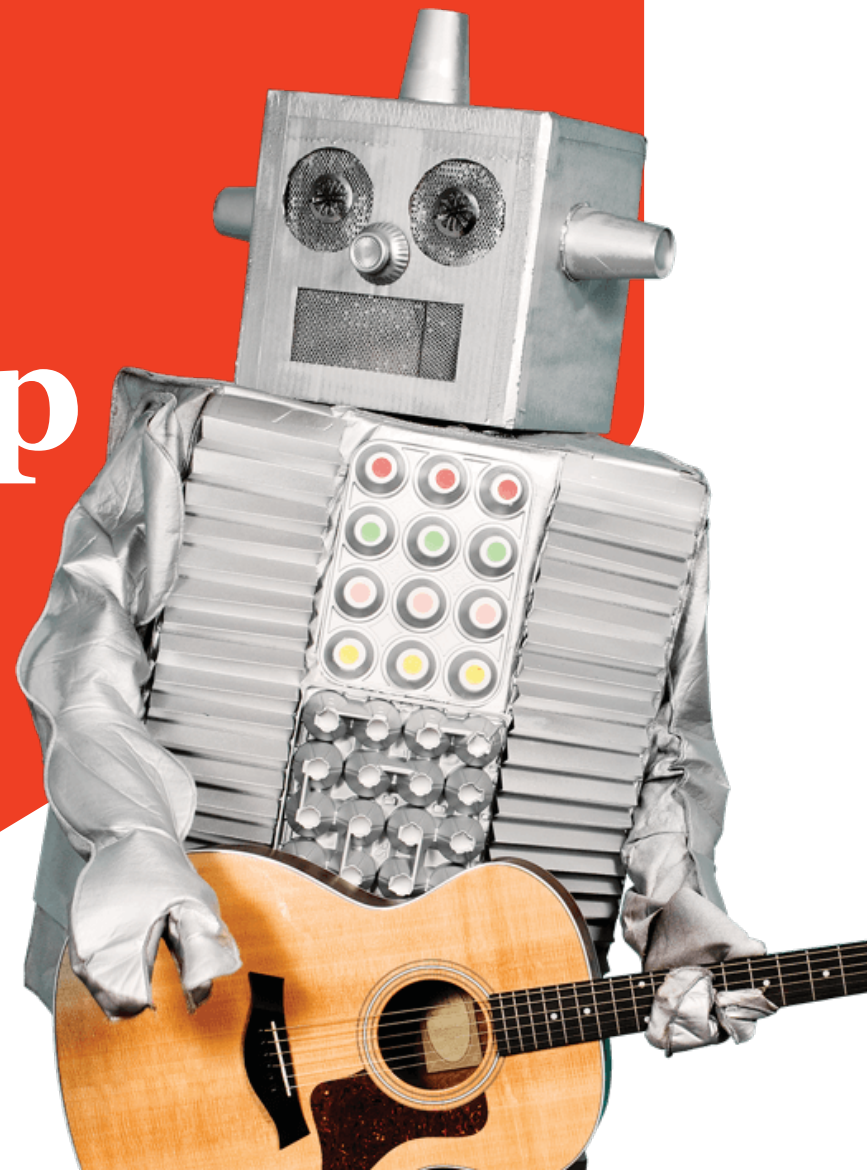
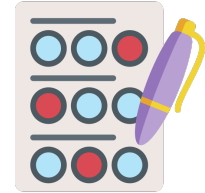


ハンズオンBの確認 Hands-on B's Recap



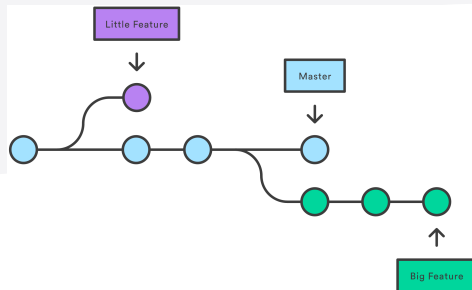


クイズの時間!



