目次

[●2. GitHubを使った文書の変更履歴管理 2](#_Toc427423070)

[**●2.1** **GitHubの利用準備** 2](#_Toc427423071)

[●2.1.1 GitHubアカウント作成 2](#_Toc427423072)

[●2.1.2 クライアントPCからGitHubへのアクセス設定 7](#_Toc427423073)

[**●2.2** **GitHubを使った文書管理** 10](#_Toc427423074)

[●2.2.1 文書変更のステップバイステップ 10](#_Toc427423075)

[●2.2.2 原本のコピーを自分の書棚に取得 10](#_Toc427423076)

[●2.2.3 自分の書棚の文書を改変 13](#_Toc427423077)

[●2.2.4 原本への改変採用依頼を出す 17](#_Toc427423078)

[●2.2.5 変更部分の確認 21](#_Toc427423079)

[●2.2.6 変更された原本の再取得 24](#_Toc427423080)

[**●2.3** **Microsoft Word形式ファイル以外の差分表示** 31](#_Toc427423081)

# GitHubを使った文書の変更履歴管理

## **GitHubの利用準備**

文書管理を多数の方と協調して進めるために役立つクラウドサービスがGitHubです。  
ここでは前章で準備の整ったGitをGitHubへ接続する設定についてお伝えします。

