

Git 講習 第 3 回

目次

- ・前回の復習
- ・ファイルの無視 (Gitignore)
- ・自分の行った変更の確認 (ステージング済みかどうか)
- ・コミット履歴の閲覧

前回の復習

- Git のブランチ機能

流れを引き継ぎながら安全に分岐して作業できる. マージすることで切る前のブランチに変更を反映できる.

- Git を使い始める

いくつかのコマンドで簡単に Git を使うことができる.

ファイルの無視

- Git を使っていると、「このファイルの変化について追跡する必要はないなあ」って思うことがあります.
- たとえば, Java コンパイル時に生成される .class ファイルなどは, 開いても人間が読めるものではないし, どこが変更されたか追跡しなくてもいいでしょう.
- このような場合, Git では .gitignore というファイルを作成してその中に記述することで, 特定のファイルを無視することが出来ます.

ファイルの無視

- .gitignore ファイルは, 普通にテキストエディタで作ります.
 - Windows なら \$ notepad .gitignore
 - それ以外なら, \$ nano .gitignore
- 例として nano を使っていますが, 別に他のエディタを使ってもいいです.
(gedit,vim,code など)
- このファイルに無視したいファイル名を書き込むか, ワイルドカードで特定のパターンのファイルをまとめて指定することも出来ます.
- 例) .class ファイルを全て無視 → *.class と記述
memo とファイル名の最初に付いているものを無視
→ memo* と記述.

実際の流れ (.class ファイルを全て無視)

```
saito@Kasagon-Ubuntu1804: ~/NMC/GitIgnoreTest
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)

saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ gedit test.java ← gedit (テキストエディタ)でtest.javaを作成, 編集
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ cat test.java ← catコマンドでファイル内容を出力
public class test{
    public static void main (String [] args ){
        System.out.printf("Hello World\n");
    }
}

saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ javac test.java ← コンパイル(test.classが生成される)
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ git status -s ← 現在の状況をシンプルに出力
 M test.java          ← Modified(修正された)の頭文字M
 ?? test.class         ← Javacで作られたGitにとって不明なファイル

saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ git add test.java ← test.javaだけをステージングしてコミット
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ git commit -m "Add:Create test.java"
[master 9747f1c] Add:Create test.java
 1 file changed, 1 insertion(+), 1 deletion(-)

saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ git status -s
 ?? test.class ← test.classは不明なファイルのままの状態

saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ gedit .gitignore ← .gitignoreファイルを作成&編集
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ cat .gitignore
*.class ← .gitignoreファイルの中身

saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ git add .gitignore ← .gitignoreファイルを追加
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ git status -s
A .gitignore ← 不明なファイルとして扱われていたtest.classが無視されている

saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ git commit -m "Ignore:.class file ignored"
[master e925cd1] Ignore:.class file ignored
 1 file changed, 1 insertion(+)
 create mode 100644 .gitignore

saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ git status
ブランチ master
nothing to commit, working tree clean
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$
```

自分の行った変更の確認

- 自分が行った変更がステージングされているかどうか確認するには, `$ git diff` を使います
- 変更したけれどステージングされていないならば変更された部分が出力されます.
- ステージングされている変更部分を見るときは
`$ git diff --staged` もしくは `$ git diff --chched` で確認することが出来ます.

自分の行った変更の確認

```
saito@Kasagon-Ubuntu1804: ~/NMC/GitIgnoreTest
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ code test.java
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ cat test.java
public class test{
    public static void main (String [] args ){
        System.out.printf("Hello World\n");
        System.out.printf("I use VSCode\n");
        printing();
    }
    public static void printing(){
        int integer=100;
        System.out.printf("%d\n",integer);
    }
}
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ git diff
diff --git a/test.java b/test.java
index e971dc7..84ffb87 100644
--- a/test.java
+++ b/test.java
@@ -1,5 +1,11 @@
 public class test{
     public static void main (String [] args ){
         System.out.printf("Hello World\n");
+        System.out.printf("I use VSCode\n");
+        printing();
+    }
+    public static void printing(){
+        int integer=100;
+        System.out.printf("%d\n",integer);
+    }
 }
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$
```