「**git checkout -b develop**」は何しますか？

新しいブランチ (develop) を作成して、そのブランチに移動するコマンド。



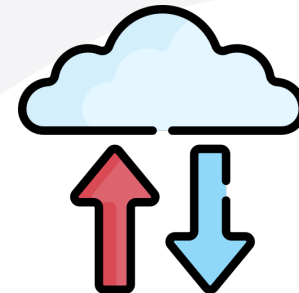
ブランチを使う理由は？

コードの開発、テスト、公開バージョンなどを分離するため。



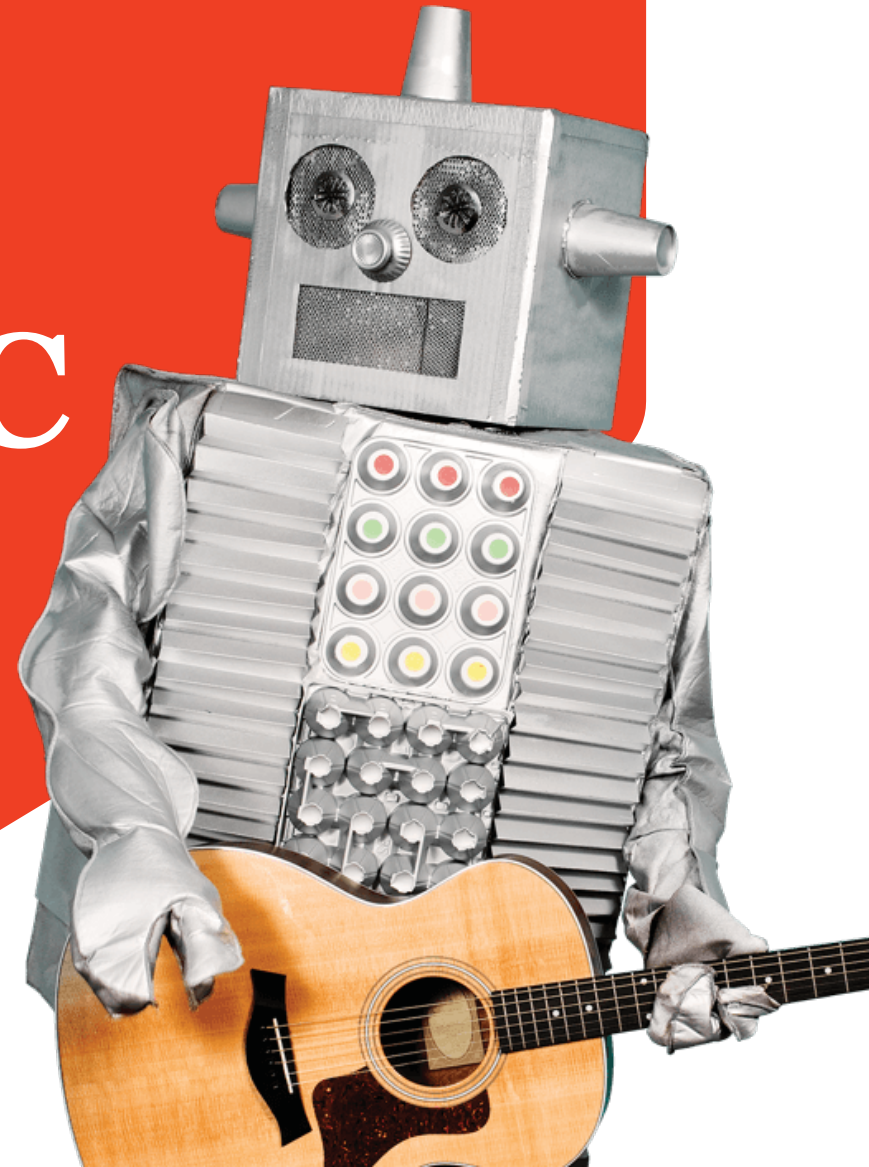
Pull Request
プルリクエスト
とは？

変更を取得するためにターゲットリポジトリをリクエストすることです。



ハンズオンCの準備

Prep for Hands-on C



準備 - local git

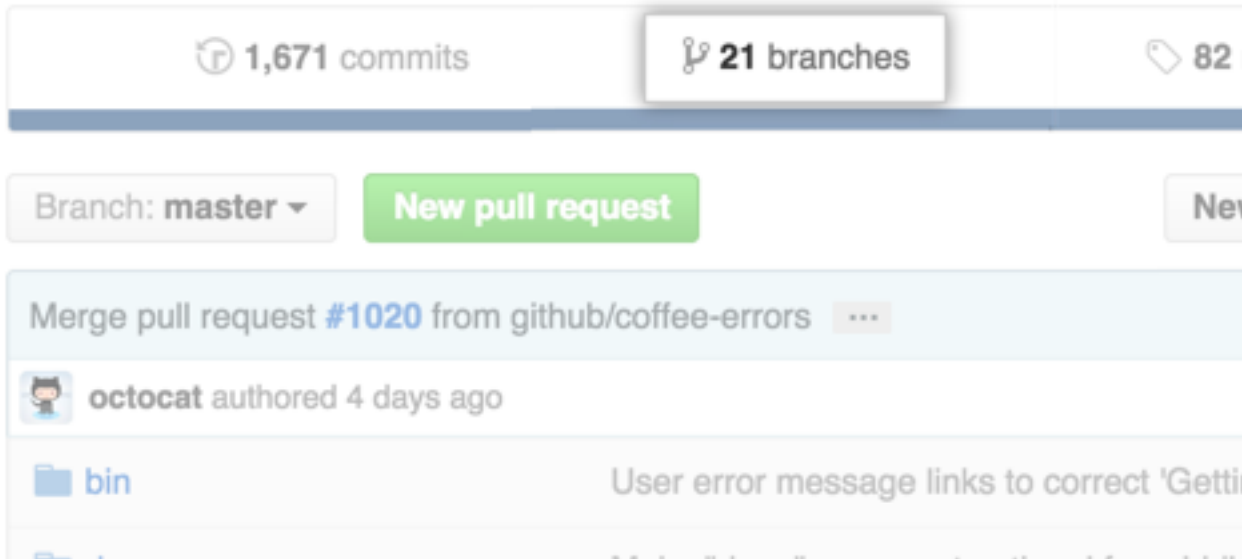
```
$ cd Documents/Kintone_Dojo
$ git status
$ git branch

$ git branch --delete develop

$rm develop_file.md
```

- 昨日、「develop」ブランチを作成、マージ、削除しました。
- 「master」ブランチだけにしましょう。
- 「develop」ブランチがまだある場合。削除して。
- develop_file.md ファイルも削除して。

準備 - GitHub



- Github.comのKintone_Dojoリポジトリに移動します。
- github.com/**USR**/Kintone_Dojo
- ブランチが1つだけかどうかを確認する。
- developブランチがあればブランチを削除してください。

Your branches

develop Updated 20 hours ago by ahandsel

3 | 0

#2 Merged



準備 - タイムトラベル

```
$ git checkout master
$ git checkout -b timeline

$ touch yr_1
$ git add yr_1
$ git commit -m "Year 1"

$ touch yr_2
$ git add yr_2
$ git commit -m "Year 2"
```

