

Git講習

2014.04.15

こんばんは

講師 紹介

- masawada
- 情報理工学部 情報・通信工学科 3年
コンピュータサイエンスコース
- わからないことがあれば
tetra1984@gmail.comまたは
@masawada (Twitter) まで



本日のおしながき

- Gitとは
- 基本的なコマンド / 考え方
- Gitの中身を見てみよう

講習を行うにあたって

- 実際にコマンドを打ちながら説明します。
- 可能な方はお手元の端末でお試してください。
- 資料はWebサイトにアップロードされています。

Gitとは

Gitとは

バージョン管理システム (VCS)

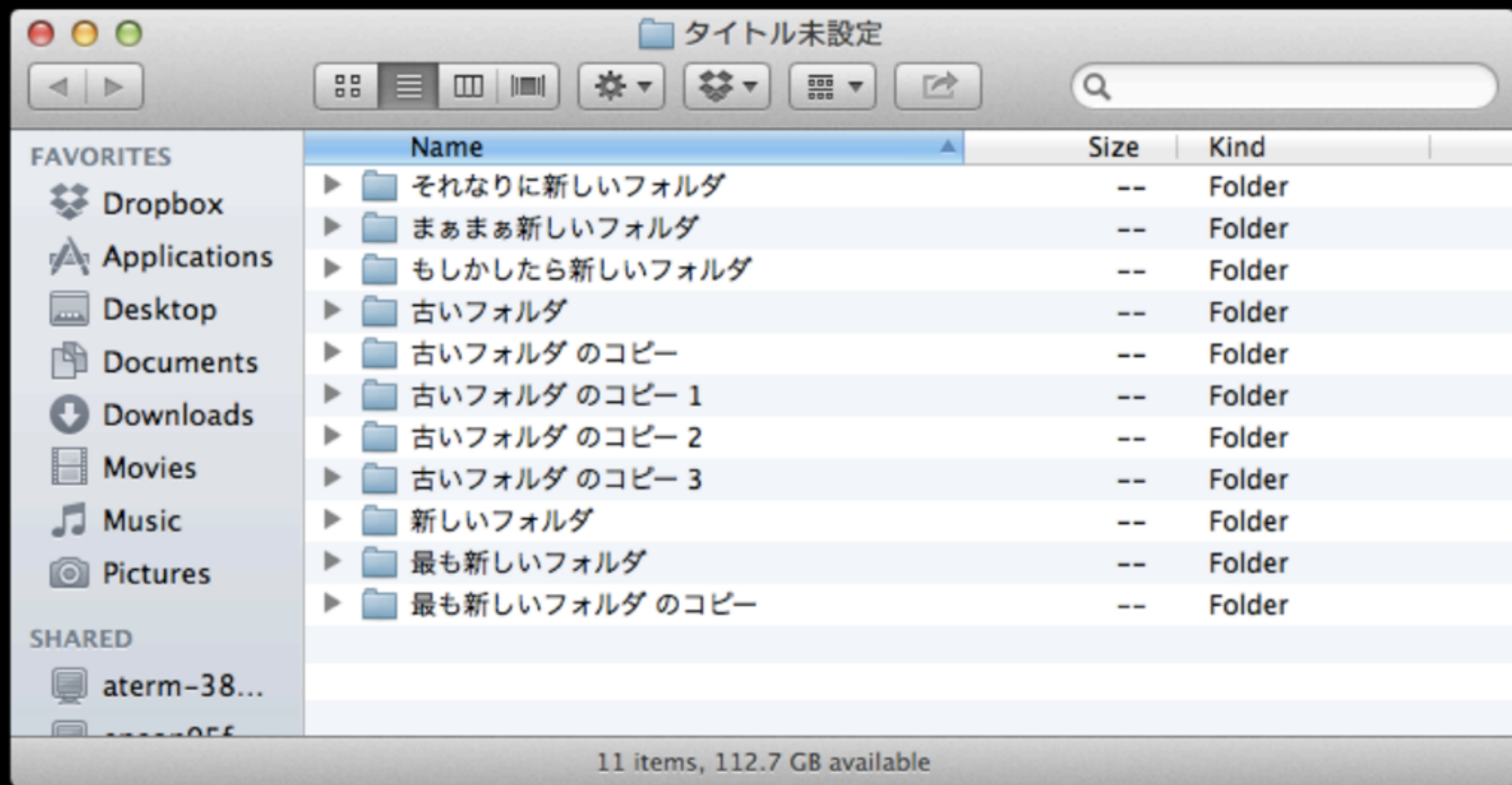
= 変更履歴を記録 / 管理するソフトウェア

他にもSubversionやBazaarなど...