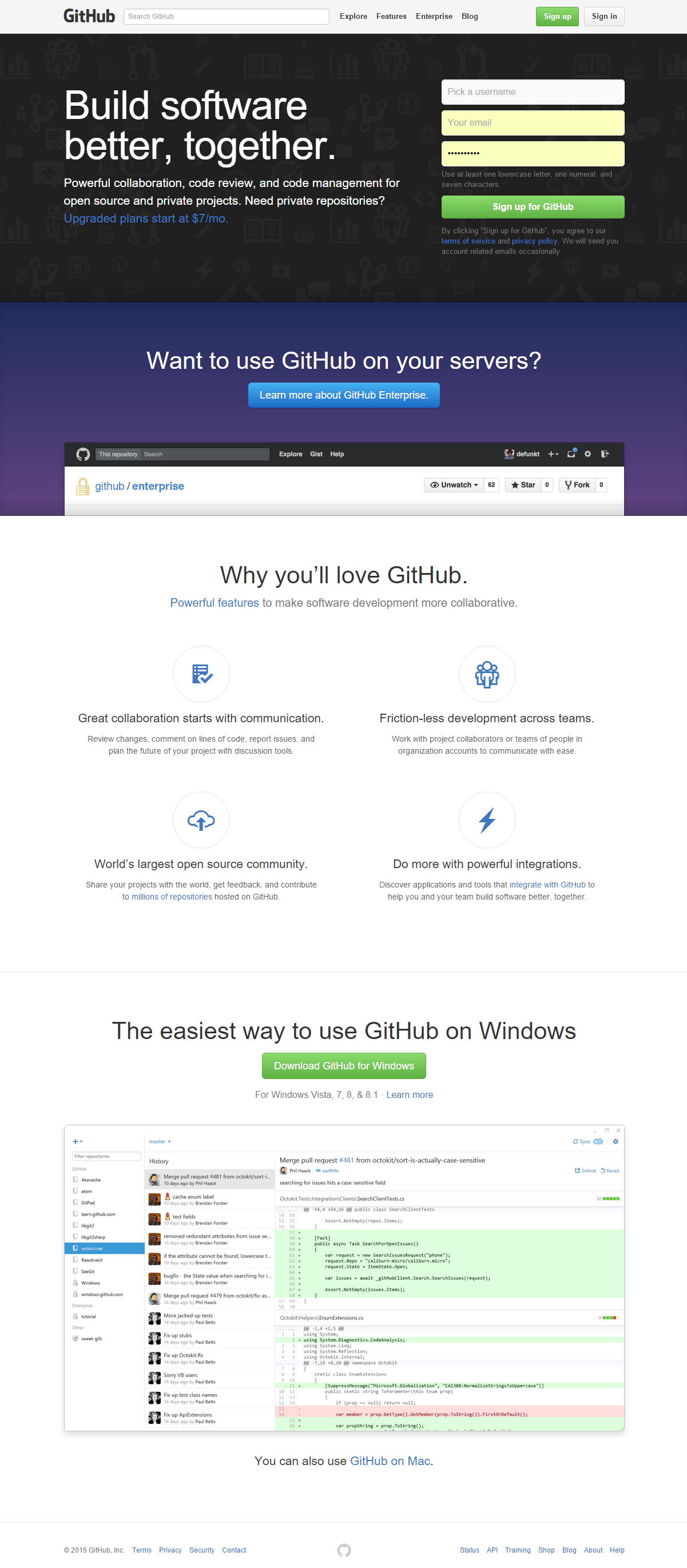
### GitHubアカウント作成

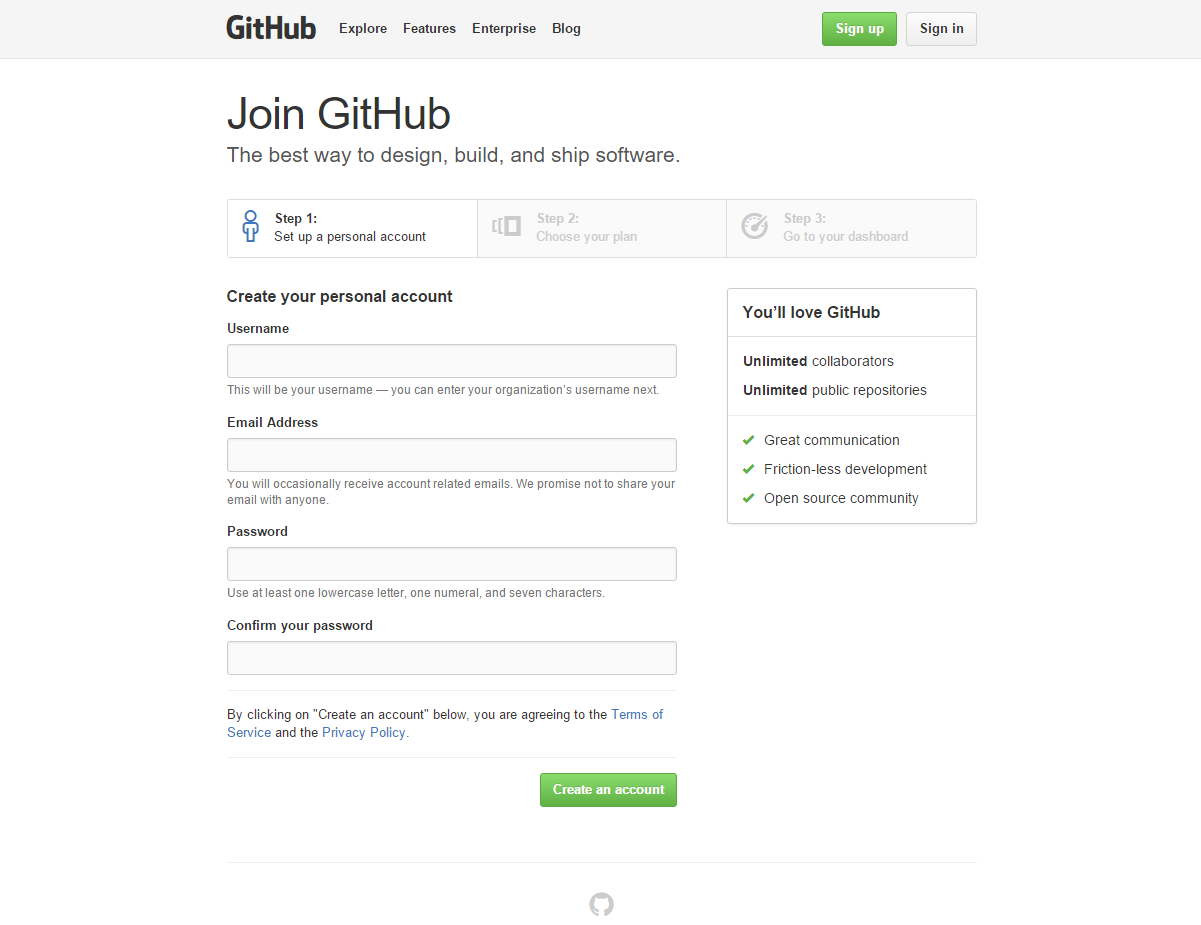
GitHubを利用するにはGitHubへのアカウント作成（ユーザー登録）が必要です。