Git diff コマンド使用例

```
saito@Kasagon-Ubuntu1804: ~/NMC/GitIgnoreTest
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ git add test.java
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ git diff --cached
diff --git a/test.java b/test.java
index e971dc7..84ffb87 100644
--- a/test.java
+++ b/test.java
@@ -1,5 +1,11 @@
 public class test{
     public static void main (String [] args ){
         System.out.printf("Hello World\n");
+        System.out.printf("I use VSCode\n");
+        printing();
+    }
+    public static void printing(){
+        int integer=100;
+        System.out.printf("%d\n",integer);
+    }
 }

saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ git diff
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ git commit -m "Add:create printing method"
[master 8879d95] Add:create printing method
1 file changed, 6 insertions(+)
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ git diff
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ git diff --cached
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ git status
ブランチ master
nothing to commit, working tree clean
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$
```

該当する変更が無い場合は何も出力されません

履歴の確認

- コミット履歴の確認は `$ git log` コマンドを使います.
- このコマンドを実行すると, 新しい順に, コミットのハッシュ値, コミットした人, 日付, コミットメッセージ が出力されます.
- `$ git log -1` とすると直近の 1 コミットのみを表示します
- `$ git log -p` とするとコミットで反映された変更を表示します.
- `$ git log --graph` で(ブランチなどが)視覚的にわかりやすく表示されます

履歴の確認

```
saito@Kasagon-Ubuntu1804: ~/NMC/GitIgnoreTest
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ git log
commit 8879d95be632d8907503b66449428f97c076ce2b (HEAD -> master)
Author: m17saitou <m17saitou@gmail.com>
Date: Tue Sep 3 07:08:34 2019 +0900

    Add:create printing method

commit e925cd1c3e3fe876eb9d5e7b8ee0ca1384e068e3
Author: m17saitou <m17saitou@gmail.com>
Date: Tue Sep 3 06:08:36 2019 +0900

    Ignore:.class file ignored

commit 9747f1cadfc6e50bdeba3638f32edb647565d34c
Author: m17saitou <m17saitou@gmail.com>
Date: Tue Sep 3 06:07:21 2019 +0900

    Add:Create test.java

commit 500b6a8b206915fe1e3db79d0d1a25b12388b79e
Author: m17saitou <m17saitou@gmail.com>
Date: Tue Sep 3 06:04:51 2019 +0900

    Add:Create test.java
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$
```

```
saito@Kasagon-Ubuntu1804: ~/NMC/GitIgnoreTest
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$ git log -1 -p
commit 8879d95be632d8907503b66449428f97c076ce2b (HEAD -> master)
Author: m17saitou <m17saitou@gmail.com>
Date: Tue Sep 3 07:08:34 2019 +0900

    Add:create printing method

diff --git a/test.java b/test.java
index e971dc7..84ffb87 100644
--- a/test.java
+++ b/test.java
@@ -1,5 +1,11 @@
 public class test{
     public static void main (String [] args ){
         System.out.printf("Hello World\n");
+        System.out.printf("I use VSCode\n");
         printing();
+    }
+    public static void printing(){
+        int integer=100;
+        System.out.printf("%d\n",integer);
+    }
 }
saito@Kasagon-Ubuntu1804:~/NMC/GitIgnoreTest$
```

引数をいくつか併用することもできる

まとめ

- 無視したいファイルは `.gitignore` ファイルで無視するものを指定する
- 自分が行った変更点を確認するときは `$ git diff` コマンド
- 今までのコミット履歴を見るときは `$ git log` コマンド
- 各コマンドに引数を指定すると使い方をちょっと変えることができる