

**何もしてないのに突然
Submodule の差分が出て泣きそ
うになったら読む本**

mochikoAsTech 著

2020-09-12 版 mochikoAsTech 発行

はじめに

2020 年 9 月 mochikoAsTech

本書を手にとってくださったあなた、こんにちは！あるいは、はじめまして。「何もしないのに突然 Submodule の差分が出て泣きそうになったら読む本」の筆者、mochikoAsTech です。

想定する読者層

本書は、こんな人に向けて書かれています。

- 仕事で使っているリポジトリに Submodule がいる
- なぜか git pull しただけなのに Submodule の差分が出て納得いかない
- git checkout して別ブランチに移動しただけなのに Submodule の差分がでて納得いかない
- ちゃんと git pull したのに Submodule が更新されなくて困っている
- Submodule のフォルダまで移動して git pull しただけなのに差分が出てわけが分からない

マッチしない読者層

本書は、こんな人が読むと恐らく「not for me だった…（私向けじゃなかった）」となります。

- Git について何も知らないので 1 から学びたい
- プロジェクトで Submodule を導入すべきか迷っていて判断材料が欲しい
- いまからリポジトリに Submodule を入れようとしているので方法が知りたい

本書のゴール

本書を読み終わると、あなたはこのような状態になっています。

- Submodule の仕組みが分かっている
- 意図せぬ Submodule の差分が出た時に対処できる
- 読む前より Submodule がちょっと好きになっている

免責事項

本書に記載されている内容は筆者の所属する組織の公式見解ではありません。

また本書はできるだけ正確を期すように努めましたが、筆者が内容を保証するものではありません。よって本書の記載内容に基づいて読者が行なった行為、及び読者が被った損害について筆者は何ら責任を負うものではありません。

不正確あるいは誤認と思われる箇所がありましたら、必要に応じて適宜改訂を行いますので GitHub の Issue や Pull request で筆者までお知らせいただけますと幸いです。

<https://github.com/mochikoAsTech/SubmoduleUpdated>

目次

はじめに	3
想定する読者層	3
マッチしない読者層	3
本書のゴール	4
免責事項	4
 第 1 章 章タイトル	 7
1.1 Git とは	7
1.2 Submodule とは	7
1.3 サブモジュールの作り方	7
1.4 親子の関係	9
1.5 git pull しただけなのに差分が出た	9
1.6 git checkout しただけなのに差分が出た	9
1.7 Submodule で git pull しただけなのに差分が出た	9
1.8 正しい差分の無くし方	9
 あとがき	 11
PDF 版のダウンロード	11
Special Thanks:	11
レビューアー	11
参考文献・ウェブサイト	12
 著者紹介	 13

第 1 章

章タイトル

1.1 Git とは

1.2 Submodule とは

プロジェクト A とプロジェクト B の両方で同じライブラリを使いたい！ そのライブラリをメンテナンスするのはプロジェクト C のみなさんだとします。

このときプロジェクト A のリポジトリにも、プロジェクト B のリポジトリにもライブラリのソースコードを直接置いてしまう（プロジェクトのツリーに取り込んでしまうと）と、プロジェクト C のメンバーがライブラリをメンテナンスしたいときに、両方のリポジトリでそれぞれ更新しなければなりません。

Git のサブモジュールを使うと、プロジェクト C のリポジトリを、別のリポジトリのサブディレクトリとして扱えるようになります。プロジェクト A やプロジェクト B を親（メイン）としたとき、子（サブ）にあたるプロジェクト C のコミットは、親のコミットとは別で管理できます。

1.3 サブモジュールの作り方

まず親を作ります。

```
メインのリポジトリ (main_project) を作る  
$ git init main_project
```

続いてメインのリポジトリのサブモジュールとして、既に存在している別のリポジトリを追加します。

第 1 章 章タイトル

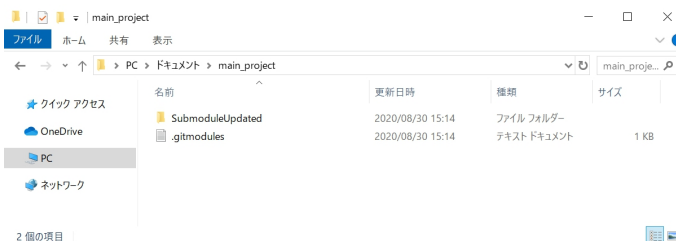
```
作ったリポジトリのディレクトリに移動する  
$ cd main_project
```

```
サブモジュールとして「SubmoduleUpdated」というリポジトリを追加する  
$ git submodule add https://github.com/mochikoAsTech/SubmoduleUpdated
```

