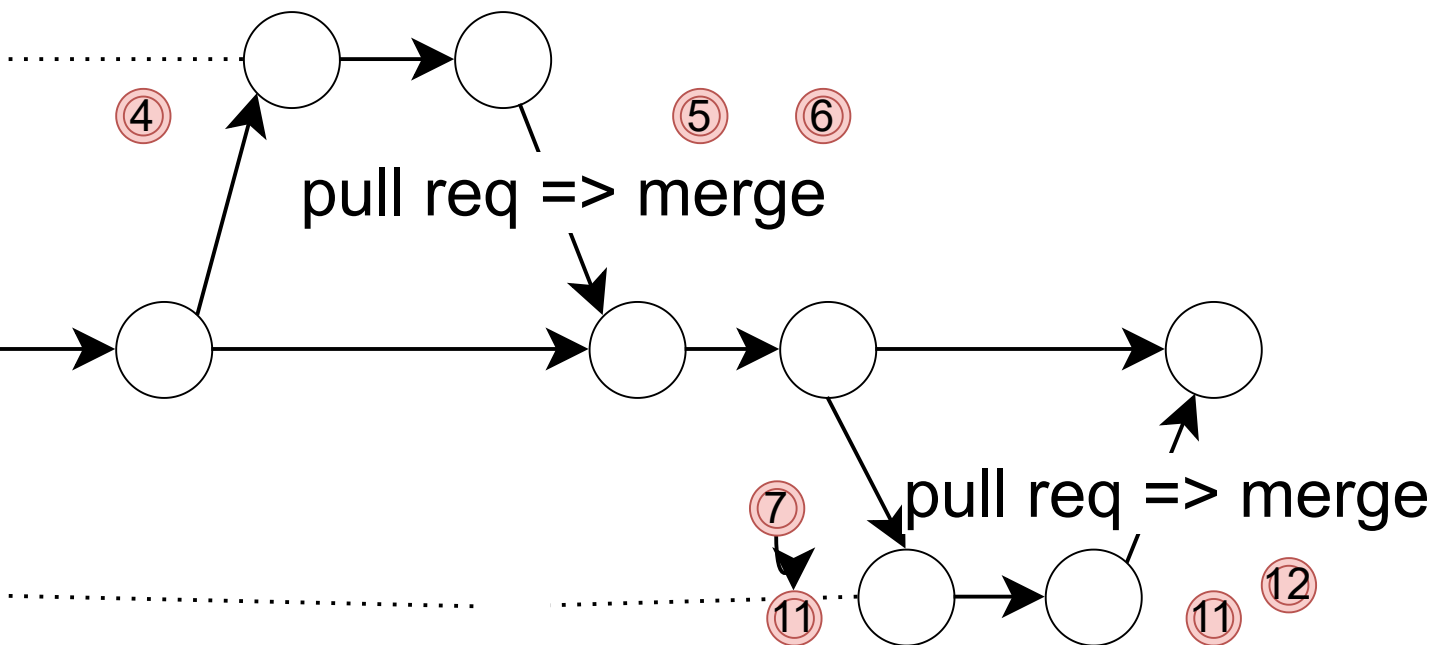
**git show origin/<branch>:**  
fetch した結果を表示して、  
現在のブランチとの差を見る

