▶▶▶ 履歴の記録　コマンドライン　Web

新しくファイルを追加する

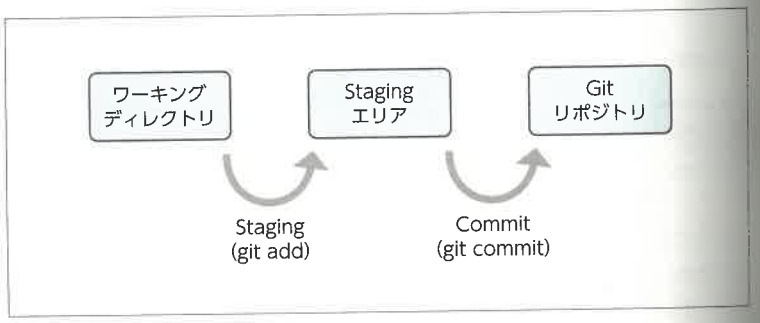
Staging(ステージング)とCommit(コミット)を実行して新しいファイルの追加を行います。

ファイルの追加・変更時の流れ

新しくファイルを追加したり変更するには、GitではStaging(ステージング)とCommit(コミット)の2段階の操作が必要となります。

Stagingという操作はGitの特徴の1つです。初めてGitを操作する際は、2段階の操作がややこしく感じるかもしれませんが、Gitの流れが理解できればうまく活用することが可能になります。Stagingがあることで、ローカルのファイルに加える変更とCommitを別に考えることができるようになります。ファイル内の一部の変更のみをStagingすることも可能となり、Commit作成に柔軟性が生まれ、履歴として見やすいCommitを作成することが可能となります。

ワーキングディレクトリは現在のローカルのファイルの状態を表しています。ワーキングディレクトリのファイルに加えられた変更はStagingエリアを経てGitリポジトリに変更履歴として保存されます。



Staging（ステージング）

Gitでは履歴の保存を行う前に、保存する単位をまとめて整理することができます。その際の一時的な作業領域をStagingエリアと呼び、保存したい変更をStagingエリアに追加することをStagingと呼びます。

新しく機能を追加する場合など、変更が複数のファイルにわたる場合、Stagingを行って、「意味のある」単位として保存しておくことにより、後からの変更が楽に行うことが可能となります。

ファイルのStagingには、git addコマンドを利用します。

書式

```
$ git add 追加したいファイルのパス
```

Commit（コミット）

Stagingを行って準備した後、Commitを行うことで、Stagingしたものを1つの履歴として保存することができます。Commitの際は、どのような変更を加えたCommitかをCommitメッセージして保存しておくことで、履歴をさかのぼってふりかえることが容易になります。

書式

```
$ git commit
```

コマンドを実行すると設定されたエディタが立ち上がり、Commitメッセージの入力画面となります。

書式

```
$ git commit -m "好きなメッセージ"
```

-mオプションを使用するとエディタを立ち上げることなく、Commitメッセージを設定してCommitを追加することも可能です。

Pull（プル）・Push（プッシュ）

Gitは分散型バージョン管理システムを採用しています。そのため、Staging、Commitを行ってもその変更はローカルリポジトリにしか反映されません。

ローカルリポジトリに加えた変更をGitHubなどのリモートリポジトリに反映させる場合は、そのための操作を行う必要があります。

ローカルリポジトリの変更をリモートリポジトリに反映させる場合は、git push

コマンドを実行します。

書式

```
$ git push origin ブランチ名
```

また一方でリモートリポジトリの変更をローカルリポジトリに反映させる場合、git pullコマンドを実行します。

書式

```
$ git pull origin ブランチ名
```

GitHub上で加えた変更などは忘れずgit pullコマンドを実行しておくようにしましょう。

コマンド 新しくファイルを追加する

hoge.txtというファイルを新規作成し、このファイルをGitにCommitして追加したい場合を考えます。

1 Stagingを行う

hoge.txtに必要な変更を加えた後、Gitに追加したい場合、まずStagingを行います。

```
$ git add hoge.txt
```

これでファイルのStagingが完了します。

2 Commitを行い、履歴を追加する

次にCommitを行います。

```
$ git commit
```

デフォルトのエディタが立ち上がるので、好きなCommitメッセージを入れることでCommitが追加されます。Commitメッセージは自由に入力できますが、以下のようなフォーマットが推奨されています。

```
変更の要約（72文字以下推奨）

3行目以降に変更内容の詳細を記入する

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
# Your branch is up-to-date with 'origin/master'.
#
# Changes to be committed:
# new file: hoge.txt
#
```

1行目はCommit内容の要約を簡単に記入します。1行空けて3行目以降に変更内容の詳細を記入します。Gitに関連する多くのソフトウェアが上記のスタイルを想定しているため、基本的にこのスタイルに沿ったCommitメッセージを書くことが推奨されています。また、「#」から始まる行は、Commitメッセージ作成時に自動的に挿入されていますが、コメント扱いとなりCommitメッセージには含まれません。

3 変更をGitHubに反映する

Pushを行い、ローカルリポジトリの変更をリモートリポジトリに反映させましょう。

現在のブランチがmasterの場合、以下のコマンドを実行します。別のブランチに変更する場合はブランチ名も指定してください。

```
$ git push origin master
```

Web 新しくファイルを追加する

1 ファイルを新規作成したいページまで移動する

「Create new file」というボタンをクリックします。