



IT日本語(2)

4.Git

授業の目的と目標

目的

- ☑ Gitの^{がいねん}概念を^{りかい}理解する
- ☑ Gitの用語(カタカナ語)が分かる

目標

- ☑ Gitの基本的な^{そうさ}操作を^{おこな}行える
- ☑ Gitを使って日本語の課題を^と解くことができる



目次

1. 環境^{かんきょうじゅんび}準備

2. ファイルの^{かんり}管理

3. Gitとは

4. Gitの^{りようれい}利用例(1)

5. Gitの利用例(2)

6. Gitの^{ようご}用語

7. Gitの^{じつれい}実例

8. 実習^{じっしゅう}(1)

9. 実習(2)

10. 実習(3)

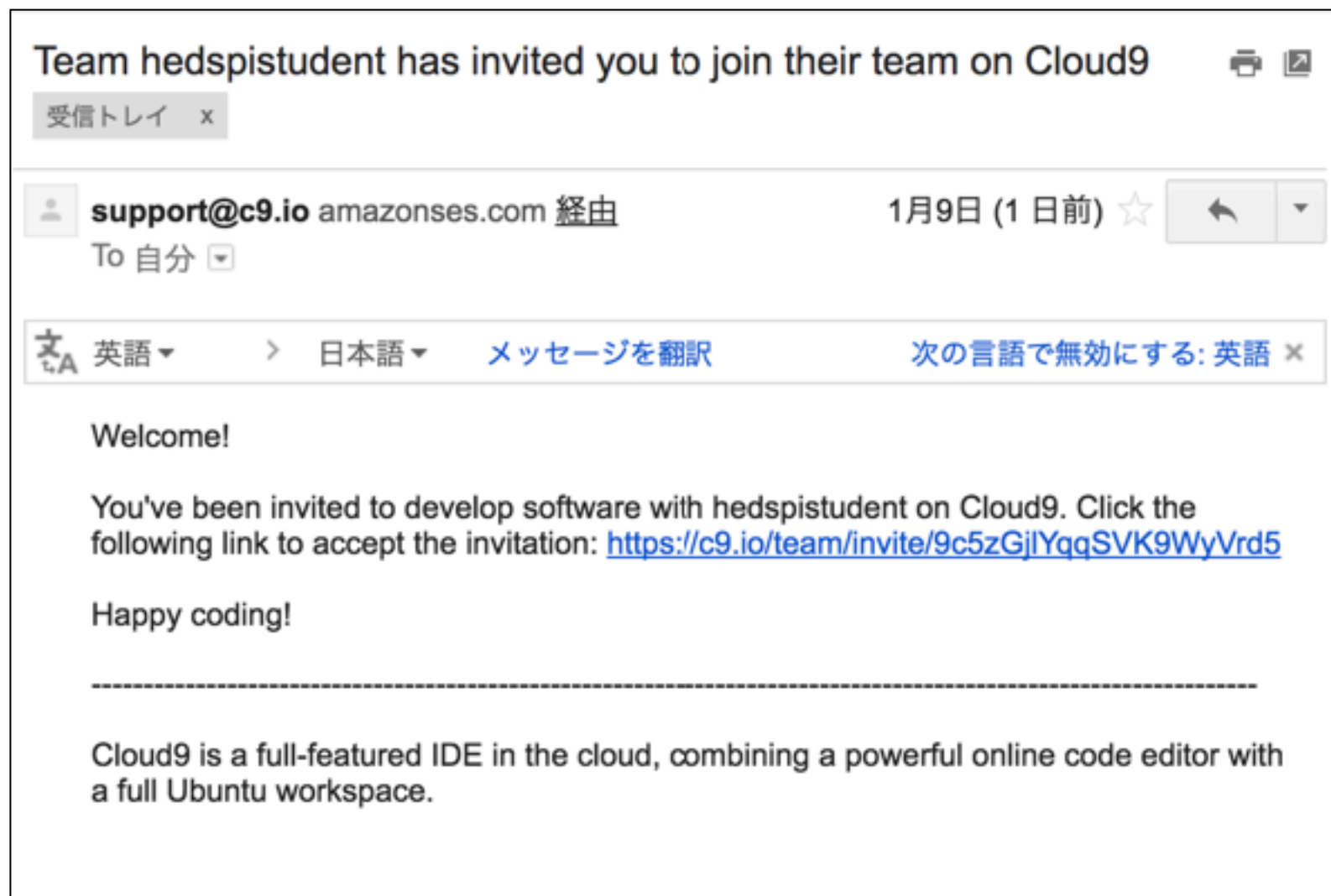
11. まとめ



環境準備 cloud9

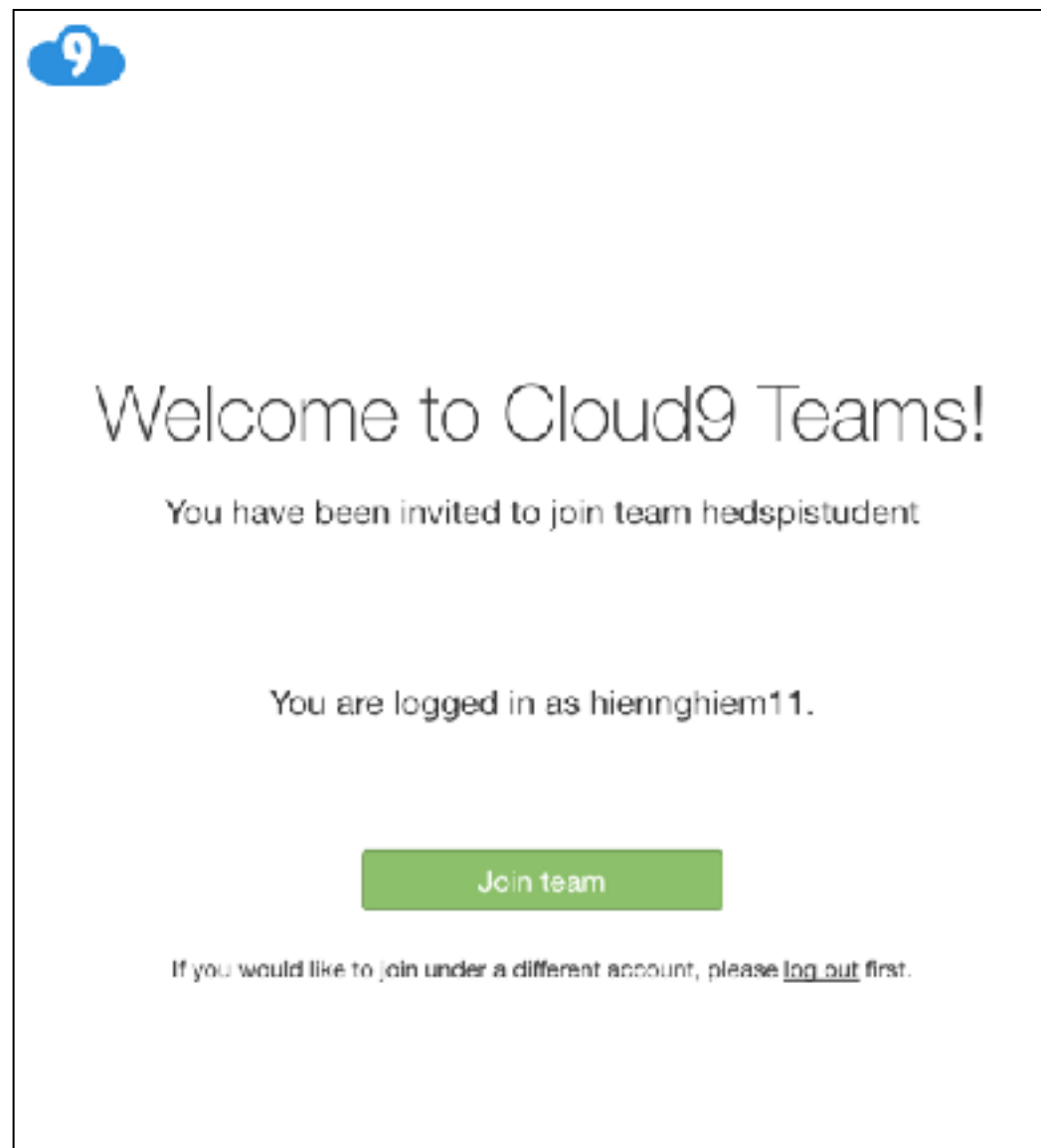
環境準備

今回の授業からcloud9を使います。
事前に皆さんにはcloud9から招待メールを送っています。



環境準備

かんたん しつもん
簡単な質問を入力しましょう。

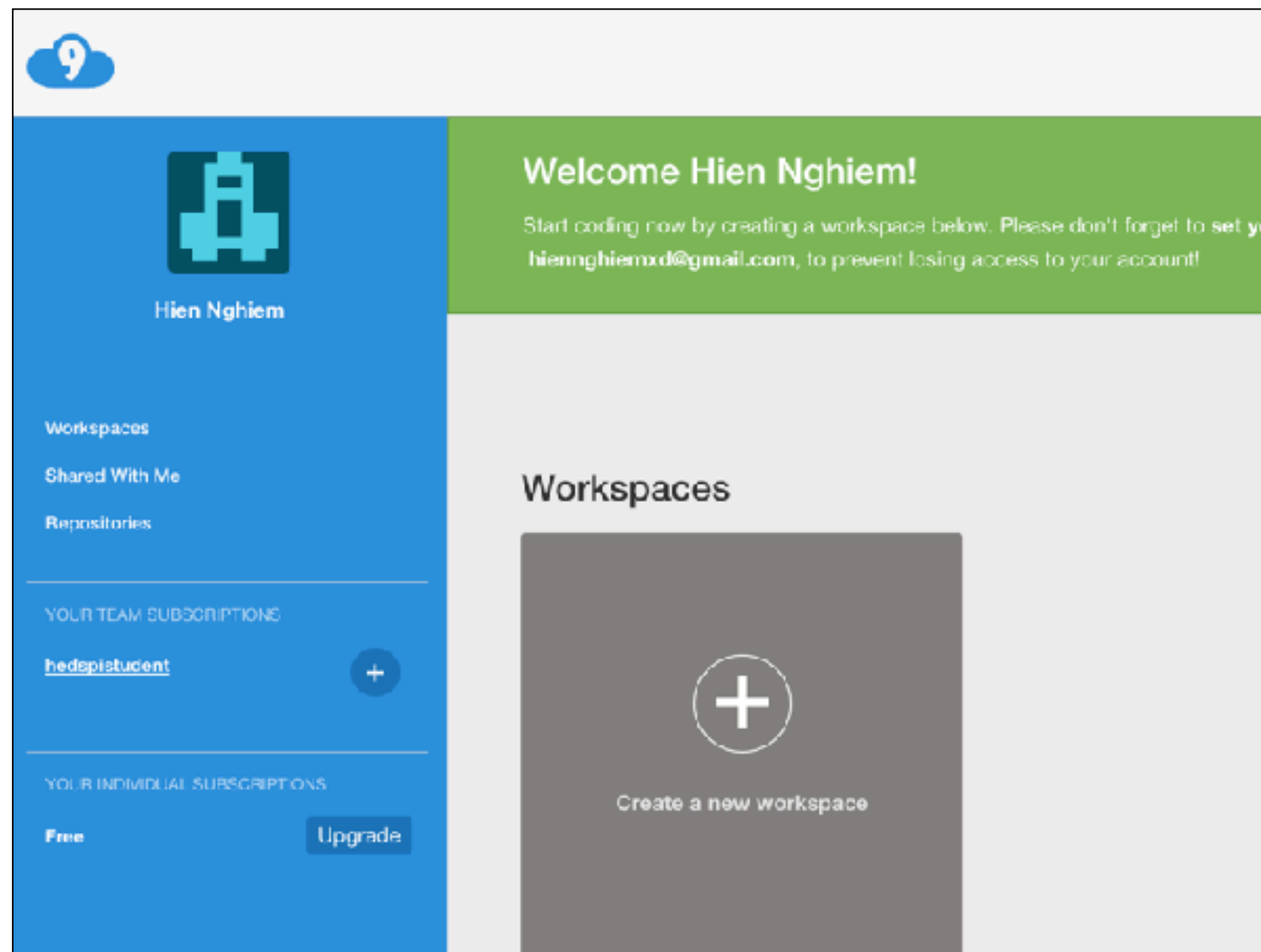


- 名前
- ユーザーネーム
- ^{かいはつ}開発目的
- ^{りょう}利用目的

環境準備

ダッシュボードが表示されたら準備完了です。

かんりょう