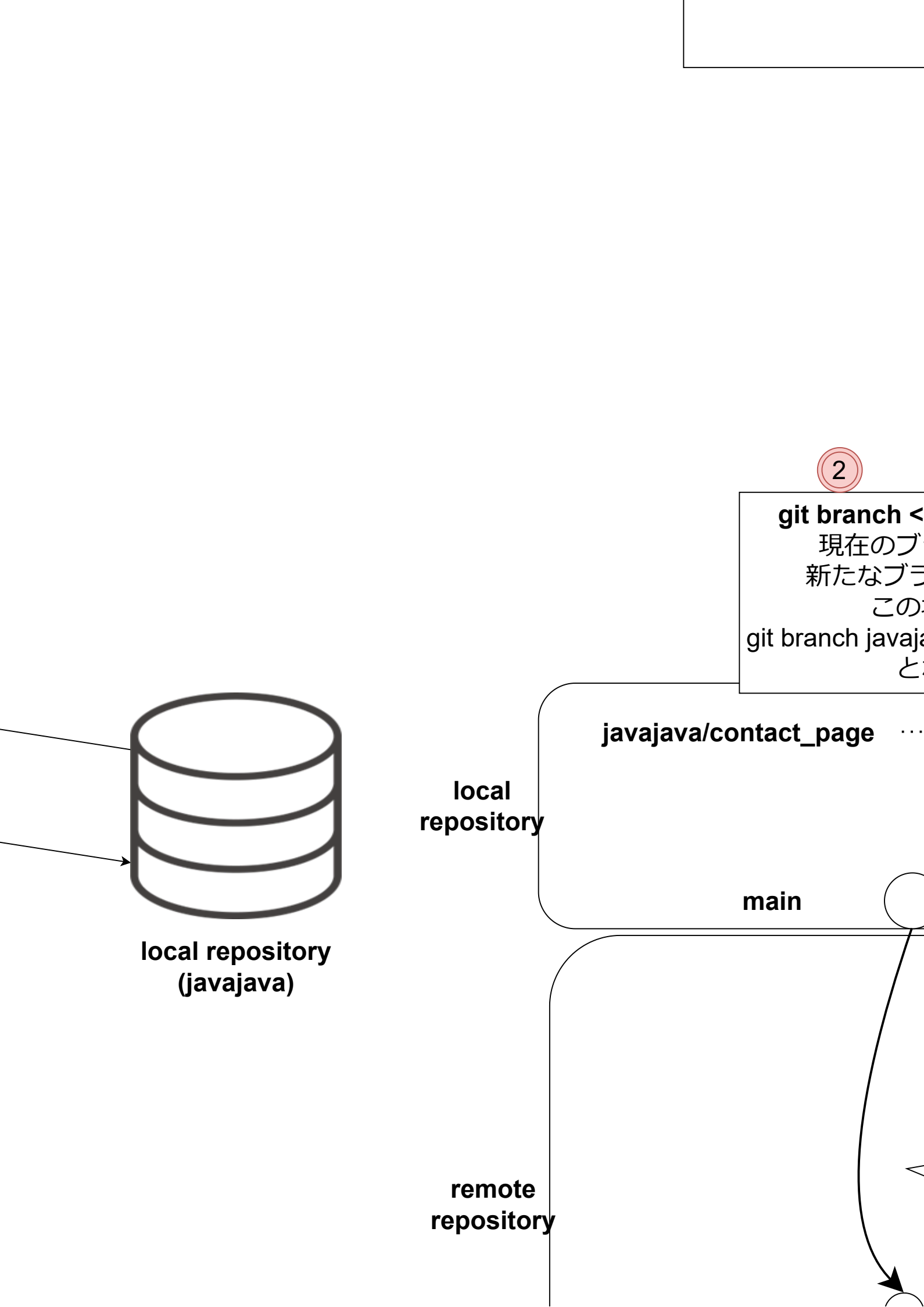
**git fetch :**  
リモートブランチから、  
ローカルのリモート追跡ブランチ  
に変更を適用するコマンド  
リモートの最新履歴を見る

