

GitHub

GitHubのインストール

GitHubの準備

バージョン確認：

```
$ git --version
```

Ubuntuの場合のインストール：

```
$ sudo apt-get install git-core
```

GitHubの初期設定：

```
$ git config --global user.name 'xxxx'
```

```
$ git config --global user.email 'xxxxx@xxx.com'
```

GitHubの設定確認：

```
$ cat ~/.gitconfig
```

```
LHL@iMac-air:~$ cat ~/.gitconfig
[user]
    name = tsulluailong
    email = un2226142477@gmail.com
LHL@iMac-air:~$
```

リポジトリの作成

-----リポジトリとは：Gitがファイルの履歴を保存している場所です。

まず初めに、作業を行うための新しいディレクトリを作成します。ディレクトリ名はなんでもいいですが、ここでは「findgrep」という名前にします。

新しいディレクトリを作成：

```
$ mkdir -p ~/git/findgrep
```

初期化（.gitファイルの作成）：

```
$ cd ~/git/findgrep
```

```
$ git init
```

```
LHL@iMac-air:findgrep$ git init
Initialized empty Git repository in /Users/LHL/git/findgrep/.git/
LHL@iMac-air:findgrep$ ls
.  ..  .git
LHL@iMac-air:findgrep$ ls -la
.  ..  .git
LHL@iMac-air:findgrep$
```

git initを実行すると、.gitというディレクトリが作成されます。

この.gitディレクトリはGitのリポジトリの実体です。

**！！.gitファイルを削除すると大変なことが起こります。
(自分の小宇宙が減びるぐらい) ！！**

ワークツリー

——リポジトリの内容をファイルとして展開する場所

ワークツリーでは普通のファイルを編集したり、追加・削除したりするところです。

リポジトリにファイルを追加

(以下では全部~/git/findgrepディレクトリに実行する)

バージョン管理するファイルを作成する：

```
$ touch findgrep.sh
```

```
$ chmod 755 findgrep.sh
```

```
$ vi findgrep.sh
```

findgrep.shファイルに以下のコードを書く

```
#!/bin/bash

pattern=$1
find . -type f | xargs grep -nH "$pattern"
```

このファイルをGitのリポジトリに履歴として追加してみましょう。

リポジトリに追加するには：

git add

git commit

git addは：
どのファイルをリポジトリに履歴として追加
するのかを指定するコマンドです。

仮の変更

git commitは：
実際にリポジトリにファイルの変更履歴を追
加する

本物の変更

今の時点では、findgrep.shファイルまだバージョン管理の対象になっていないため。
リポジトリに履歴として追加する：

```
$ git add findgrep.sh
```

続いて、git commitを実行して、実際の履歴に反映する

なお、git commitする際には、次のように-mオプションを指定して、今回の修正に対するメッセージを入力する：

```
$ git commit -m 'findgrep.sh新規作成'
```

```
LHL@iMac-cir:findgrep LHL$ git add findgrep.sh
LHL@iMac-cir:findgrep LHL$ ls
findgrep.sh
LHL@iMac-cir:findgrep LHL$ git commit -m 'findgrep.sh新規作成'
[master (root-commit) 5a4c671] findgrep.sh新規作成
1 file changes, 4 insertions(+)
create mode 100755 findgrep.sh
LHL@iMac-cir:findgrep LHL$
```

差分の表示と再コミット

これから、別の修正を行ってもう一度commitしてみましょう。

findgrep.shファイルを開いて、内容を次のように変更。

```
#!/bin/bash

pattern=$1
directory=$2
if [ -z "$directory" ]; then
    directory='.'
fi
find "$directory" -type f | xargs grep -nH "$pattern"
```

git status

——現在のワークツリーの状態を表示するためのコマンド

```
LHL@iMac-cir:findgrep LHL$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   findgrep.sh

no changes added to commit (use "git add" and/or "git commit -a")
LHL@iMac-cir:findgrep LHL$
```

6行目に「modified:」と表示したのはファイルの差分

git diff

——文字どおり、変更した内容をチェックする

```
LH ~-iMac-cin:findgrep LH $ git diff
diff --git a/findgrep.sh b/findgrep.sh
index 2e5c91d..de72f69 100755
+++ b/findgrep.sh
@@ 1,4 1,7 @@
 #!/bin/bash

 pattern=$1
-find . -type f -xargs grep -nH "$pattern"
+find "$1" -type f -xargs grep -nH "$pattern"
+
+if [ -z "$1" ]; then
+  echo "Usage: findgrep.sh [pattern]"
+fi
LH ~-iMac-cin:findgrep LH $
```

こういうの変更のチェックをお勧めします。

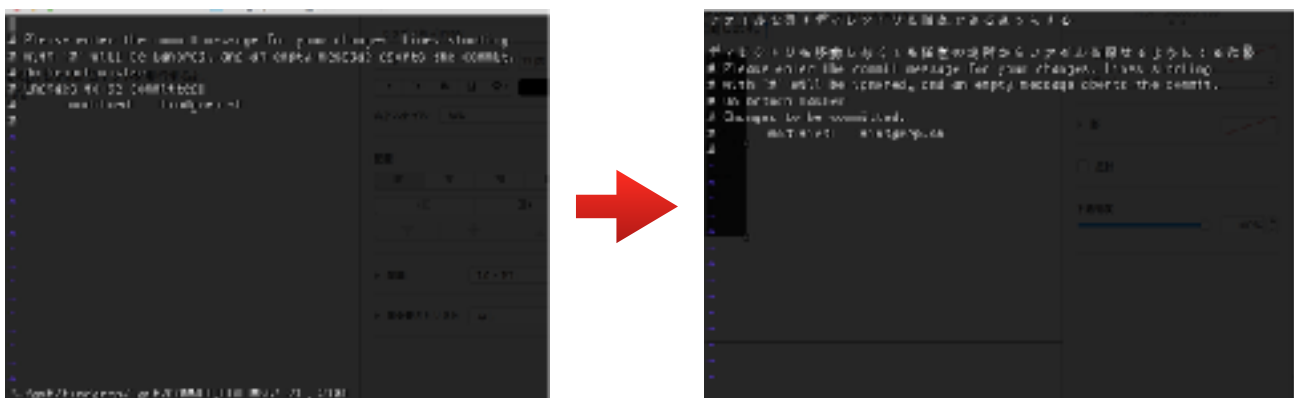
Gitでは新規作成でも編集でも手順は変わりません。

ここで、もう一度コミットしましょう。

```
$ git add findgrep.sh
```

```
$ git commit
```

今回では-mオプションを指定せず、そのままgit commitコマンドを実行すると、特に設定しない場合は、Vimエディタが起動します。



1行目 : 変更の概要

2行目 : 空行

3行目 : 詳細メッセージ

(:wqでVimを終了する)

変更がない場合はgit commitを実行したら、

このようなメッセージが表示されます。

——変更の履歴を確認する

今までは2回commitを行ったため、↑のように2つcommitが表示しています。

(各自やってみましょう！！)

git diff 7c883bfのような特定のcommitとの違いを確認することができます。

(各自やってみましょう！！)

ワークツリーとインデックス