今回はサブモジュールとして、本書の原稿リポジトリを追加してみました。^{*1}追加した SubmoduleUpdated をクローンしてくるため、こんな表示がされたと思います。このときサブモジュールの中身はクローンしてきますが、サブモジュールのさらにサブモジュール以下については再帰的にはクローンしてきてくれないので注意が必要です。

```
$ git submodule add https://github.com/mochikoAsTech/SubmoduleUpdated  
Cloning into 'C:/Users/mochikoAsTech/Documents/main_project/SubmoduleUpdated'..  
remote: Enumerating objects: 162, done.  
remote: Counting objects: 100% (162/162), done.  
remote: Compressing objects: 100% (143/143), done.  
remote: Total 162 (delta 43), reused 26 (delta 7), pack-reused 0 receiving objects: 100% (162/162), 518.33 KiB | 772.00 KiB/s, done.  
Resolving deltas: 100% (43/43), done.
```

サブモジュールを追加するとき、今回のように特にディレクトリ名を指定しないと、次のようにリポジトリの名前 (SubmoduleUpdated) がそのままディレクトリ名となります。(図 1.1)



▲ 図 1.1 SubmoduleUpdated がサブモジュールとして追加された

^{*1} 今回は GitHub の URL を指定しましたが、このような絶対の URL に限らず、`git submodule add ../SubmoduleUpdated` のような相対パスでサブモジュールのリポジトリを指定することも可能です。ただし相対パスで追加すると、サブモジュールのリモートリポジトリ (`remote.origin.url`) が `C:/Users/mochikoAsTech/Documents/SubmoduleUpdated` のようになってしまうので、特に理由が無ければリモートの URL で指定の方がお勧めです。

次のように末尾でディレクトリ名を指定してやれば、たとえばディレクトリ名を「sub」にした状態でサブモジュールを追加できます。

```
git submodule add https://github.com/mochikoAsTech/SubmoduleUpdated submodule
```

1.4 親子の関係

このとき Git のリポジトリを親、サブモジュールを子だとします。・サブモジュールの master ブランチのコミットが $A \rightarrow B \rightarrow C$ で進んでいる（C は最新だけど未リリースなやつ）・親の master は B を参照しているこの場合、親の master で git submodule update したら、サブモジュールは B を引っ張ってくるので、安心だと思っている。

submodule update とは、「"親が指定している子のコミット"に更新しておくれ」なんだと思っております。

1.5 git pull したただけなのに差分が出た

1.6 git checkout したただけなのに差分が出た

1.7 Submodule で git pull したただけなのに差分が出た

1.8 正しい差分の無くし方

あとがき

この原稿を書いたことで、「prh-rules（校正のルールを書いたファイルたち）をサブモジュールにすれば、本のリポジトリを作るたびに前回のリポジトリから prh-rules 持ってなくてもいいんじゃない！」と気づいてしまったので、本当に本を書くのは自分にとっていいことだなぁ、と思いました。まる。

数ある技術書の中から「Submodule 本（仮題）」を手にとってくださったあなたに感謝します。

2020 年 9 月
mochikoAsTech

PDF 版のダウンロード

本書（紙の書籍）をお買い上げいただいた方は、下記の URL から PDF 版を無料でダウンロードできます。

- ダウンロード URL
 - <https://mochikoastech.booth.pm/items/xxxxxxx>
- パスワード
 - xxxxxx

Special Thanks:

- ねこ

レビューアー

- かしこいねこ

参考文献・ウェブサイト

- git-scm.com

著者紹介

mochiko / @mochikoAsTech

テクニカルライター。元 Web 制作会社のインフラエンジニア。ねこが好き。「分からない気持ち」に寄り添える技術者になれるように日々奮闘中。技術書典で頒布した「DNSをはじめよう」「AWSをはじめよう」「技術をつたえるテクニック」「技術同人誌を書いたあなたへ」は累計で 9,700 冊を突破。

- <https://twitter.com/mochikoAsTech>
- <https://mochikoastech.booth.pm/>
- <https://note.mu/mochikoastech>
- <https://mochikoastech.hatenablog.com/>
- <https://www.amazon.co.jp/mochikoAsTech/e/B087NBL9VM>

Hikaru Wakamatsu

表紙デザインを担当。

Shinya Nagashio

挿絵デザインを担当。

何もしてないのに突然 Submodule の差分が出て泣きそうになったら読む本

2020 年 9 月 12 日 技術書典 9 初版

著 者 mochikoAsTech
デザイン Hikaru Wakamatsu / Shinya Nagashio
発行所 mochikoAsTech
印刷所 日光企画

(C) 2020 mochikoAsTech