**git fetch origin  
<remote branch>**

**リモート追跡ブランチ:**  
ローカルリポジトリにある、  
リモートリポジトリを  
追跡するブランチ

**git pull origin  
<remote branch>**







3

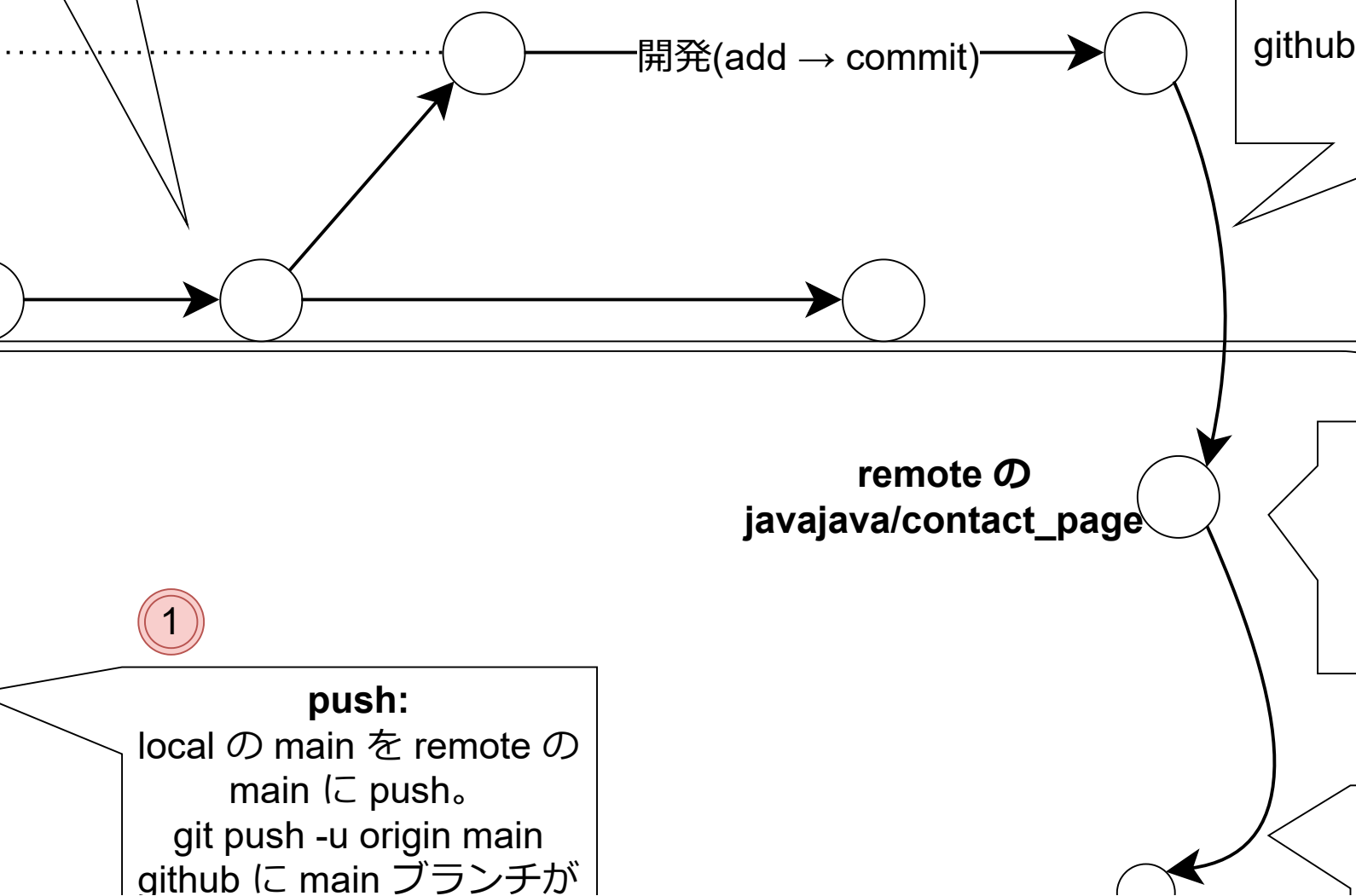
**git switch <branch>:**  
ブランチを移動する。  
この場合、  
`git switch javajava/contact_page`  
となる。またこれは  
`git checkout <branch>`  
と同じ

**new branch>:**

ランチから  
ランチを作成  
場合、  
`ava/contact_page`  
なる

4

local の  
remote  
github



1

**push:**  
local の main を remote の  
main に push。  
`git push -u origin main`  
github に main ブランチが

**push:**

の javajava/contact\_page を  
の javajava/contact\_page  
に push。  
に javajava/contact\_page  
が作成される

5

**pull request (github):**

github 上で pull request をして、  
merge する許可を得る。  
この場合は main に merge。  
master ← javajava/contact\_page