例えば

- レポートやプログラムを書いている区切りの良いところで保存しておきたい
- まともに書けている / 動いている状態のものを保存しておいて、問題が発生したときに戻せるようにしたい

いままでの 変更履歴管理



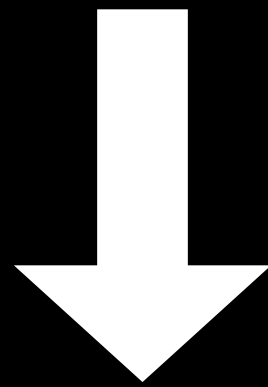
ファイルベース

バージョン管理

識別No.		説明	ファイル名	追加日時
N00001	v0.1	初期版	-	2014/1/5
N00002	v0.1.1	app.jsにコメントを追加	app.js	2014/1/6
N00003	v0.1.1	exec.jsのエスケープミスを修正	exec.js	2014/1/6
N00004	v0.1.1	view.htmlを追加	view.html	2014/1/6

Excelベース

手動で変更履歴を管理するので
人為的なミスが多くなる



これらをすべて
自動化したい

という考えのもと生まれたのが
バージョン管理システム

- ・ 一年次後期の基礎プログラミングおよび演習
 - C言語の書き方を学ぶ
 - 変更履歴を管理する必要があまりない
- ・ 本ワークショップ
 - 動くモノを作る
 - 変更履歴を管理する必要がある

基礎編

基礎編

- リポジトリ (repository)
- コミット (commit)

リポジトリとは

データの変更履歴を管理する
帳簿のようなもの

コミットとは

リポジトリに対して
変更履歴を追加すること

コミット作成の流れ

1. ワークツリーでファイルを追加/変更/削除する
2. ある程度まとまったらインデックスに追加する
インデックスとは、機能ごとなどに分けてコミットを作るために変更したファイルを溜める場所
インデックスに登録されていないファイルはコミットされない
3. インデックスの内容をリポジトリにコミットする

リポジトリとコミット



ワークツリー
(作業ディレクトリ)