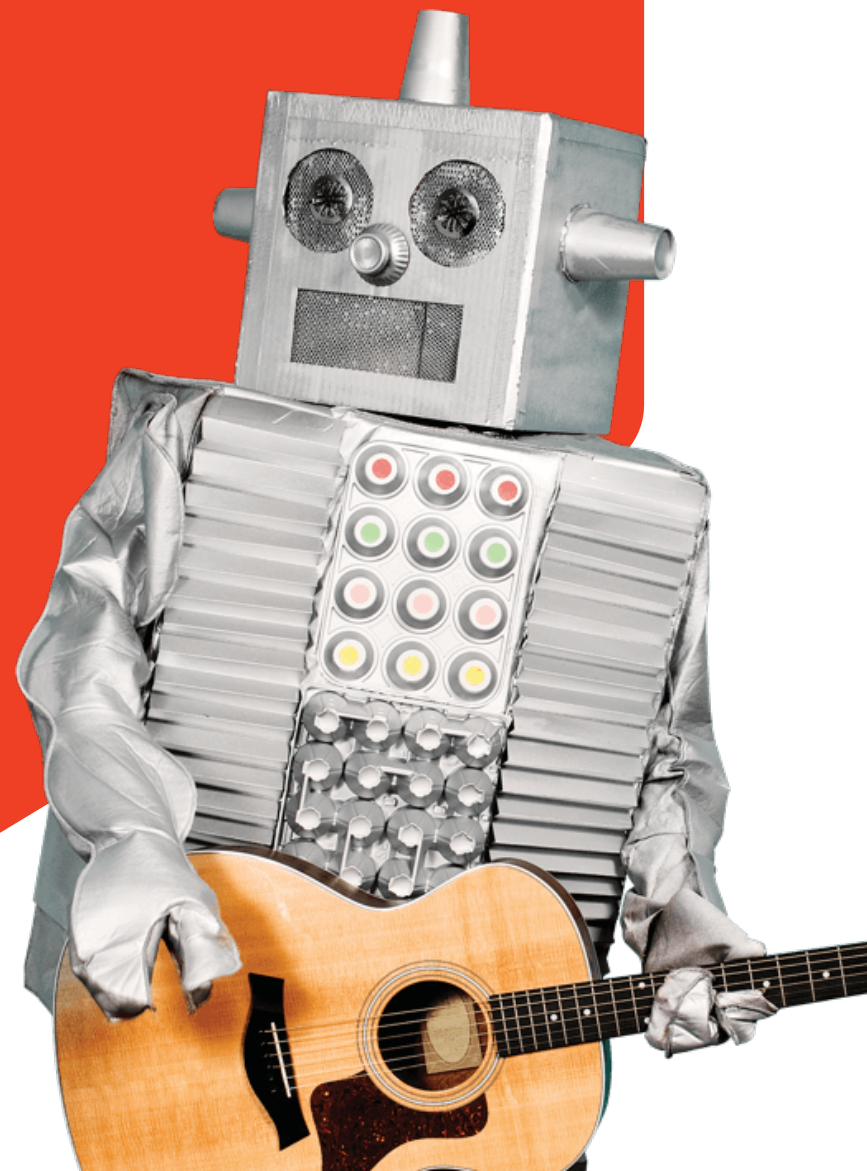
「**timeline**」ブランチを作成します。

遊ぶためにファイルを作成する。

次のファイルを作成して別々にコミットします。

- yr_1 yr_2 yr_3

元に戻す：
時計を巻き戻す!
ハンズオンC



歴史を見る

```
$ ls
  README.md yr_1 yr_2 yr_3
$ git log --oneline
489ca71 (HEAD -> timeline) Year 3
7470790 Year 2
444b696 Year 1
8766df2 (origin/master, master) ...
$ git push origin timeline
```

git log --oneline

ブランチ上でのコミットを一覧表示します。

チェックアウトおよび復帰コマンドに提供されたコミットハッシュを使用する。

git push origin timeline

これで、**timeline**ブランチはGitHub.comにプッシュされています。ブラウザでリポジトリにアクセスして確認します。

GitHub.com - timeline branch

Your recently pushed branches:

 **timeline** (17 minutes ago)

 [Compare & pull request](#)

Branch: **timeline** ▼

[New pull request](#)

[Create new file](#)

[Upload files](#)

[Find file](#)

[Clone or download](#) ▼

This branch is 3 commits ahead, 5 commits behind master.

 [Pull request](#)  [Compare](#)

 **ahandsel** Year 3

Latest commit f7cf1cb 24 minutes ago

 [README.md](#)

README file created

yesterday

 [yr_1](#)

Year 1

27 minutes ago

 [yr_2](#)

Year 2

24 minutes ago

 [yr_3](#)

Year 3

24 minutes ago

GitHub.com - timeline's commits

Branch: timeline ▾

Commits on Jun 10, 2020

Year 3

 ahandsel committed 25 minutes ago



f7cf1cb



Year 2

 ahandsel committed 26 minutes ago



f7fb07c



Year 1

 ahandsel committed 28 minutes ago



e4df7f2



Commits on Jun 9, 2020

README file created

 ahandsel committed yesterday



03098e7



過去を訪ねる

```
$ $ git log --oneline
489ca71 (HEAD -> timeline,
origin/timeline) Year 3
7470790 Year 2
444b696 Year 1
8766df2 (origin/master, master) ...

$ git checkout 444b696

$ ls
README.md  yr_1
```

git log --oneline

- ブランチ上でのコミットを一覧表示します
- チェックアウトおよび復帰コマンドに提供されたコミットハッシュを使用する

git checkout [commit hash]

- 作業ディレクトリを[commit]とまったく同じ状態に変換します。
- これが元に戻すコミットかどうかを確認します。
- この状況で行われた変更は保存されません。