6

**merge (github):**

github 上で merge をして、  
ブランチを統合する



7

**git pull origin <branch> :**  
remote の master の変更を、  
local に直接 pull。

この場合は

git pull origin master  
になる。

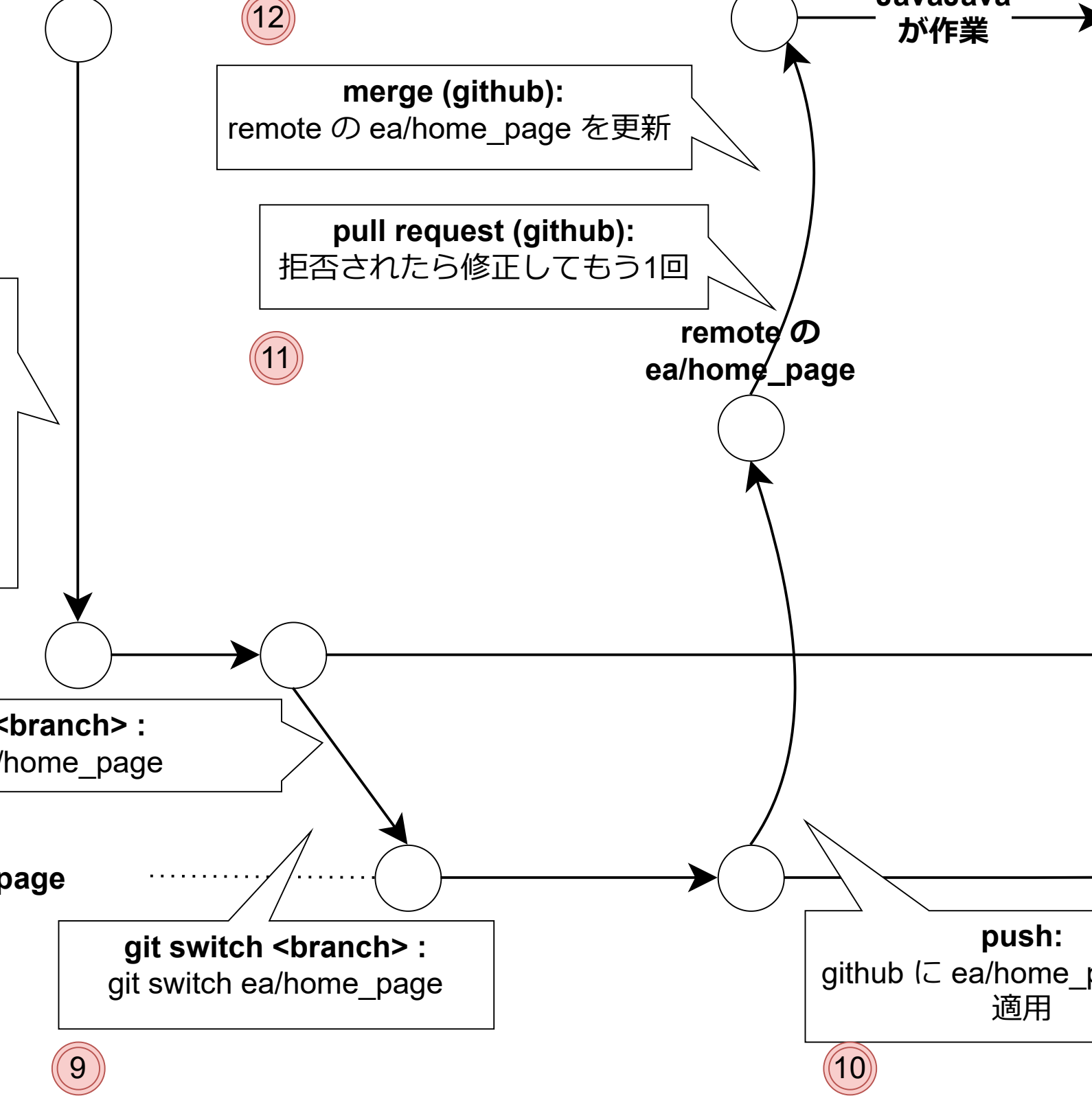
もし上流ブランチが設定されていれば  
git pull のみでよい。

main

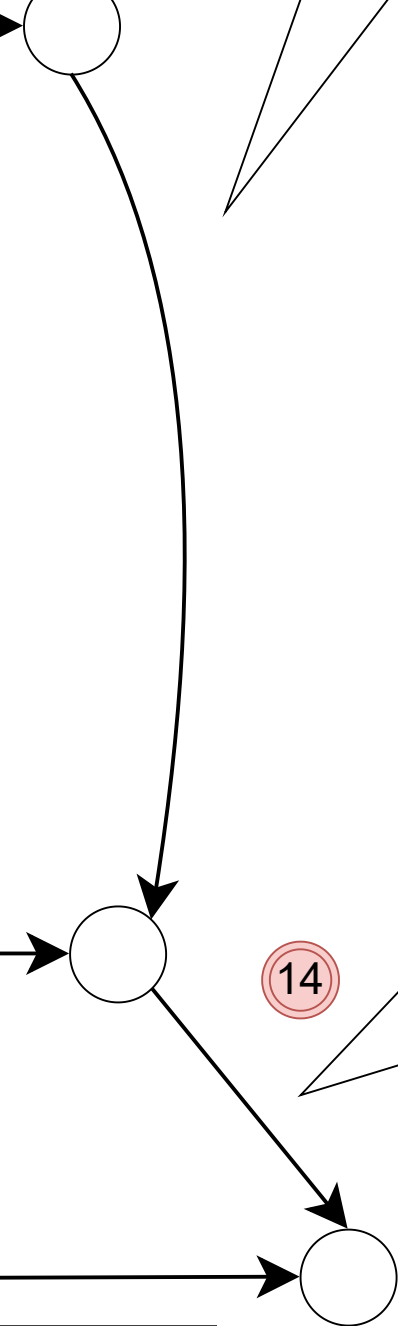
**git branch <branch>**  
git branch ea/