インデックス

リポジトリ

リポジトリの初期化

```
$ git init
```

インデックスに変更を登録

```
$ git add [filename]
```

```
$ git add .
```

コミットの作成

```
$ git commit
```

```
$ git commit -m “message”
```

ファイルの状態を表示

```
$ git status
```


変更履歴の表示

```
$ git log
```

ここまでをデモ

差分の表示

```
$ git diff [old] [new]
```

ここまでをデモ

ブランチ編

ブランチとは

枝

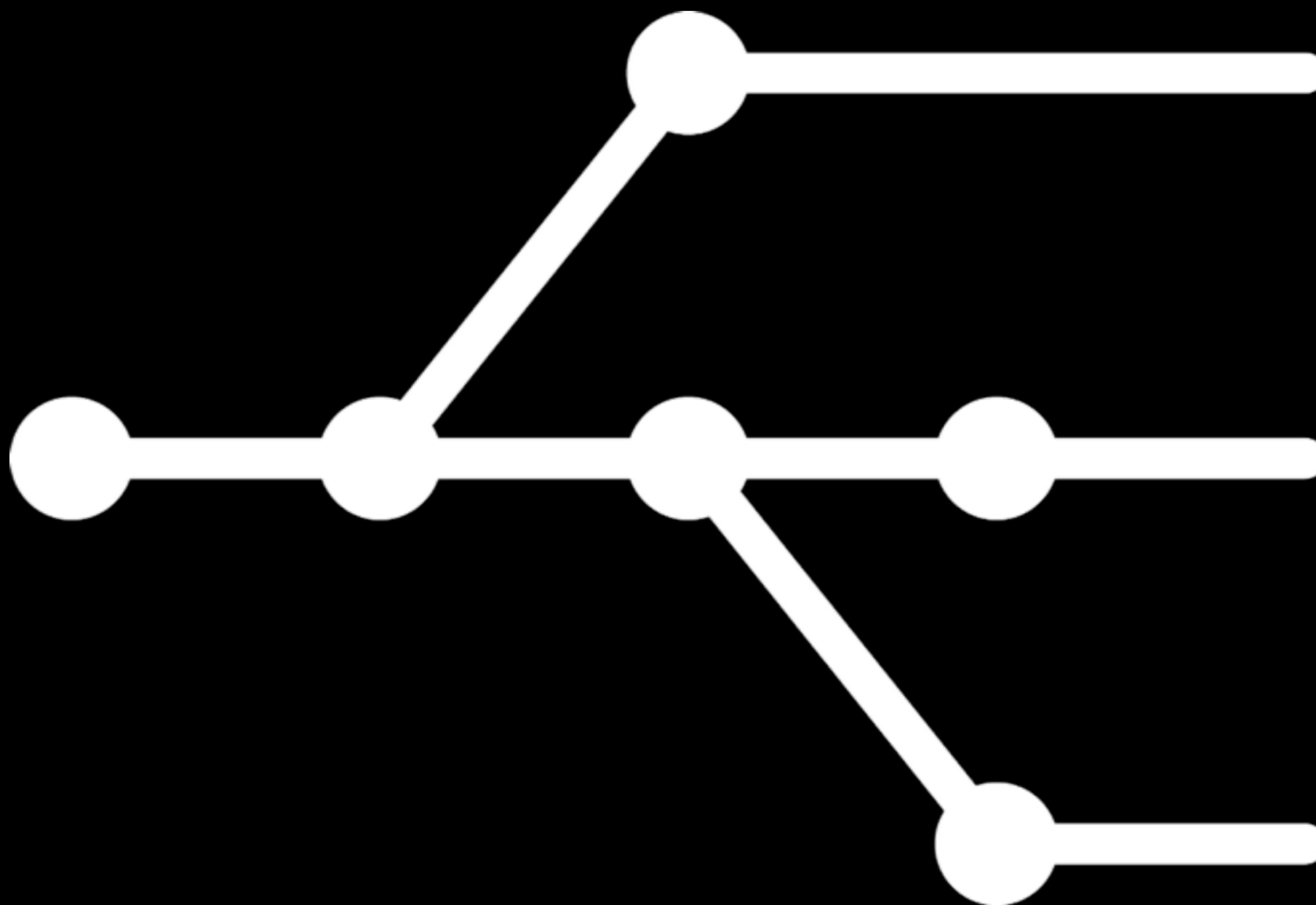
ブランチとは

支部, 支局

ブランチとは

- 複数の機能を同時に並行して作成できる
- チーム開発をするときにブランチを作ることで他人の変更に迷惑をかけないように作業できる
- 不安な作業はブランチを作ってから行うことでブランチを消すだけで元の状態に戻れる

ブランチとは



ブランチの作成

```
$ git branch [name]
```

ブランチの乗り換え

```
$ git checkout [name]
```

ブランチの作成と乗り換え

```
$ git checkout -b [name]
```

ブランチの統合

```
$ git merge [name]
```

ブランチの削除

```
$ git branch -d [name]
```

```
$ git branch -D [name]
```

mergeされていないブランチは-Dで消す

ここまでをデモ

ブランチの競合

ブランチの競合

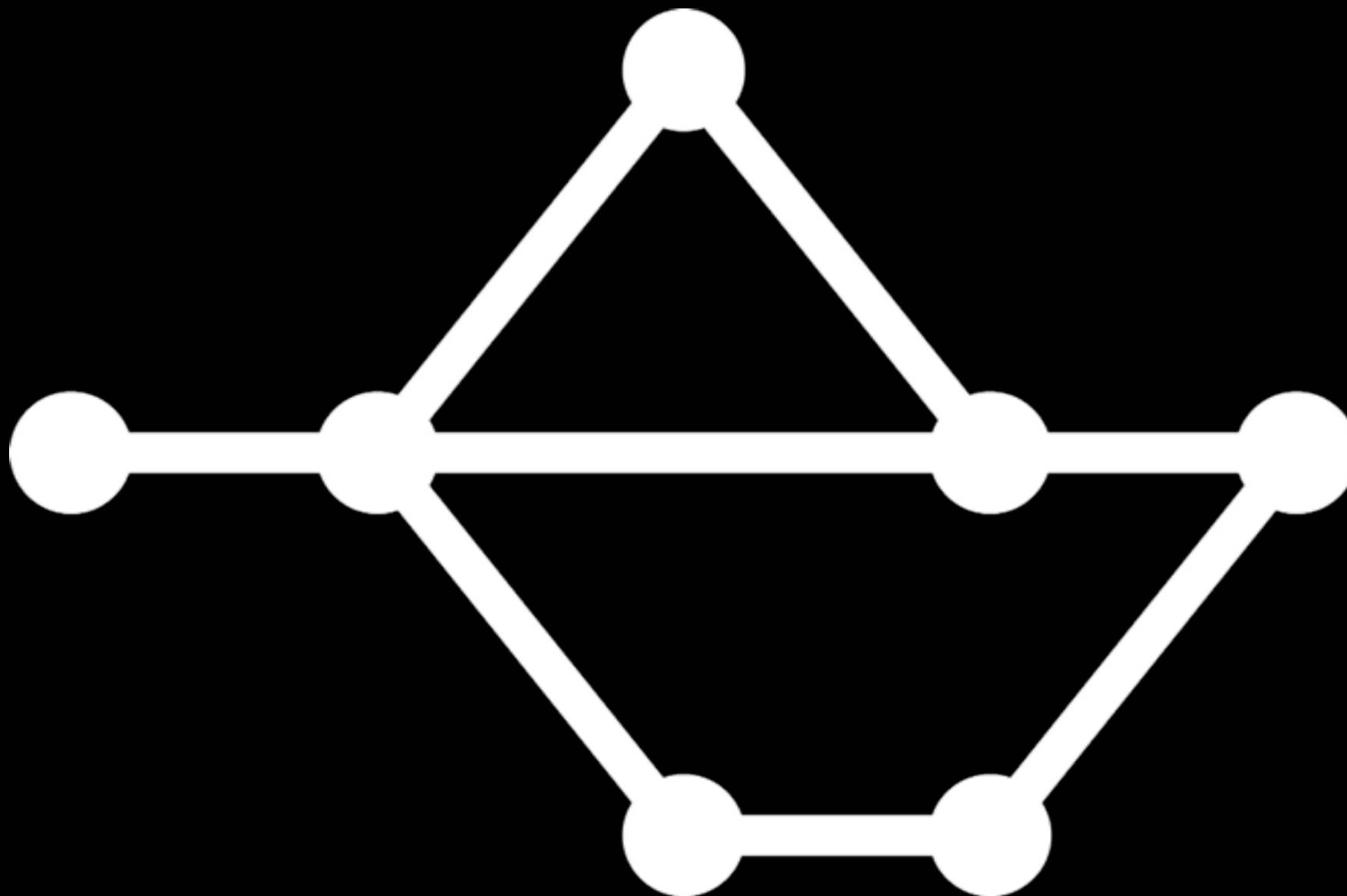
1. masterからfeature_1, feature_2ブランチを作成
2. feature_1ブランチに機能を追加、コミットする
3. feature_1ブランチをmasterブランチにmerge
4. feature_2ブランチに機能を追加、コミットする
5. feature_2ブランチをmasterブランチにmerge