1 Commit文過去に戻る

```
$ git checkout timeline
Switched to branch 'timeline'

$ ls

README.md  yr_1  yr_2  yr_3

$ git revert HEAD
Removing yr_3.md

$ ls

README.md  yr_1  yr_2
```

git revert HEAD

- 1コミット前に戻ります。

コミットハッシュで過去に戻る

```
$ git log --oneline
```

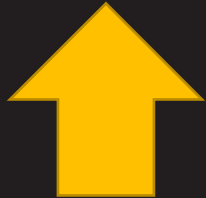
```
727642d (HEAD -> timeline) Revert "Year 3"
```

```
f7cf1cb (origin/timeline) Year 3
```

```
f7fb07c Year 2
```

```
e4df7f2 Year 1
```

```
03098e7 (origin/master, master) README file created
```



```
$ git push origin timeline
```













```
...
```

```
Your branch is up to date with 'origin/timeline'.
```




コミットハッシュで過去に戻る

Branch: timeline ▾

Commits on Jun 10, 2020

Revert "Year 3" ...		727642d	
 ahandsel committed 18 minutes ago			
Year 3		f7cf1cb	
 ahandsel committed 1 hour ago			
Year 2		f7fb07c	
 ahandsel committed 1 hour ago			
Year 1		e4df7f2	
 ahandsel committed 1 hour ago			

Commits on Jun 9, 2020

README file created		03098e7	
 ahandsel committed yesterday			

コミットハッシュで過去に戻る

```
$ git log --oneline
```

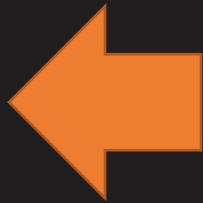
```
727642d (HEAD -> timeline, origin/timeline) Revert "Year 3"
```

```
f7cf1cb Year 3
```

```
f7fb07c Year 2
```

```
e4df7f2 Year 1
```

```
03098e7 (origin/master, master) README file created
```



コミットハッシュで過去に戻る

```
$ git revert f7fb07c // Year 2 commit's hash
```

```
Removing yr_2
```

```
...
```

```
$ git log --oneline
```

```
f3fc335 (HEAD -> timeline) Revert "Year 2"
```

```
727642d (origin/timeline) Revert "Year 3"
```

```
f7cf1cb Year 3
```

```
f7fb07c Year 2
```

```
e4df7f2 Year 1
```

```
03098e7 (origin/master, master) README file created
```

コミットハッシュで過去に戻る

```
$ ls
```

```
  README.md  yr_1
```

```
$ git push origin timeline
```

git revert [commit hash]

- 前進する取り消しコマンド。
- 指定された[commit]によって加えられた変更を反転し、新しいコミットとして結果を追加します。

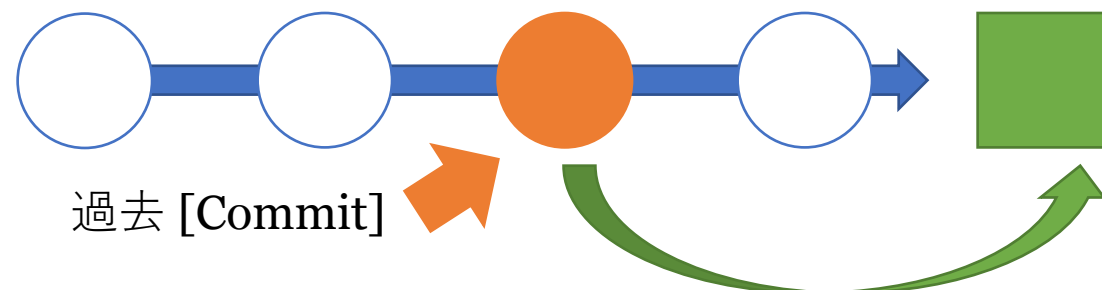
Reset vs Revert

git reset [commit]