The background of the slide is a composite image. It features Mount Fuji in the distance, its peak covered in snow and partially obscured by soft, white clouds. In the foreground, there are dense, blooming cherry blossom trees with light pink flowers. To the right, a portion of a traditional Japanese temple with multiple tiers of dark green tiled roofs and red wooden railings is visible. The entire scene is set against a clear blue sky.

Git

ファイルの管理

開発でファイルを^{へんしゅう}編集する時、どのようにしていますか？
もっと ^{てがる}手軽なのは、あらかじめ編集前のファイルをコピーしておく^{ほうほう}方法
です。



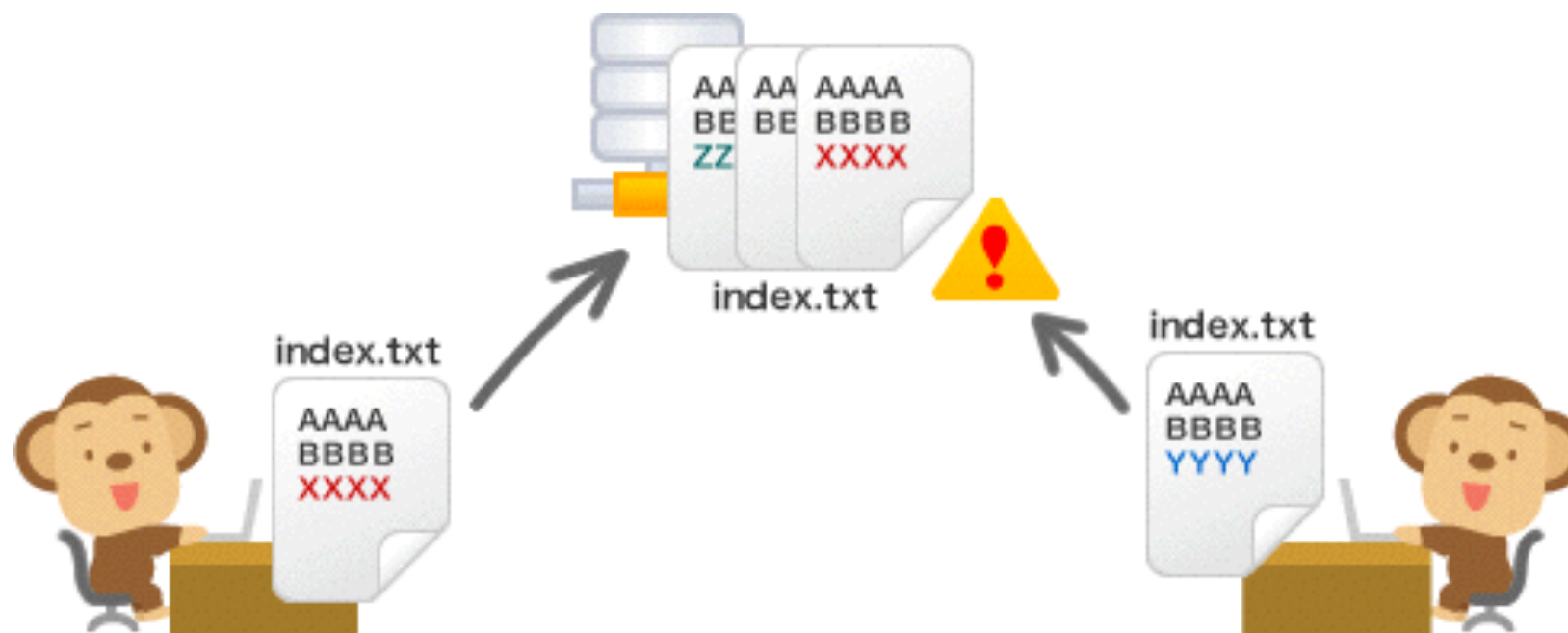
A list of text files, each preceded by a small icon of a document with a purple flower. The files are:

- 180123_index_ver2.txt
- 180123_index.txt
- 180124_index_sanada.txt
- 180124_index_ver2.txt
- 180124_index_ver2のコピー2.txt
- 180124_index.txt
- 180130_index_master.txt
- 1801230_index.txt

あと^み後で見ると
「どれが^{さいしん}最新」
だれ^{だれ}が編集した」
のかわからない・・・

ファイルの管理

チームで共有しているファイルの場合は、二人で同時に編集してしまっただけのために、先に編集した人の変更内容が消えてしまったという経験はないでしょうか？



Gitはこの問題を解決するために作られたシステムです

Gitとは

「Gitは分散型バージョン管理システム」です
以下のようなことができます

1. 更新したファイルの履歴管理
2. 古いソースコードへの復元
3. ソースコードの配布
4. ファイルの競合の防止
5. 複数のブランチの作成
6. 適応力の高い開発

Gitの利用例(1)

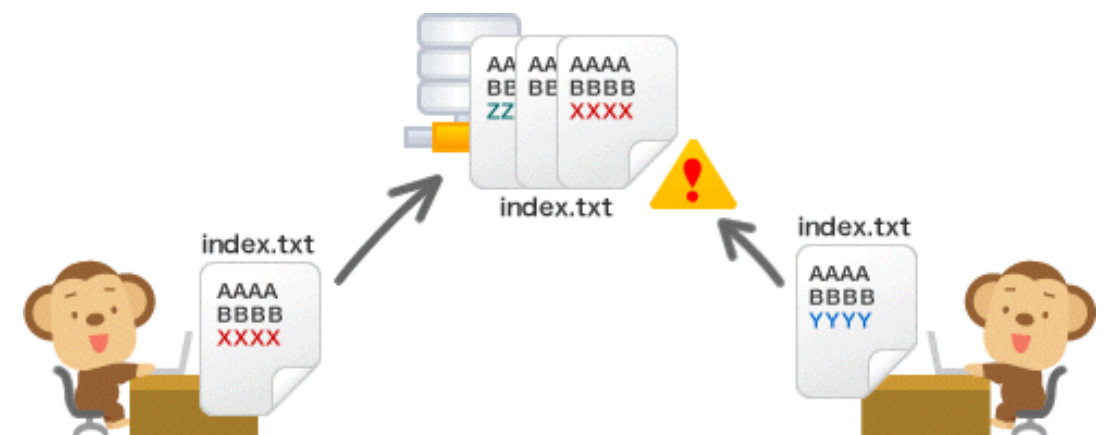
問題：ファイルの競合^{きょうごう}

複数人^{ふくすうにん}で同じファイルを使った場合に古いファイルは消えてしまう

- ☑ 複数人^{ふくすうにん}で同じファイルを使った場合に古いファイルは消えてしまう
- ☑ これを察知^{さっち}することは難しい^{むずか}

Gitを用^{もち}いることで！

- ☑ バージョンの不整合^{ふ せいごう}を察知後、競合^{きょうごう}した箇所^{かしよ}を知ることが可能



Gitの利用例(2)

問題: バグの追跡^{ついせき}

- ☑ バグ^{はっせい}発生時の原因^{げんいん}調査^{ちやうさ}が難しい

Gitを用いることで

- ☑ 発見されたバグが、それがいつ・なぜ書かれたコードなのかを知ることができる
- ☑ 原因がわかることで、対処^{たいしよ}方法を適切^{てきせつ}に考えることができる

Gitの用語

Git用語

- Repository
リポジトリ
- Staging
ステージング
- WorkingTree
ワーキングツリー
- Commit
コミット
- Head
ヘッド
- Branch
ブランチ
- Push
プッシュ
- Pull
プル
- Index
インデックス
- Merge
マージ
- Conflict
コンフリクト

Repository

Repository(リポジトリ)

- ☑ 複数のソースコードを^{かんり}管理する^{たんい}単位
- ☑ リポジトリ内のソースコードは1つのまとまりとして管理される
- ☑ 1つ1つのリポジトリは^{どくりつ}独立した^{りれき}履歴を持つことができる
- ☑ また複数のリポジトリ間で^{どうき}同期を取ることが可能



Staging

Staging(ステージング)