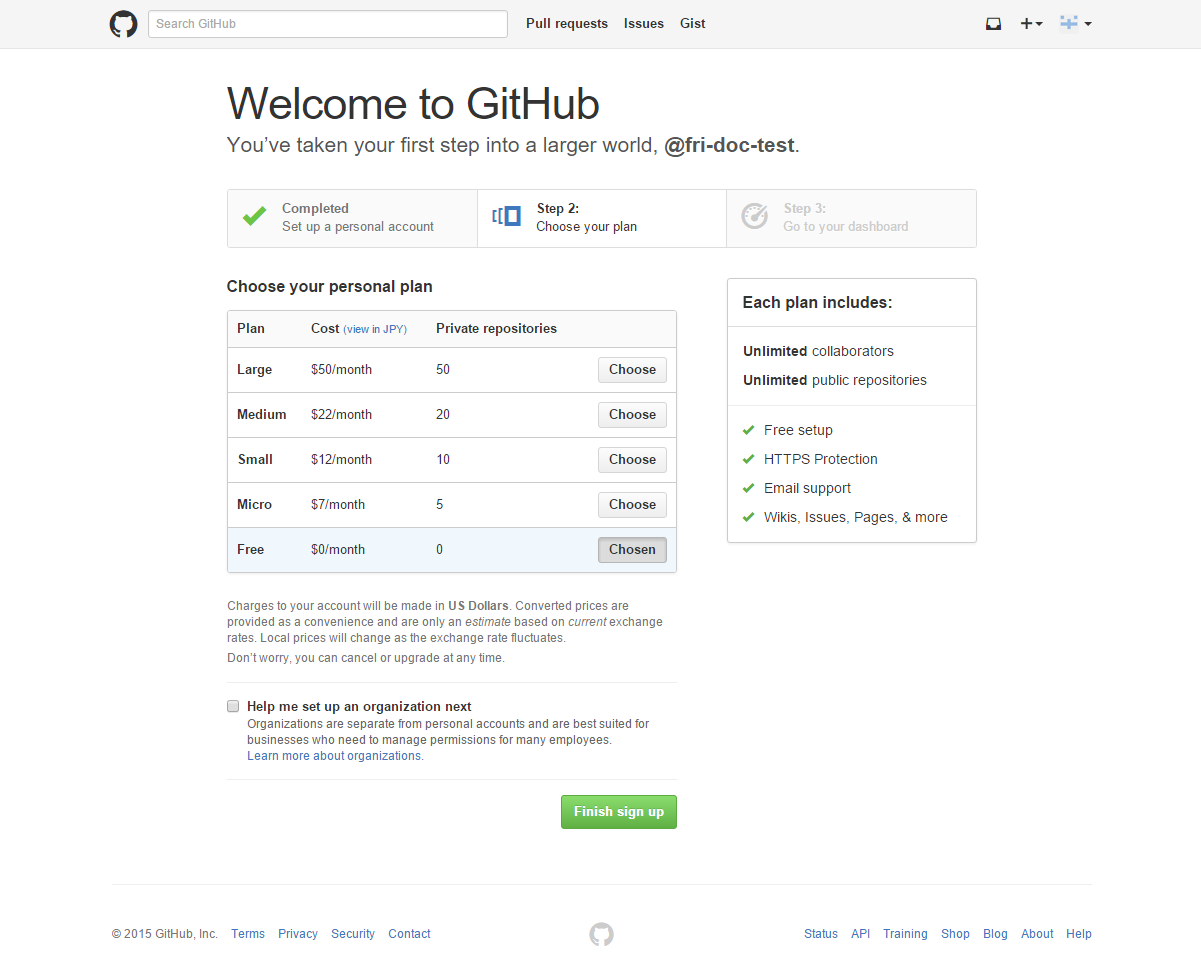
ユーザー登録は　<https://github.com/>　より下記手順に沿って行ってください。