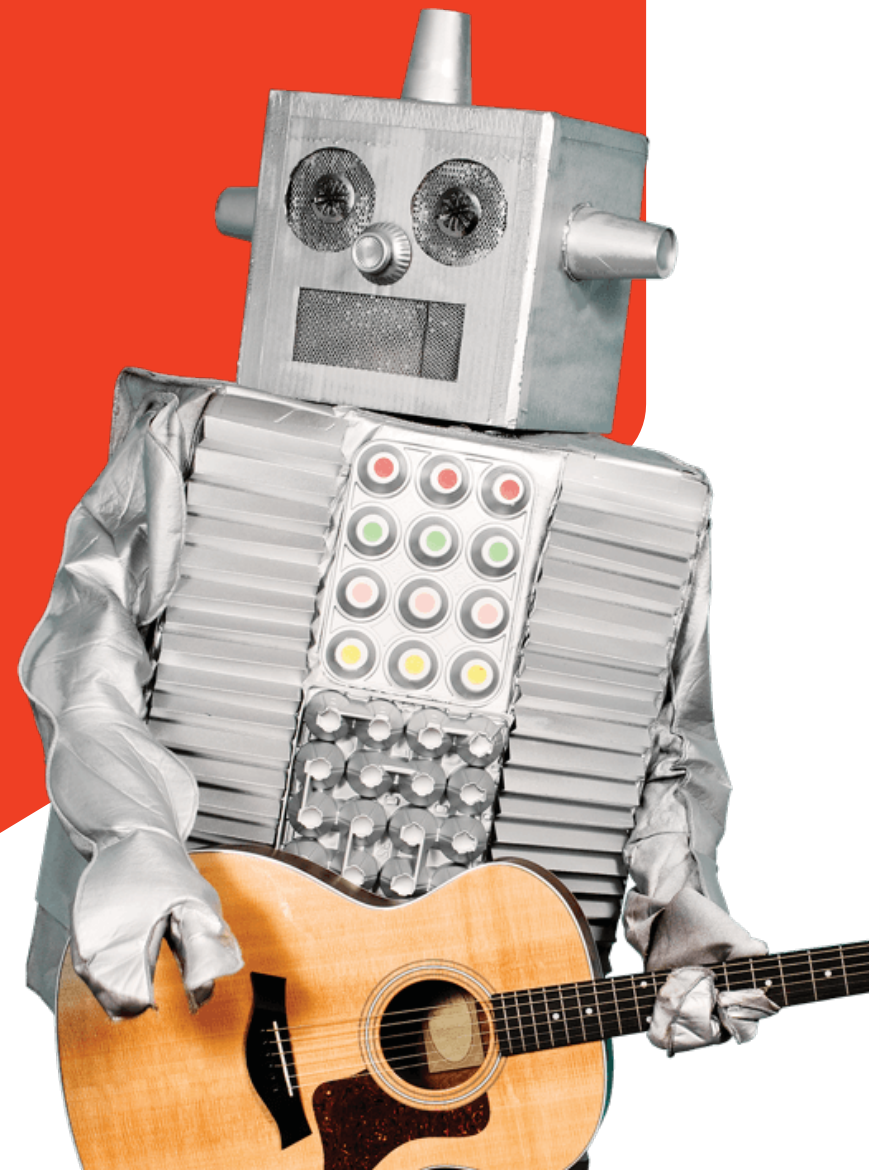
- 元に戻る取り消し操作の削除
- 過去の[commit]に戻り、それまでのすべてのコミットを削除します。
- これはすべてをクリーンアップします。
- ただし、削除されたコミットの履歴は失われます。

git revert [commit]

- 前進する取り消し操作
- 過去の[commit]で新しいコミットを作成します。
- コミットは削除されません。
- 公開/共有リポジトリに使用



Git Command- Line カンニング ペーパー



Create a Repository

Command	Description
<code>git init</code>	ローカルリポジトリを作成します
<code>git clone [url]</code>	GitHub(リモートリポジトリ)からリポジトリをクローン(ダウンロード)します
<code>git remote add origin [url]</code>	ローカルリポジトリをGitHub(リモートリポジトリ)に接続します

URL例 = `https://github.com/user/repo`

<https://github.github.com/training-kit/downloads/ja/github-git-cheat-sheet/>

Branches

Command	Description
<code>git status</code>	コミット可能なすべての新規または変更のあるファイルを一覧で表示します
<code>git branch [branch-name]</code>	新規ブランチを作成します
<code>git branch -d [branch]</code>	指定されたブランチを削除します
<code>git checkout [branch]</code>	指定されたブランチに移動します
<code>git merge [branch]</code>	指定されたブランチを現在のブランチにマージします