ブランチの競合

feature_1

master

feature_2



実際にデモ

リモートリポジトリ

リモートリポジトリ

- リポジトリをサーバにバックアップ
- リポジトリを他人と共有できる

Gitリポジトリ ホスティングサービス

The GitLab logo, featuring the word "GITLAB" in a bold, sans-serif font, flanked by two sets of three horizontal lines on each side.

The Atlassian Bitbucket logo, featuring the word "Atlassian" in a small, blue, sans-serif font above the word "Bitbucket" in a large, bold, blue, sans-serif font. To the left of "Bitbucket" is a blue icon of a bucket with a white circle in the center.

The GitHub logo, featuring the word "github" in a bold, lowercase, sans-serif font, with the words "SOCIAL CODING" in a smaller, uppercase, sans-serif font below it.



リモートリポジトリの登録

```
$ git remote add [name] [url]
```

リモートリポジトリの取得

```
$ git clone [url]
```


リモートリポジトリの更新

```
$ git push
```

```
$ git push -u [name] [branch]
```

ローカルリポジトリの更新

```
$ git pull
```

ここまでをデモ

コミットの
取り消し/やり直し

直前のコミットのやりなおし

```
$ git commit --amend
```

ひとつまえのコミットをやりなおす

インデックス追加の取り消し

```
$ git reset
```

インデックスに追加したファイルのリストを
インデックスから取り除く

コミットの削除

```
$ git reset --hard [commit]
```

指定したコミットまで戻る

指定コミットから現在までのコミットは残らない

コミットの削除

```
$ git reset --soft [commit]
```

指定したコミットまで戻る
ワークツリーの変更履歴は残る

コミットの打ち消し

```
$ git revert [commit]
```

指定したコミットを打ち消すコミットを作成する
そのコミットで追加した部分をごっそり削除する

ここまでをデモ

その他のTips

.gitignore

.DS_Storeや.swpなどのいらないファイルを
追加しないようにできる

ここまでをデモ

最後に

まだまだ機能は
たくさんある



Pocket「著作権保護コンテンツ」
Reference



Git

ポケットリファレンス

岡本 隆史
武田 健太郎 著
相良 幸範

人気バージョン管理システムGitの
使い方とトラブルシューティングが
この1冊でわかる!

- 重要コマンドとオプションをリファレンス形式で網羅
- コマンドごとのエラーメッセージと対処法も解説
- GitとGitHub活用法の全体像を示すチートシート、
共有リポジトリ構築・運用などの活用ノウハウも紹介

技術評論社

「著作権保護コンテンツ」

ありがとうございました