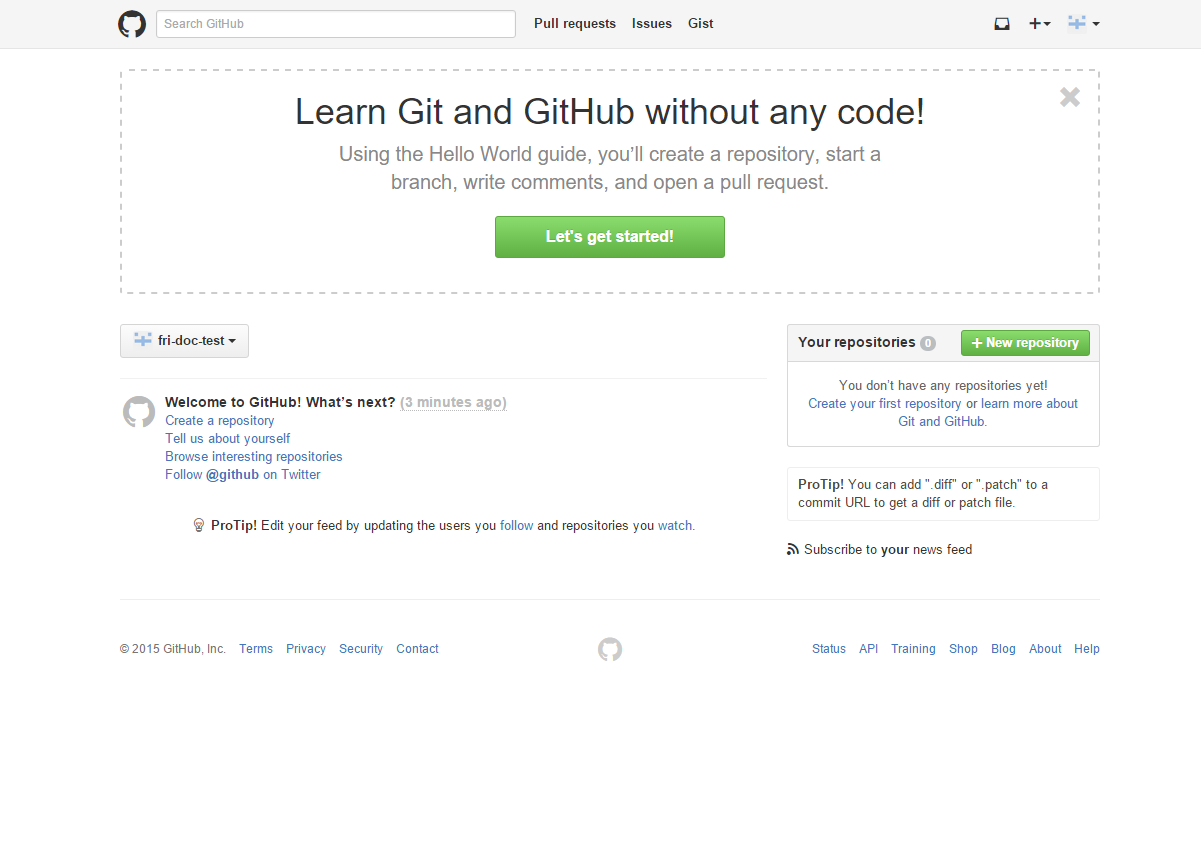
須藤さん

下記手順をGitの操作説明編のように整形をお願いできますでしょうか。

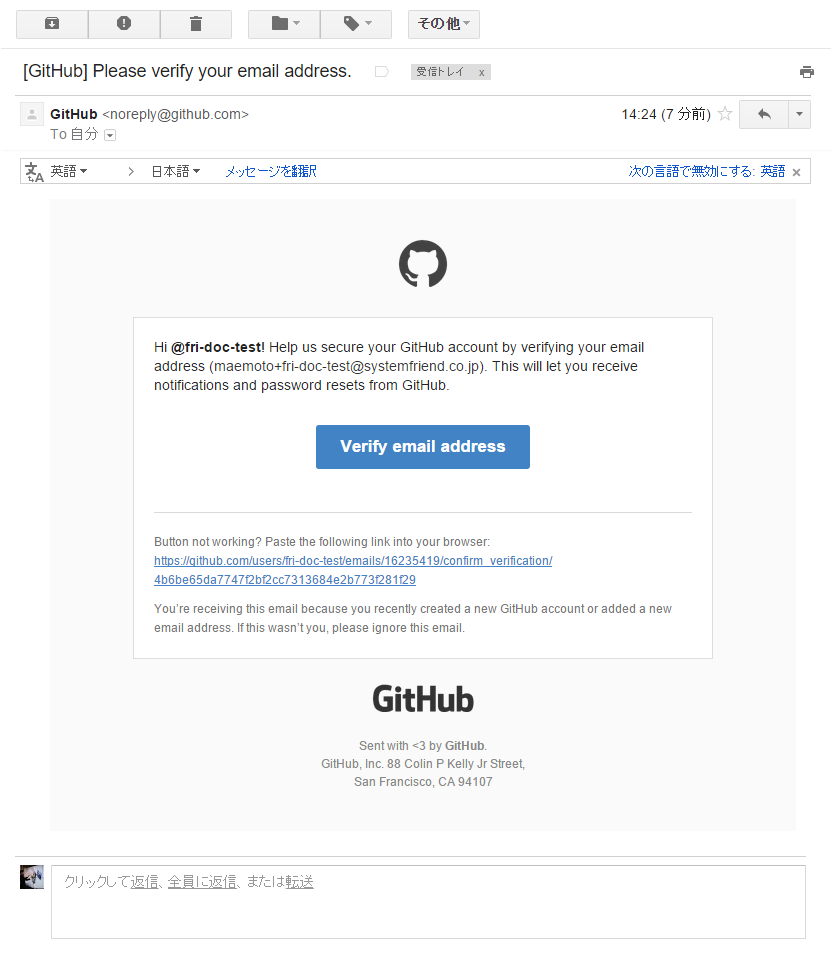