- ☑ コードに変更があった場合に追加される^{ついか}記憶領域^{きおくりょういき}
- ☑ この^{じょうたい}状態になったときに、コミットできる

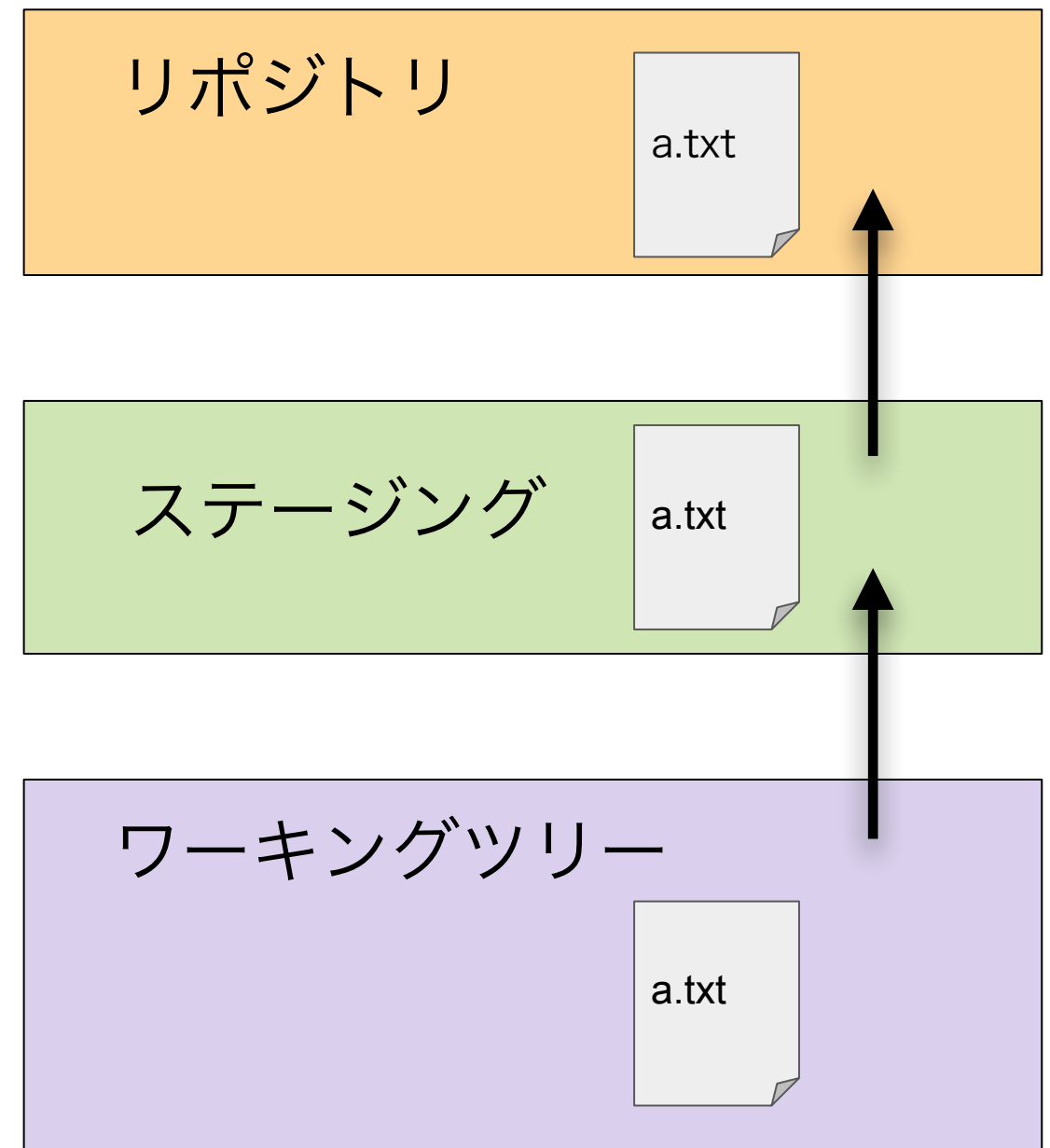
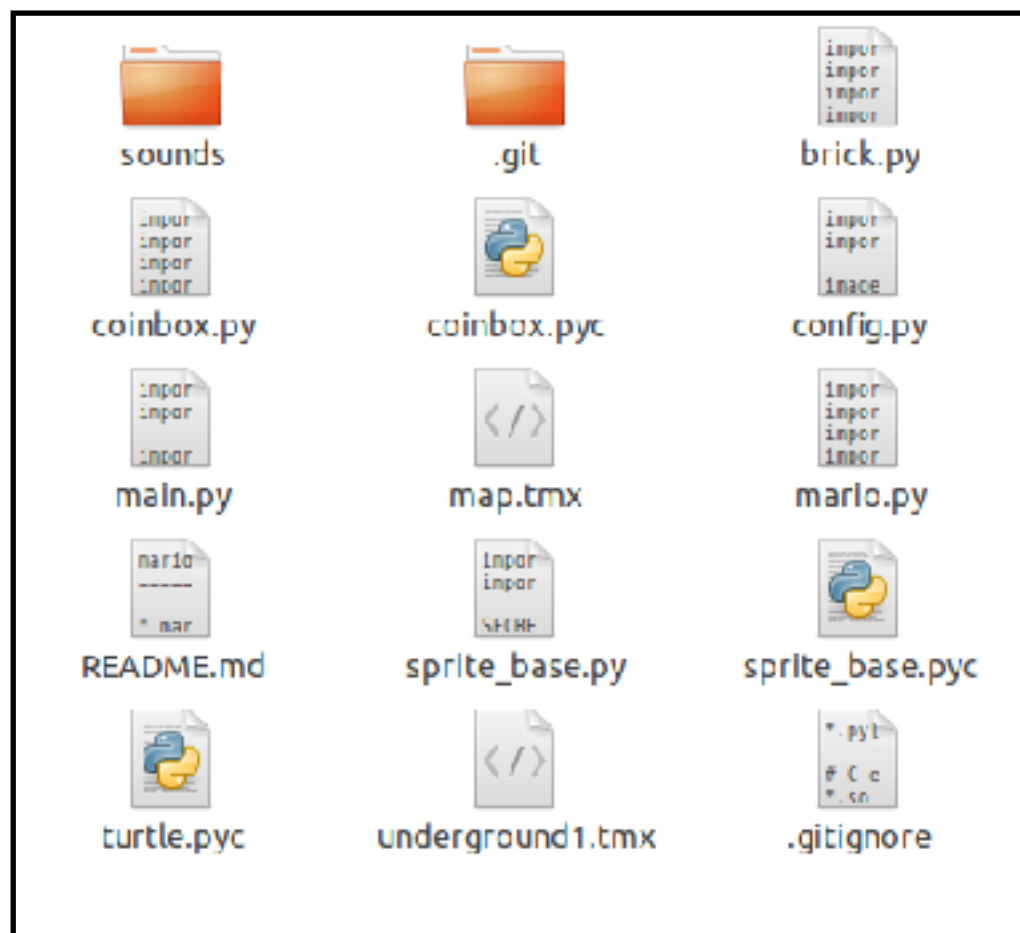
WorkingTree

WorkingTree (ワーキングツリー)

- ☑ ユーザが^{さぎょう}作業しているディレクトリ^{りょういき}領域
- ☑ .git のディレクトリがある。

WorkingTree - Staging - Repository

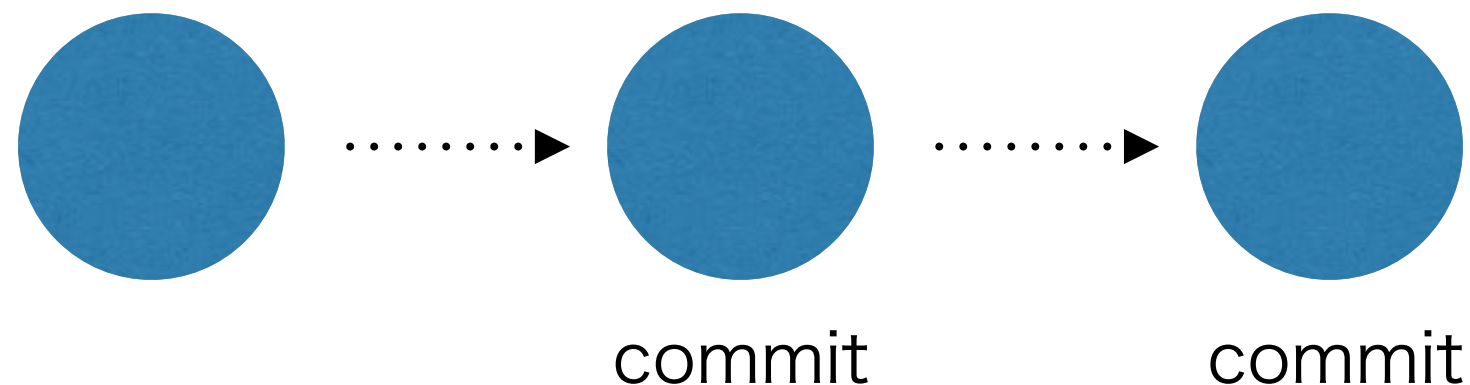
ディレクトリー例



Commit

Commit(コミット)

- ☑ ソースコード^{じょうたい}状態の単位、または新しい履歴を作ること
- ☑ コミットを^つ積み^{かさ}重ねることでソースコードの履歴ができる
- ☑ コミット単位で^ま巻き^{もど}戻すことができる
- ☑ 開発者はコミットに対して^{たい}コメントを^つ付けることができる

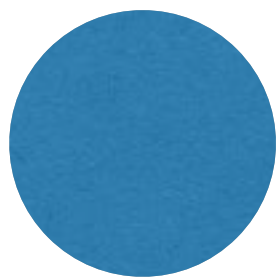


Head

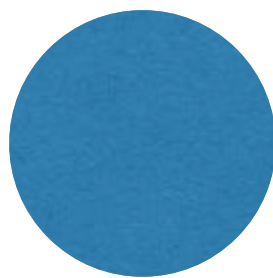
Head(ヘッド)