Sync Changes Between Local & Remote

Command	Description
<code>git fetch [remote] [branch]</code>	GitHub(リモートリポジトリ)から最新情報をローカルリポジトリへダウンロードします
<code>git push [remote] [branch]</code>	ローカルリポジトリのコミットをGitHub(リモートリポジトリ)にアップロード(プッシュ)します
<code>git pull [remote]</code>	GitHubの変更履歴をローカルにダウンロードします

Snapshotting

Command	Description
<code>git add [file]</code>	作業ディレクトリからステージングエリアへfileを追加します
<code>git status</code>	コミット可能なすべての新規または変更のあるファイルを一覧で表示します
<code>git add -A</code>	新規及び変更されたすべてのファイルをステージングエリアに追加します
<code>git commit -m "[message]"</code>	メッセージと共に、リポジトリへ変更を記録します
<code>git rm -r [file]</code>	Gitの管理対象からファイルまたはディレクトリを削除する

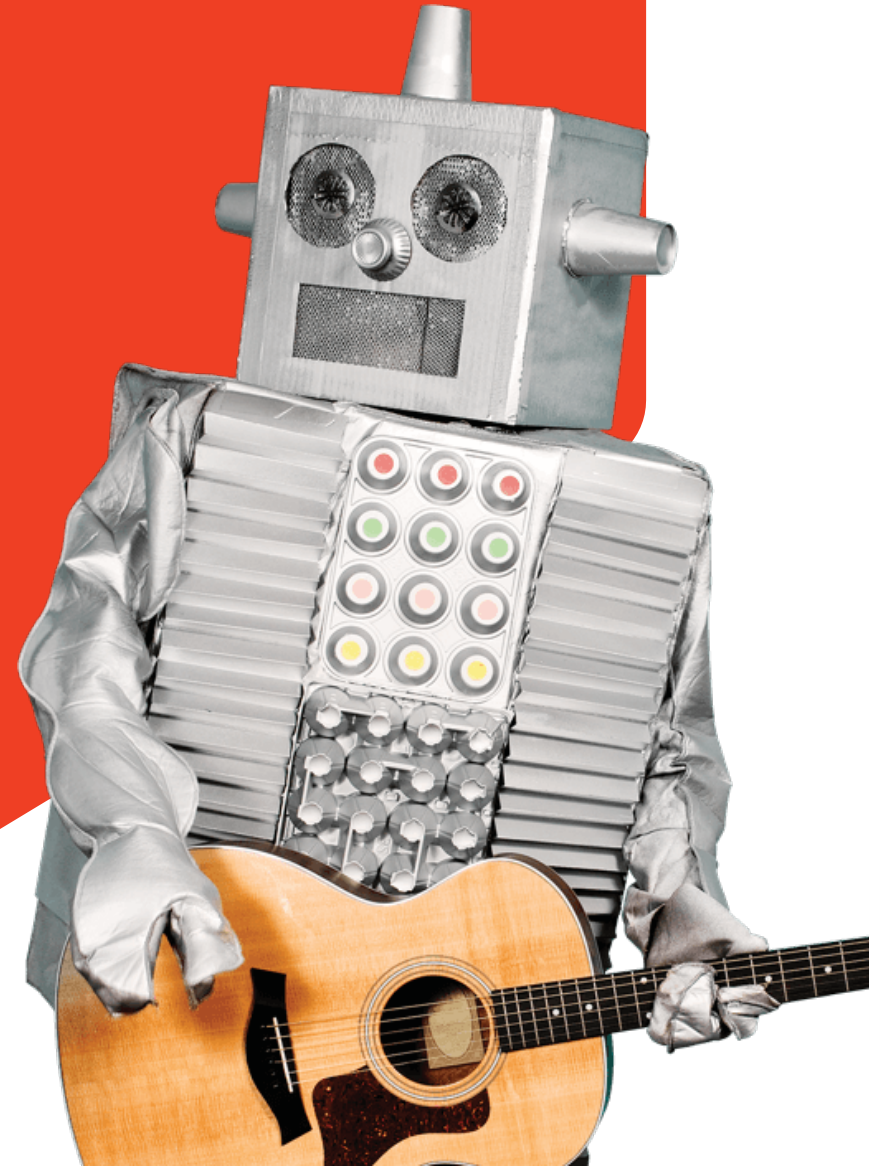
Make Changes

Command	Description
<code>git log</code>	コミット履歴を一覧で表示します
<code>git log --follow [file]</code>	名前の変更を含む指定したファイルのバージョン履歴の一覧を表示します
<code>git diff [branch A] [branch B]</code>	2つのブランチ間の差分を表示します
<code>git show [commit]</code>	指定されたコミットのメタ情報と変更内容を表示します

