



Part 1 はじめの一歩

gitが管理するもの

特定のフォルダを指定し、その中にある ファイル・フォルダのバージョンを管理する





git プロジェクトを作る

```
$ mkdir git-sample
$ cd git-sample
$ git init
Initialized empty Git repository in /.../git-sample/.git/
```

※ コマンドでやっているけど GUI でもいいよ



git の登場人物

- 1. ブロブ
- 2. コミット
- 3. リポジトリ

(Blob)

(Commit)

(Repository)





ブロブ

古いバージョンのファイルに別名を与えたもの

ということでまずは適当にファイルを作る

```
$ cat sample.csv
71,fb2735f5-88f3-48ec-8b7a-109c1cfea7f0,田中 愛菜
83,bd3f7e8f-99c4-4882-b0be-5cc26670f92a,山本 美咲
51,ccf3a690-d087-4c69-b843-ccf7b1f393ed,鈴木 大翔
```

コミット

= プロジェクトのブロブ + 親コミット

バージョンの時系列・前後関係を与えている!

試しにコミットを作ってみよう

\$ git add . && git commit -m "First commit"
[master (root-commit) a117493] First commit
 1 file changed, 3 insertions(+)
 create mode 100644 sample.csv



コミットの中身

実はコミットには他に以下の3つを保存してる

- ・ 誰がコミットしたか (名前とメールアドレス)
- ・いつコミットしたか
- コミットメッセージ ("First commit" の部分)



コミットの中身

次のエラーメッセージは「誰が?」が足りない

```
Your name and email address were configured automatically based on your username and hostname. Please check that they are accurate.

You can suppress this message by setting them explicitly:

git config --global user.name "Your Name"
```

git config --global user.email you@example.com

リポジトリ

ブロブやコミットを保存しておく貯蔵庫 gitというフォルダがリポジトリの本体

git というフォルダの中に 全ての古いバージョンのファイルや コミットが詰まっている

git の3つの世界

- 1. ワークツリー
- 2. インデックス
- 3. ヘッド

(Work Tree)

(Index)

(HEAD)

この3つの段階を踏んでコミットを作る





ワークツリー

保存する前の段階で作業中の空間を表している 保存する前なのでブロブがない

試しに sample.csv を書き換えてみよう

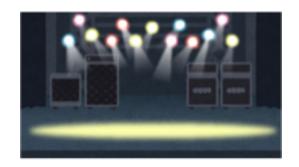
```
@@ -1,3 +1,3 @@ 
-71,fb2735f5-88f3-48ec-8b7a-109c1cfea7f0,田中 愛菜
+72,fb2735f5-88f3-48ec-8b7a-109c1cfea7f0,田中 愛菜
83,bd3f7e8f-99c4-4882-b0be-5cc26670f92a,山本 美咲
51,ccf3a690-d087-4c69-b843-ccf7b1f393ed,鈴木 大翔
```

インデックス

コミットが作られる前の段階

次のコマンドで持ち上げれる

\$ git add sample.csv





ヘッド

現在プロジェクトに反映しているコミット

試しにコミットを作ってみましょう

```
$ git commit -m "Change sample.csv"
[master 0757244] Change sample.csv
1 file changed, 1 insertion(+), 1 deletion(-)
```





まとめ

- ・ブロブ: 古いバージョンのファイル・フォルダ
- ・コミット: プロジェクトのブロブ+親コミット
- ・リポジトリ: ブロブやコミットの貯蔵庫
- ワークツリー・インデックス・ヘッドの3段階

でバージョン管理