8

ea/home\_



あるローカルブランチの最新の  
コミットと現在のブランチを  
統合させる。  
この場合は  
`git merge origin/<branch>`  
とする。



**git merge <branch>:**  
ブランチの状態を更新。  
ここでは main => ea/home\_page なので  
ea/homepage に switch してから  
`git merge main`

page 作成

の  
を

origin/<branchname>  
ブランチ

fetch  
リモートブ  
在のブランチ

git merge <branch>

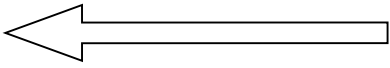
現在の  
ブランチ



local repository  
(ea)

変更があったもの  
だけを  
コミット

git commit



## git pull :

+ merge = pull

ランチの変更を自分の現  
ランチに一気に取り込む。

ランチに適用させるコマンド。

始めのプッシュで

git push -u origin <branch>

として**上流ブランチ**を設定すれば、  
以後 git push だけでよい。

## 上流ブランチ(upstream):

他のブランチに追跡されるブランチ。

git push -u origin main では、  
main ブランチはorigin.main を追跡する

ステージングエリア  
にある追跡対象のフ  
ァイル一覧を表示

git ls-files

git add で  
その時の常態で  
ステージングエリアに  
追加される  
(変更したら再度 git  
add する)