Redo Commits

Command	Description
<code>git reset [commit]</code>	指定したcommitの状態まで強制的に戻します
<code>git revert [commit]</code>	指定したcommitを取り消すためにコマンドです
<code>git revert HEAD</code>	直前のコミットを取り消すための新しいコミットを作成します

GitHub Desktop App





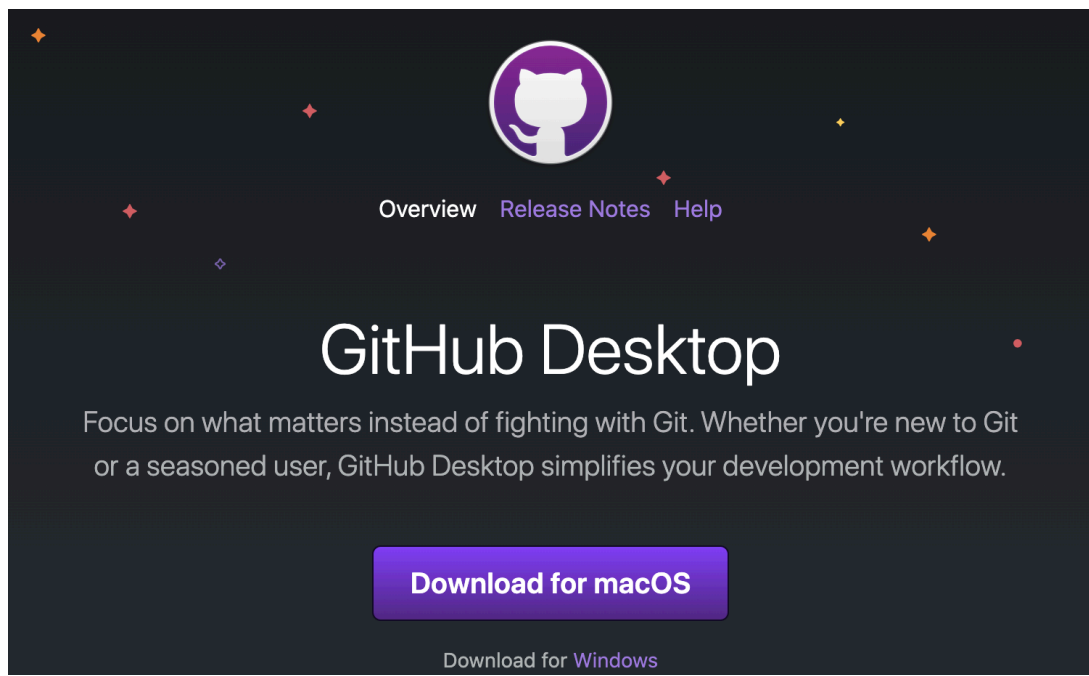
GitHub Desktop App

ダウンロード :

- <https://desktop.github.com/>

GitHub Desktopのドキュメント

- <https://help.github.com/ja/desktop>





```
$ cd Documents/Kintone_Dojo_3
$ git branch
  master
* timeline
$ git log --oneline
1c787bb (HEAD -> timeline, origin/timeline) Revert "Year 2"
7c60e0e Revert "Year 3"
f0c82a0 Year 3
62c6bd2 Year 2
128c014 Year 1
03098e7 (master) README file created
```

```
$ git reset f0c82a0
```

Unstaged changes after reset:

```
D yr_2
```

```
D yr_3
```

```
$ git log --oneline
```

```
f0c82a0 (HEAD -> timeline) Year 3
```

```
62c6bd2 Year 2
```

```
128c014 Year 1
```

```
03098e7 (master) README file created
```

```
$ git push origin timeline
```

```
To https://github.com/ahandsel/kintone_dojo_3.git
```

```
! [rejected]        timeline -> timeline (fetch first)
```

```
error: failed to push some refs to
'https://github.com/ahandsel/kintone_dojo_3.git'
```

```
hint: Updates were rejected because the remote contains
work that you do not have locally. This is usually caused
by another repository pushing to the same ref. You may want
to first integrate the remote changes (e.g., 'git pull
') before pushing again
```