- ☑ コミット履歴の中で最も最新さいしんのもの
- ☑ HEADを基準きじゅんにして、
1つ前はHEAD~1、2つ前はHEAD~2と、表現ひょうげんする

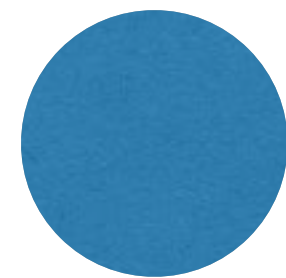
HEAD~2



HEAD~1



HEAD

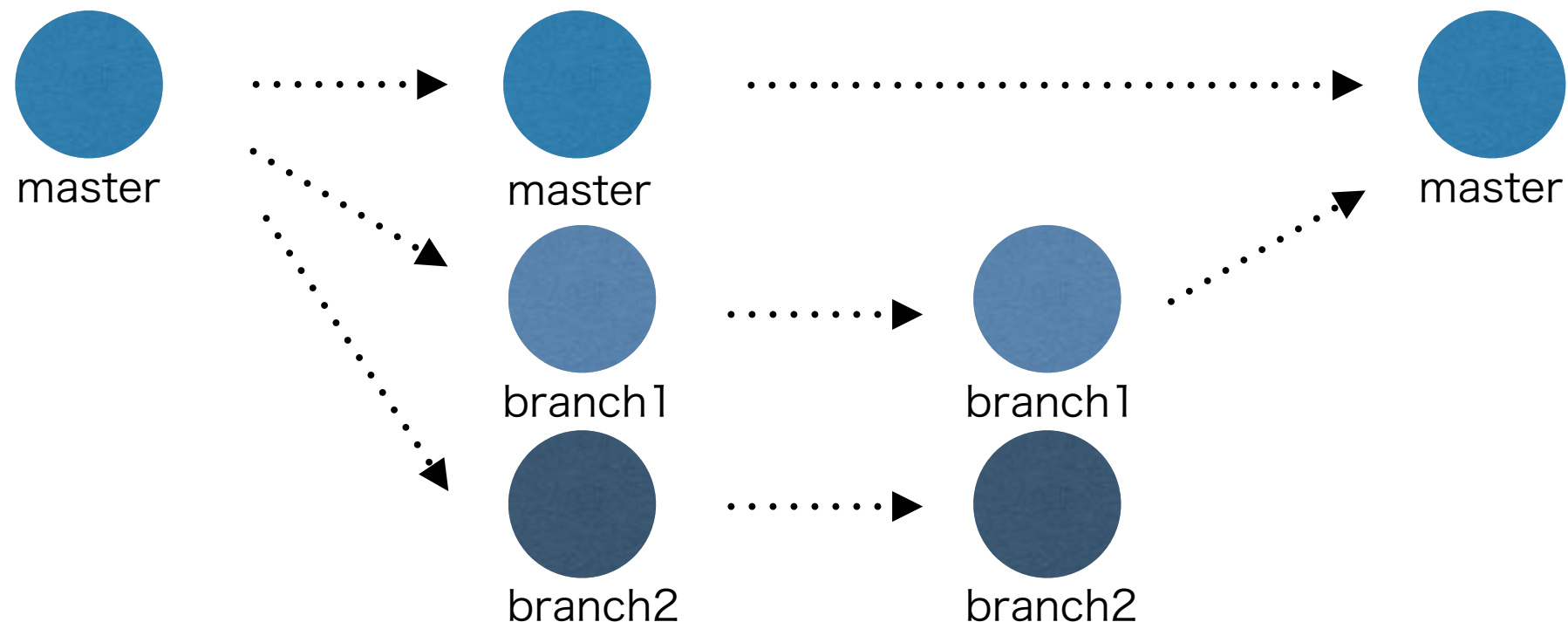


最新

Branch

Branch(ブランチ)

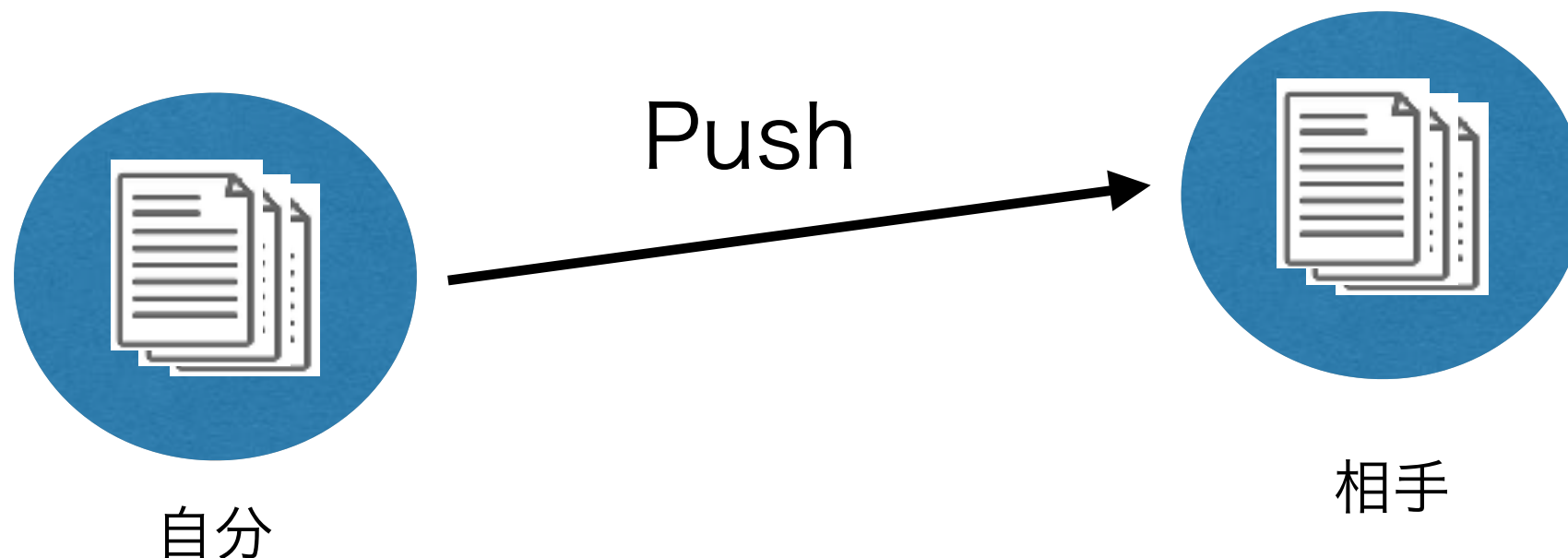
- ☑ 1つのリポジトリ内でのバージョンの^{はせい}派生
- ☑ メインのバージョンをmasterとした上で、
これ^{いがい}以外のバージョンを作り管理することができる
- ☑ ブランチ間の^{ぶんき}分岐・^{けつごう}結合が自由に行える



Push

Push(プッシュ)

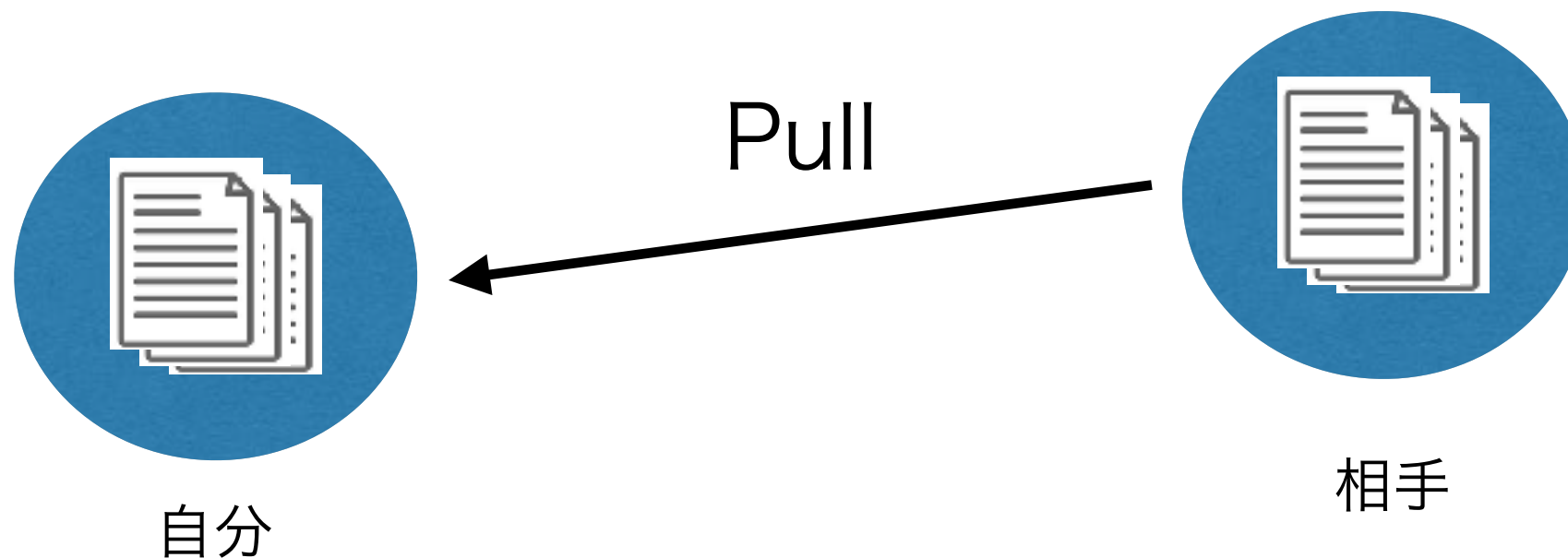
- ☑ 他のリポジトリへの^{とくてい}特定のブランチのソースコードを^{そうしん}送信すること
- ☑ 送信先のリポジトリを指定や^{さくせい}作成することが可能
- ☑ このとき、送信^{さき}先のソースコードと^{きょうごう}競合がある場合はエラーが出る



