

# 目次

## 目次

### Gitとはなにか

- Gitとは一言でいうと
- なぜソースコードを管理することが必要なのか？
- Gitの変更履歴はどこに保存されるの？
- リポジトリとは？
  - ローカルリポジトリ
  - リモートリポジトリ
  - なんで2種類あるの？

### Gitにおける開発の流れ

- Gitリポジトリの取得（準備）
- 変更内容のリポジトリへの記録
  - 三つの状態
    - Gitディレクトリ
    - ステージングエリア
    - 作業ディレクトリ
  - 変更内容をリポジトリに記録するためには？
- ブランチ機能について
  - ブランチ機能とは
  - masterブランチ機能とは
  - ブランチは最終的にmasterに結合する

### まとめ

### 参考文献

# Gitとはなにか

## Gitとは一言でいうと

ソースコードをはじめとするファイルの変更履歴を管理することができるシステムです。

## なぜソースコードを管理することが必要なのか？

開発中に特定のファイルを元に戻す必要が出てきたときに、Gitを使えば簡単に戻すことができます。また、複数人に開発をしているときに誰がこのファイルを編集したか履歴が残れば、開発効率もよくなるはずです。

## Gitの変更履歴はどこに保存されるの？

Gitの変更履歴はリポジトリという入れ物に保存されます。

ディレクトリやファイルをリポジトリの管理下に置くことによって、ディレクトリが監視されて変更が保管されるようになります。

リポジトリには種類が2種類あります。

## リポジトリとは？

### ローカルリポジトリ

自分のPC（ローカル環境）に置くリポジトリです。自分のPCの中のファイルやディレクトリを監視することができます。

### リモートリポジトリ

外部のサーバーなどネットワークに置く、リポジトリです。

複数人に開発を行うときに共有して、ファイルやディレクトリを監視することができます。

## なんで2種類あるの？

複数人に開発を行うときは、リモートリポジトリが必須となります。リモートリポジトリは常に最新版の状態にしておき、ローカルリポジトリにコピーを行い、ローカルで変更を加え、リモートリポジトリにプッシュ（同期）を行い反映します。

※リモートとは・・・

ローカルの対義語であり、「遠隔の～」と指す言葉で、ITにおいては、ネットワーク上のと意味します。

Gitにおけるリモートとは、主にリモートリポジトリを指します。

ここからは実際に開発の流れに沿って、Gitについて説明いたします。

# Gitにおける開発の流れ

## Gitリポジトリの取得（準備）

実際に、開発をするときにはまずGitリポジトリを取得する必要があります。ここでは、2パターン想定できます。

一つは、自分のPCなどにある、既存のディレクトリやプロジェクトをGitにインポートするパターン

二つ目は、何か協力したいプロジェクトがあるときに、別のサーバーに存在しているリポジトリからクローンするパターンです。

これらの作業を行うと、リポジトリが作成されて変更をすることが可能になります。

続いて、どのように変更が加えられるかを説明したいと思います。

## 変更内容のリポジトリへの記録

### 三つの状態

まずはじめに、3つの概念について説明いたします。

作業ディレクトリ、ステージング・エリア、Gitディレクトリの三つです。

### Gitディレクトリ

リポジトリと同意義です。前述の通り、レポジトリには2種類あり、ローカルリポジトリとリモートリポジトリです。

### ステージングエリア

Gitリポジトリにコミットするファイルを置いとくエリアになります。ステージングエリアがあることによって、コミットする内容を選別することができます。

※コミットとは・・・

コミットとは、ディレクトリやファイルの状態を記録するための操作です。コミットをすることで現在の状態が日時や変更を加えた人の情報と一緒に保存され、いつでもその状態に戻れるようになります。

### 作業ディレクトリ

ファイルの変更などを行うエリアになります。

ファイルの変更を加えたらaddを行い、ステージングエリアにステージをおこないます。

※アッドとは・・・

新しいファイルを追加追加しただけでは、あるファイルの変更など監視されません。アッドを行うと追跡開始されるようになります。また、ファイルのステージング・マージ（後述）時に衝突が発生したファイルに対する「解決済み」マーク付けなどで使用します。

ファイルを追加するという意味合いではなく、コミットするファイルを追加するという意味合いになります。

#### **変更内容をリポジトリに記録するためには？**

Gitリポジトリを取得した直後には、Gitディレクトリと作業ディレクトリは同じ内容になっています。そして、作業ディレクトリで変更を加えて、ステージングエリアにaddしてGitディレクトリにcommitを行います。この一連の作業が、Gitのワークフローの基本となります。

これで、変更を行うことによってGitに変更履歴が保存されて、開発を進めることができました。

Gitで複数人で開発を行うときに、リモートリポジトリから自分のローカルにクローンを行い、修正等を行います。そしてリモートリポジトリに対して同期を行い進めますが、同じ機能の修正を複数人で行っていた場合に、リモートリポジトリのプロジェクトにおいて不具合が発生してしまう可能性があります。ここで使う機能がブランチになります。続いてブランチについて説明したいと思います。

## **ブランチ機能について**

### **ブランチ機能とは**

ブランチとは、開発の本流から分岐して本流の開発を邪魔することなく開発を進めることができる機能になります。

それぞれのブランチは独立しており、お互いに干渉しません。また、本流のブランチ以外のブランチは必ずあるブランチから分岐して作成します。

ブランチにはそれぞれ変更の履歴が保存されています。

順を追って説明します。

### **masterブランチ機能とは**

リポジトリを取得した時点で、最初のブランチが自動的に作成されます。これをmasterブランチと呼びます。

そして、masterから分岐して、ブランチを作成して開発を進めることになります。ブランチの名前は自由につけることができます。

そして、作成したブランチから更に分岐することも可能です。

開発者はブランチを自由に移動することが可能で、それぞれのブランチで修正を行って開発を進めます。

### **ブランチは最終的にmasterに結合する**

ブランチにて、変更や修正や新規機能の実装が完了したら、masterブランチに結合を行います。このことをmergeと呼びます。

masterブランチは常にデプロイできる状態にしておくのが一般的になります。つまり、本番環境にアップロードしても問題ない状態にしておく必要があります。

## まとめ

以上で、Gitにおける開発の流れを説明いたしました。ここでは実際にGitをつかうためのGitコマンドの説明は省きましたが、流れを覚えていれば、Gitコマンドさえ覚えれば使えるようになります。

また、Gitを便利に使えるGit Hubというウェブサービスもあります。多くの企業や開発者の間で使用しているので、こちらも併せて確認してみましょう。

## 参考文献

Git公式

<https://git-scm.com/book/ja/v2>