git add

add していないファイルを指  
定のコミットと同じ状態に  
する。

ex) git checkout HEAD a.txt

git checkout  
<commit id> <file path>



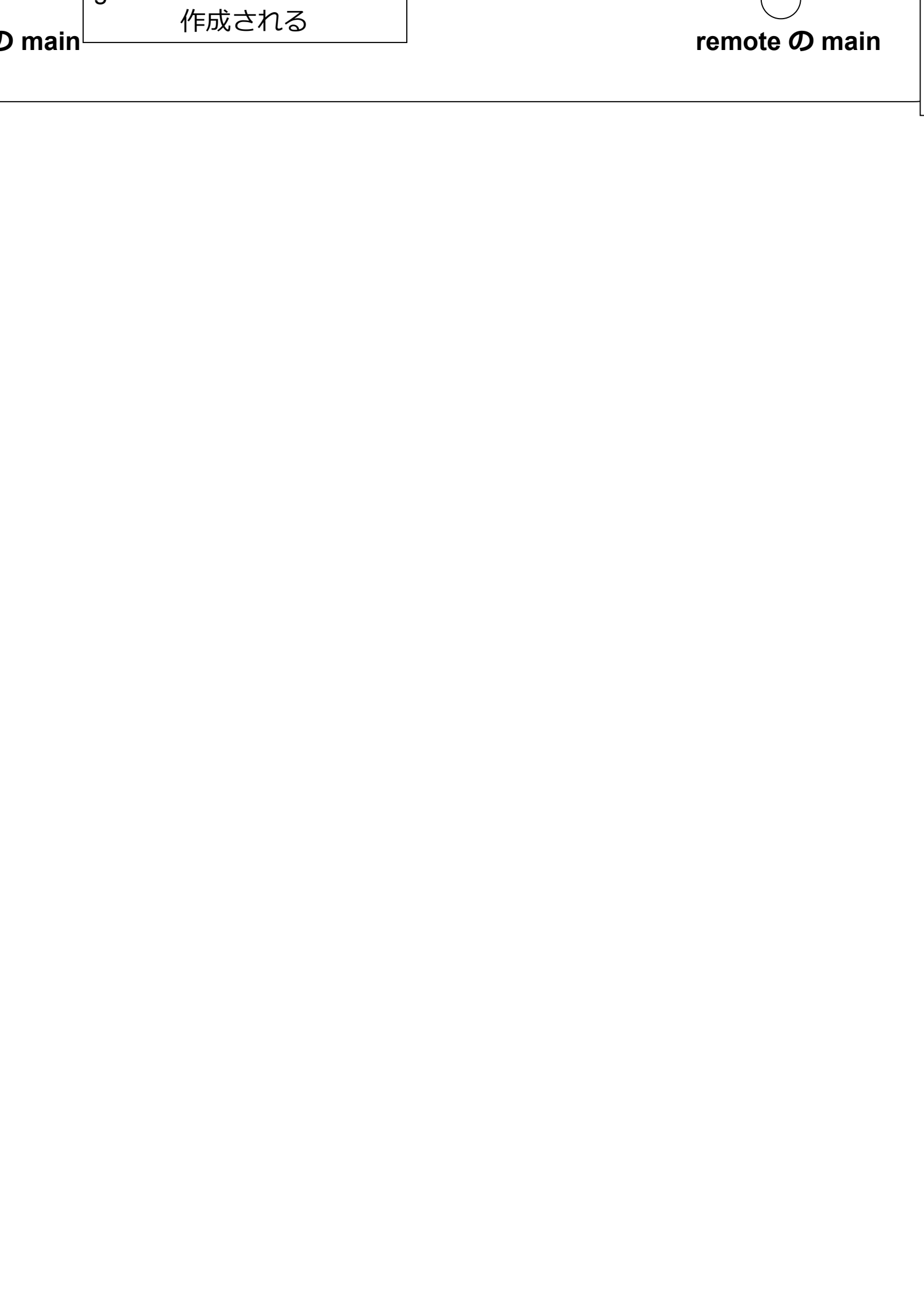
ステージングエリア



Text

### 一番最初

- **git init**
  - 何もないところから始める
  - その場合、
    - **git branch -M main**  
でデフォブランチをgithub用に  
変更
    - **git remote add origin <URL>**  
でリモートリポジトリを  
originとして追加する必要
- **git clone <URL>**
  - リモートリポジトリを  
コピーして始める



フランチを統合する  
この場合は main に  
javajava/contact\_pages  
の変更を適用する



個人開発 -  
(適宜 re

m

ea/hc

での local branch  
(remote へ push)

**git branch <branch> :**  
現在のブランチから別のブ  
ランチを派生させる。  
ここでは  
git branch ea/home\_page

ain

me\_page

**git switch <branch> :**  
ブランチを移動する。  
ここでは  
git switch ea/home\_page

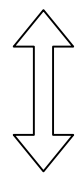
**git branch -avv :**  
リモート追跡ブランチを含  
めたすべてのブランチの詳  
しい情報がわかる



**git merge <branch>:**  
あるローカルブランチの最新の  
コミットと現在のブランチを統合させる。  
この場合は  
`git merge ea/home_page`  
これで master に ea/home\_page の変更が更新される

**git switch <branch> :**  
ブランチに戻りたいので、  
`git switch main`

ea のPC



git diff HEAD^



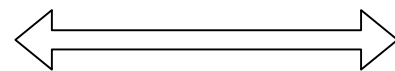
古い  
コミット

git reset --soft

git reset --mixed

git reset --hard

ローカルリポジトリ



git diff --staged

最新のコミットと  
インデックスを  
比較

ス

もし  
監視  
ファ

**HEAD:**

現在のブランチの  
最新のコミットを  
指している。

^ が付くほど古いコ  
ミットになる

インデックス

git reset

ea

ステージングエリア  
インデックス

git diff

作業ディレクトリ  
ワーキングツリー

インデックスと  
現在の作業ディレクトリ  
の違いを比較  
1回も add してないファイル  
は表示されない

git status

差があればどちらも表示  
対象になったことがない  
ファイルがあればそれも表示

- **git reset [<filename>]**
  - ステージングエリアをクリア
  - ファイル名を指定したら、そのファイルの add だけ取消
- **git reset --soft <commit id>**
  - 現在の状態(HEAD)を指定したコミットと同じ状態にする
  - `git reset --soft HEAD^` で直前のコミットを取消
  - ステージングエリアは消えずに残る
- **git reset --mixed <commit id>**
  - --soft と似ているがステージングエリアも変化
  - `git reset --mixed HEAD^` で直前のコミットを取消、ステージングエリアも削除
- **git reset --hard <commit id>**
  - 上記2つと似るが、ファイルも指定されたコミットと同じ状態になる
  - add 前のファイルを一齐に削除する。その場合は `git reset --hard`

アする  
そのファイ

されたコミ

ない

ジングエリ

ステージン

レ自体も指  
態にする。  
訂正可。  
HEAD













# local でのバージョン管理の仕方

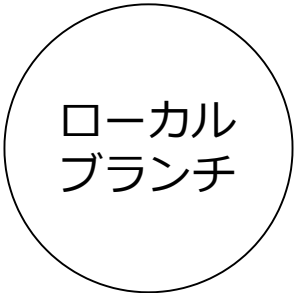













ローカル  
ブランチ



ローカル  
ブランチ