Pull

Pull (プル)

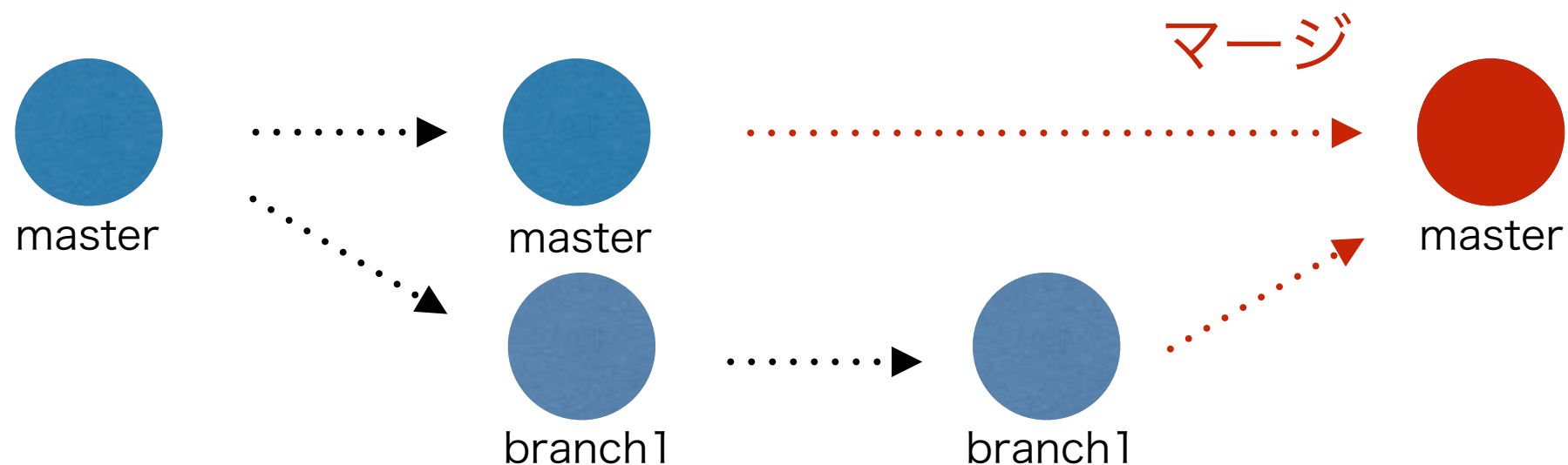
- ☑ 他のリポジトリから、ソースコードを^{しゅとく}取得すること
- ☑ どのブランチへ送信するかは^{してい}指定することができる



Merge

Merge (マージ)

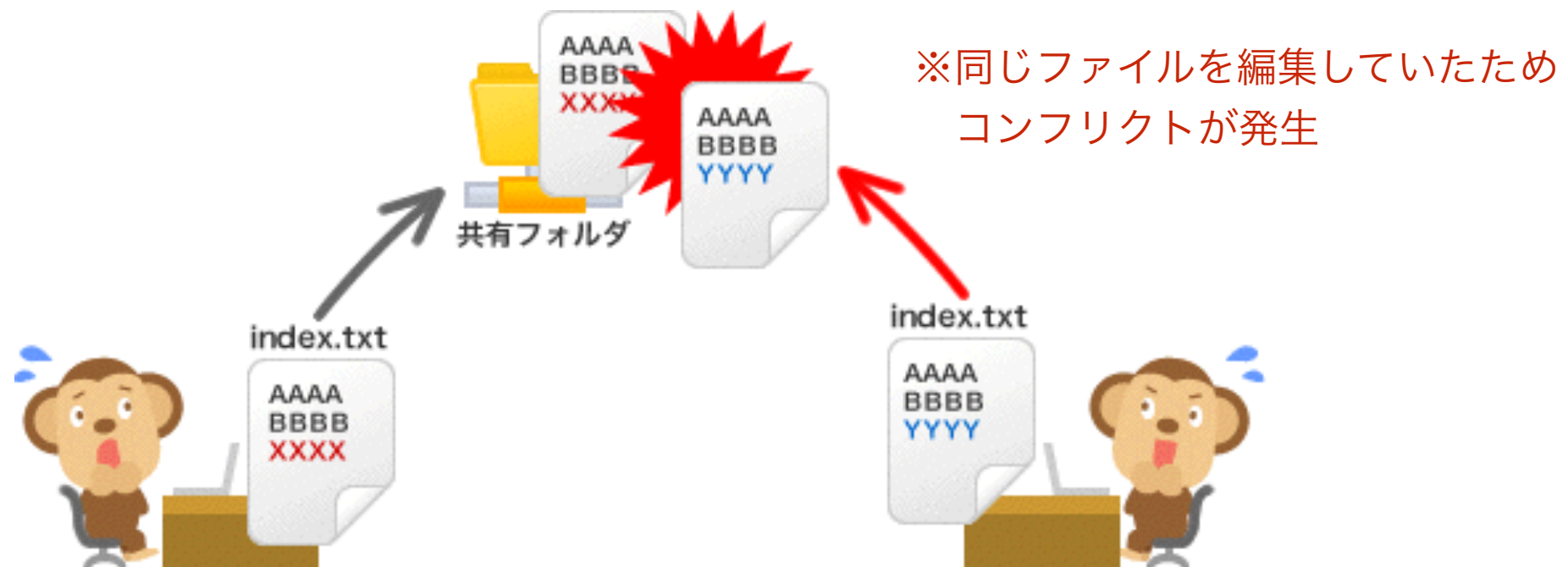
- ☑ 複数のブランチ間で、ソースコードを統合^{とうごう}すること
- ☑ この際^{さい}、競合^{きょうごう}が発生していれば競合（コンフリクト）が起こる。



Conflict

Conflict (コンフリクト)

- ☑ ファイルをマージする際に、ソースコードが競合すること
- ☑ コンフリクトを解決した後で、^{さいど}再度マージを^{じっし}実施する
^{ひつよう}必要がある

