

ハンズオンタイム1



Macの人は**ターミナル**、windowsの人は**Gitbash**を準備してください。

ここからは、前回学んだqitの基本的操作を実際にやってみます!

前回あまり理解できなかった人でも実際に手を動かすことで、理解できると思いま す。

自分はMacのため、ターミナルの画面で説明していきます。

Windowsの人も基本的には同じなので、一緒に進めていってください。

グレーで囲まれているところをターミナル、またはGitbashで入力してください。

まず、qitを使うための設定をしていきます。最初に以下のコマンドで確認してくだ さい。



■ git config --list

いくつか情報が書かれていると思いますが、最低以下の2つが書かれてあれば大丈夫 です。

「user.name」は、自分のgitアカウントのユーザーネームです。

「user.email」は、qitアカウントを作成した際に使ったメールアドレスです。基 本、大学のかな?



🥧 user.name="gitのユーザーネーム" user.email="gitで設定したメールアドレス"

2つの情報が設定されていなかった人は、以下のコマンドをそれぞれ実行してくだ さい。



▶ git config --global user.email "gitで設定したメールアドレス"

最後にもう一度、「git config --list」で確認し、入力したユーザーネームとメールアドレスが確認できたら設定終了です。

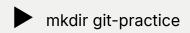
ここから実際に操作していきます。

ターミナルに慣れている人は、任意のディレクトリで進めてください。

慣れていない人は、以下で進めてください。これは、ホームディレクトリに移動するよという意味です。「pwd」で確認してみてください。



次に、ホームディレクトリに新たに「git-practice」というディレクトリを作ります。



作成したディレクトリに入ります。



「git-practice」をワークツリーとし、この中にgitディレクトリを作ります。 前回学んだ、「init」を使います。



以下のように、「.git」という隠しディレクトリができていれば大丈夫です。 この隠しディレクトリが自分たちの歴史を管理するために使われるものです。



Initialized empty Git repository in /Users/toriyabekako/gitpractice/.git/

今回は、練習のため自己紹介のテキストファイルgitで管理してみましょう! 以下のコマンドを入力してください。



touch self-introduction.txt

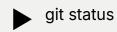
Macの人はFinder、Windowsの人は、エクスプローラーを開き「qit-practice」の 中に「self-introduction.txt」があることを確認してください。

その「self-introduction.txt」を開き、1行何か追加して保存してください。 できたら、閉じてOKです。

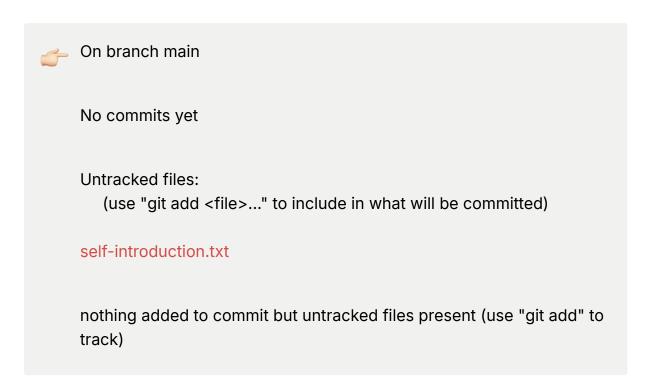
例)



前回の説明にはなかった便利なコマンドをここで使います。



以下のようなレスポンスが返ってくると思います。



このようにステータスを見ると、まだこのファイルは作成してから、qitの管理下に 置かれていません(コミットされていない)ということが読み取れます。そのた め、コミットをしていきましょう。コミットする前に、ステージ済みにするために インデックスに上げる必要があるということを前回学んだはずです。そのために は、「add」する必要があります。



git add self-introduction.txt

その後、またステータスを確認してみてください。



git status

次は、以下のレスポンスが返ってきたと思います。



On branch main

No commits yet

Changes to be committed: (use "git rm --cached <file>..." to unstage)

new file: self-introduction.txt

1つ前のステータスでは、「self-introduction.txt」が「Untracked files」の中に入 っていたと思います。しかし、今回は、「Changes to be committed」に入ってい ます。これは、コミットする準備ができていますよという意味です。(ステージ済み という意味)

コミットの準備ができたため、早速コミットしてみましょう。



git commit -m "Initial commit"

""の間には、コメントを入れることができます。コメントを入れる時は、「-m」を つけます。

今は、「self-introduction.txt」にプロフィールを追加したということをコミットで 記録しようとしています。それと同時に、コミットには、いろいろな記録が付与さ れます。例えば、誰が記録したか、いつ記録したかなどの情報です。その中の1つ に、qitが提供している**コミットコメント**というものがあります。これは、このコミ ットがどのようなことをしているかを一言で表す機能です。

今回は、最初のコミットという意味の"Initial commit"を付与します。

では、またステータスを確認してみましょう。



git status

以下のようなレスポンスが返ってくると思います。



On branch main nothing to commit, working tree clean

これは、最後にコミットしてから、ワーキングツリーになんの変更もありませんと いうことが書いてあります。これで全てのファイルがコミット済みということが確 認できました。

ここまで、常にステータスを確認してきましたが、git操作になれるまでは、何かと 操作をするたび確認することをおすすめします!状況の確認を行えるので、苦手な 人は、確認しながら進めてみてください!

次に、コミットしたファイルのどのような情報が付与されたのかを実際に確認して みましょう。



git log

以下のようなレスポンスが返ってくると思います。先ほど、説明した通り、誰が、い つ、どのようなコミットをしたのかが読み取れるはずです。一行目は、コミットを

一意に特定するためのハッシュ値になります。基本的に違うコミットであればハッ シュ値は被りません。

今はコミットを一回しかしていないため、1回分の情報しかありませんが、コミット を何回か行っていくうちに、数は増えていきます。



commit a2cba7ae1a1bfb45026f6ceefbc144082c6f2cf7 (HEAD → main)

Author: b2211700 <b2211700@photon.chitose.ac.jp>

Date: Sat Oct 2114:56:58 2023 +0900

Initial commit

これで、ローカルリポジトリでの基本的な操作はできているということになりま す。

あとは、回数を重ねて慣れていきましょう!