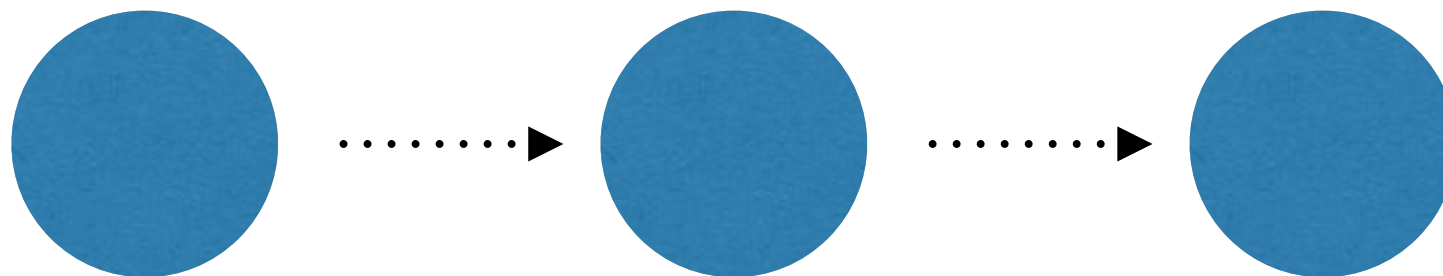


Gitの実例

一人で使う

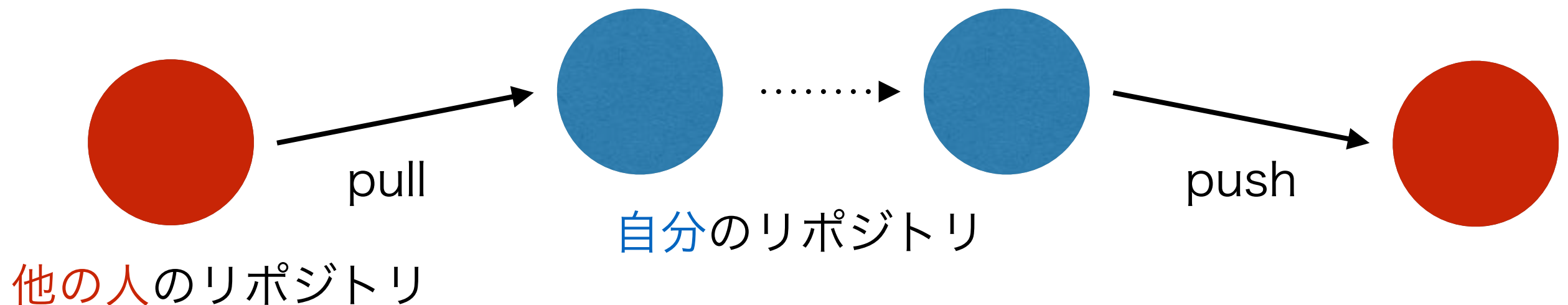
1. Gitを^{しよき}初期化^かを行う
2. ソースコードを書く
3. コミットする
4. 2~3を^く繰り返^{かえ}す

コミット



2、3人で使う

1. 他のリポジトリからソースコードをプルする
2. ソースコードを^{しゅうせい}修正する
3. コミットする
4. 元のリポジトリに対してプッシュする

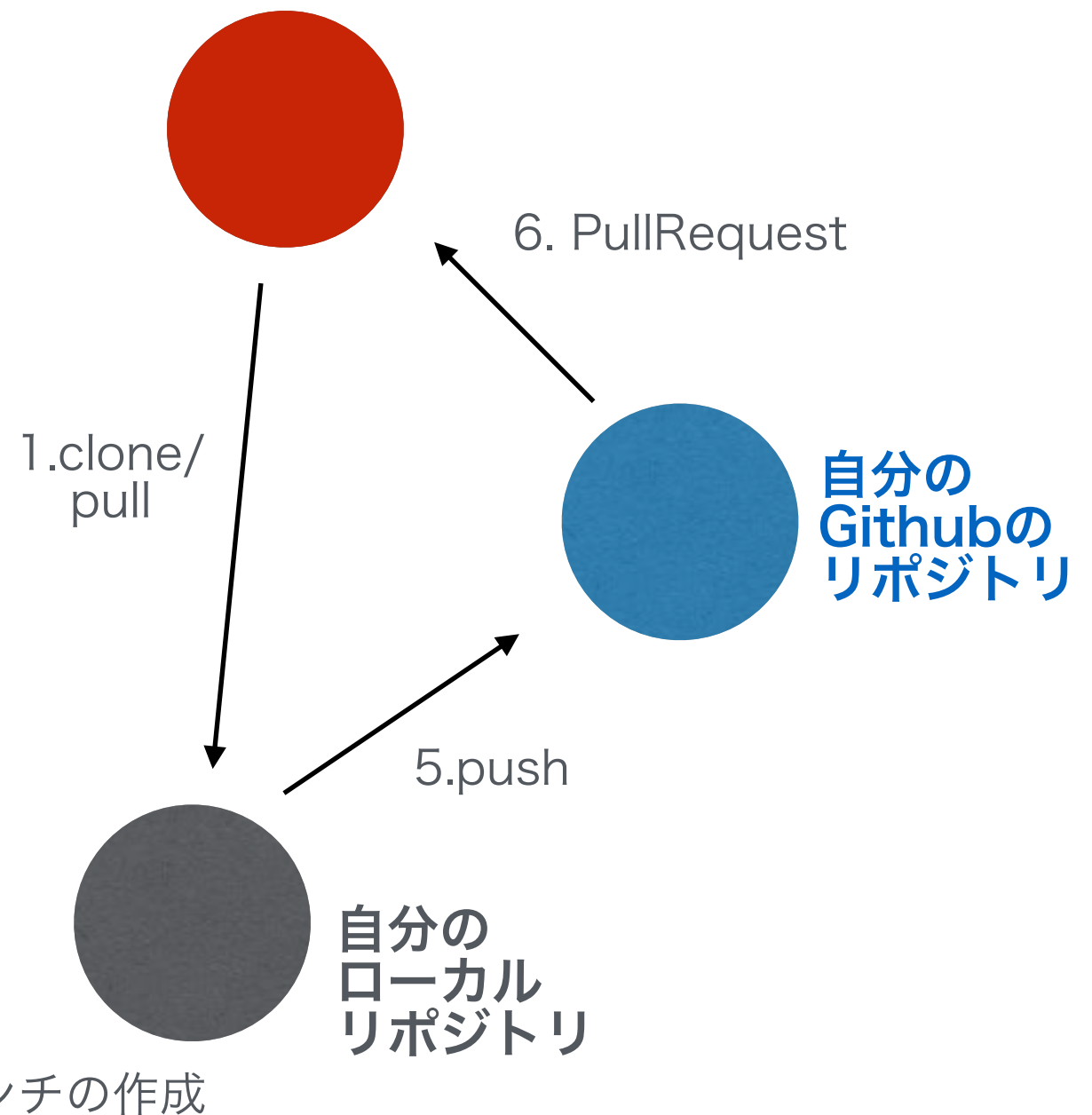


チームで使う

1. 他のリポジトリから
ソースコードをプルする
2. ブランチを作る
3. ファイルを^{へんしゅう}編集する
4. コミットする
5. プッシュする
6. プルリクエストを送る

※^{らいしゅう}来週、GitHubの授業で
^{くわ}詳しく説明します

セントラルリポジトリ



実習準備

実習準備

Cloud9上に新しいPJを^{さくせい}作成する

1. Workspace name

exec-git1

2. Team

hedspistudent

3. Choose a template

HTML5

The screenshot shows the Cloud9 workspace creation form. The 'Workspace name' field is highlighted with a red box and contains the text 'exec-git1'. The 'Team' dropdown menu is also highlighted with a red box and shows 'hedspistudent' selected. The 'Choose a template' section is highlighted with a red box and shows a grid of templates including HTML5, Node.js, PHP, Python, Django, Ruby, C++, Wordpress, Rails Tutorial, Blazor, and Harvard's C... The 'Create workspace' button is at the bottom left of the template grid.

Workspace name:

Description:

Team:

Hosted workspace | Clone workspace | Remote SSH workspace | Salesforce

Private: This is a workspace for your eyes only

Public: This will create a workspace for everybody to see

Clone from Git or Mercurial URL (optional)

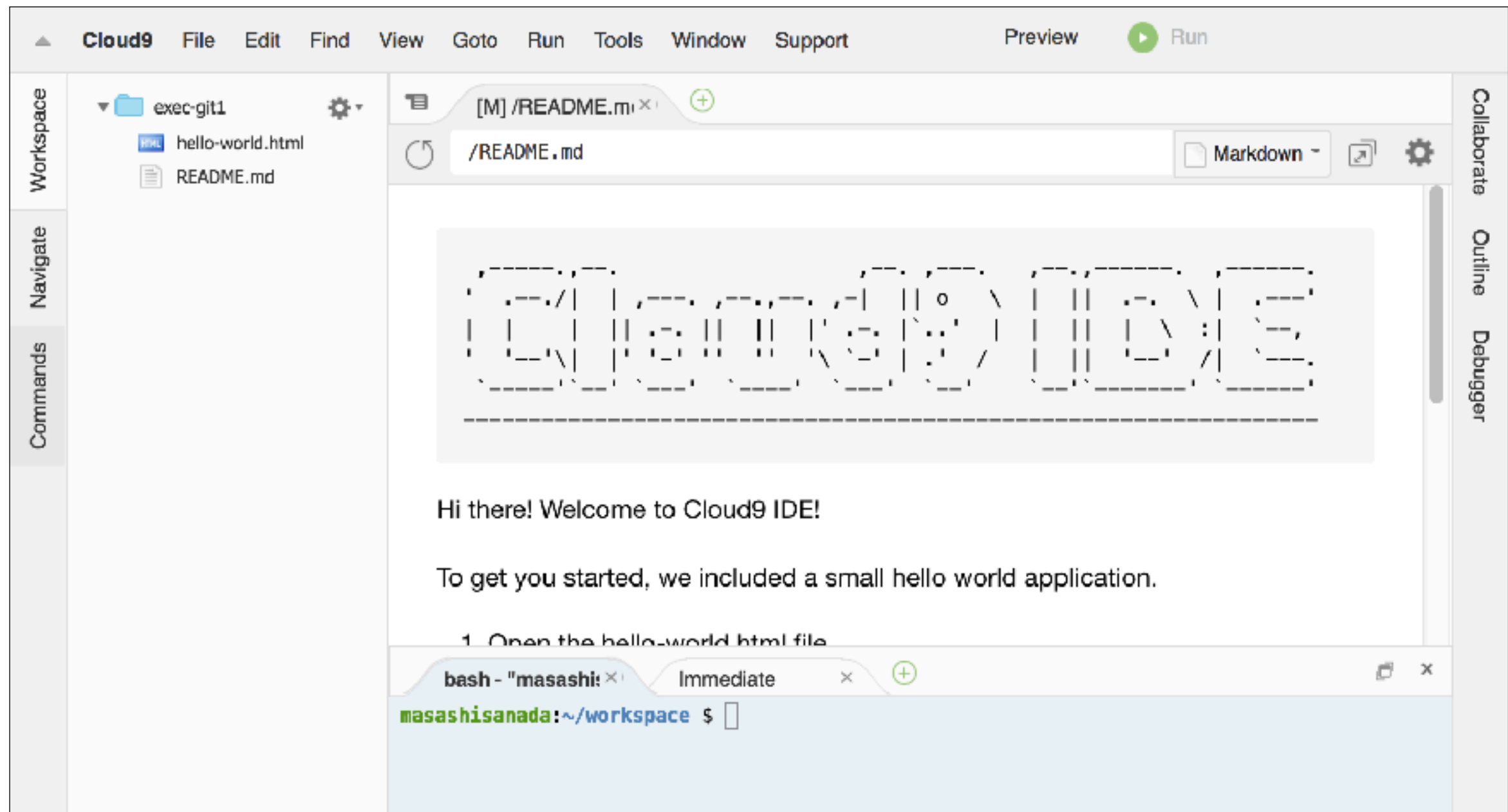
Choose a template

HTML5 | Node.js | PHP, Apache... | Python | Django | Ruby

C++ | Wordpress | Rails Tutorial | Blazor | Harvard's C...

Create workspace

実習準備



課題(1)

課題(1)

Gitの最初の使い方について勉強します。
リポジトリ作成から最初のコミットまでを実施します。

■^{てじゅん}手順

1. リポジトリを作成する
2. ワーキングツリーの^{じょうたい}状態を確認する
3. コミットするファイルを決める
4. ファイルをコミットする
5. コミットしたファイルの^{りれき}履歴を確認する

■コマンド

git init

git status

git add

git commit

git log

課題(2)

課題(2)

^{さぎょう}作業ブランチの作成を行います。

作業ブランチでファイルを編集しコミットします。

■手順

1. ブランチの^{じょうきょう}状況を確認する
2. 新しいブランチを作成する
(以降^{いこう}"作業ブランチ"と呼ぶ^よ)
3. 作業ブランチに^{いどう}移動する
4. 作業ブランチでファイルを編集する
5. 作業ファイルをコミットする
(add→commit)

■コマンド

git branch

git branch
[作業ブランチ名]

git checkout

vi README.md

git add

git commit

課題(3)

課題(3)

masterブランチへファイルのマージを実施します。
マージ後、作業ブランチを削除します

■手順

1. masterブランチに移動する
2. ファイルをマージする
3. 作業ブランチを削除する

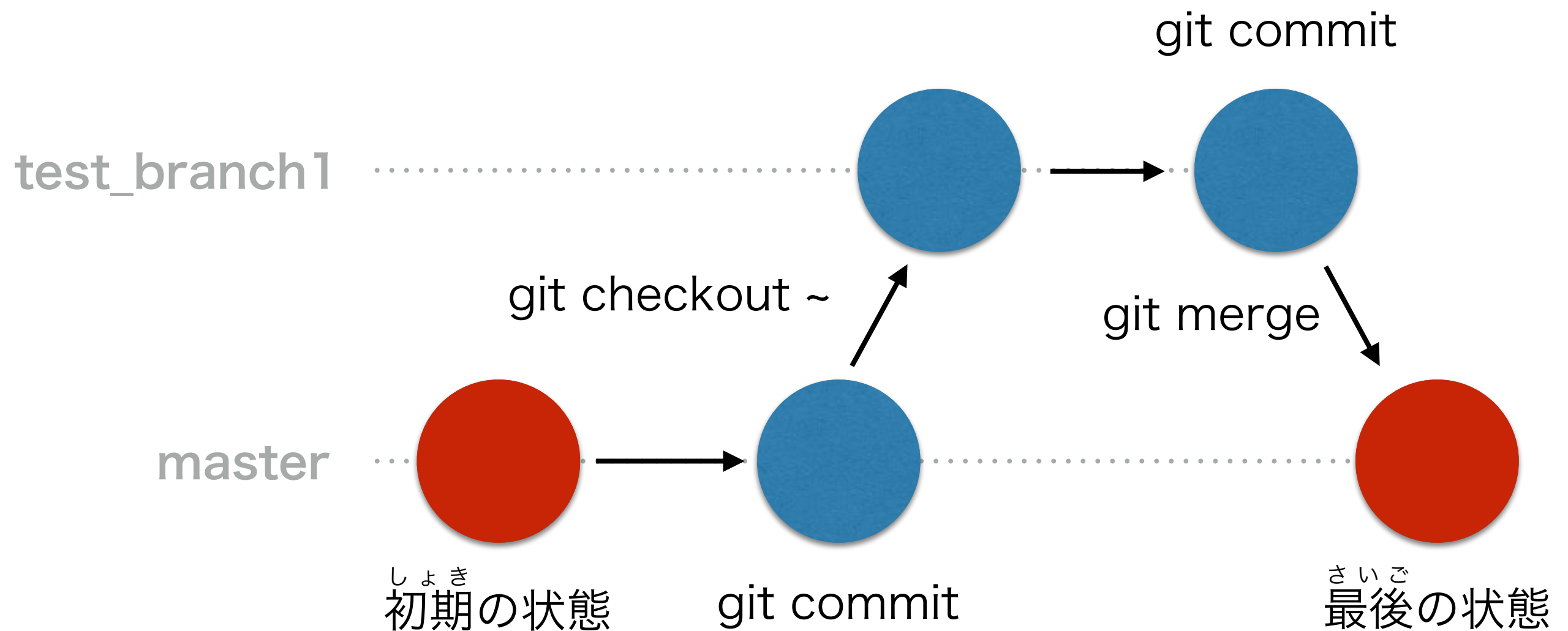
■コマンド

git checkout
[作業ブランチ名]

git merge

git branch -D

課題(1)~(3)のまとめ



The background of the slide features a scenic view of Mount Fuji under a clear blue sky with scattered white clouds. In the foreground, there are lush pink cherry blossom trees. On the right side, a portion of a traditional Japanese temple with a dark, multi-tiered roof is visible.

今日の授業は終わりです
来週も元気に会いましょう！

Hẹn gặp lại vào tuần tới