ここで確認メールが届くので「Verify email address」をクリックします。



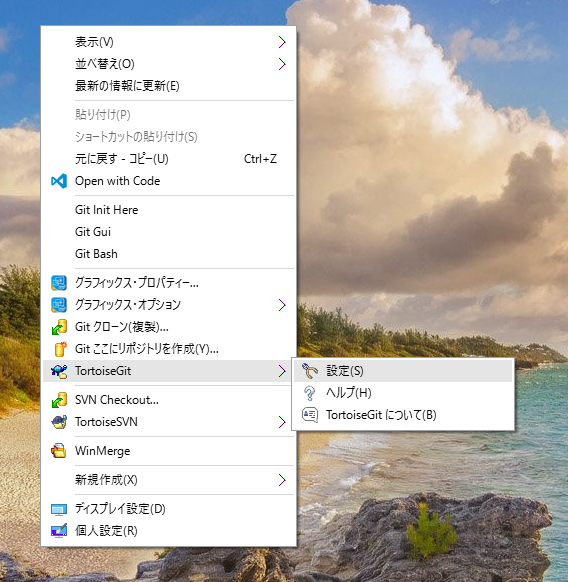
### クライアントPCからGitHubへのアクセス設定

次にお使いのPCからGitHubへアクセスするための設定を行いましょう。

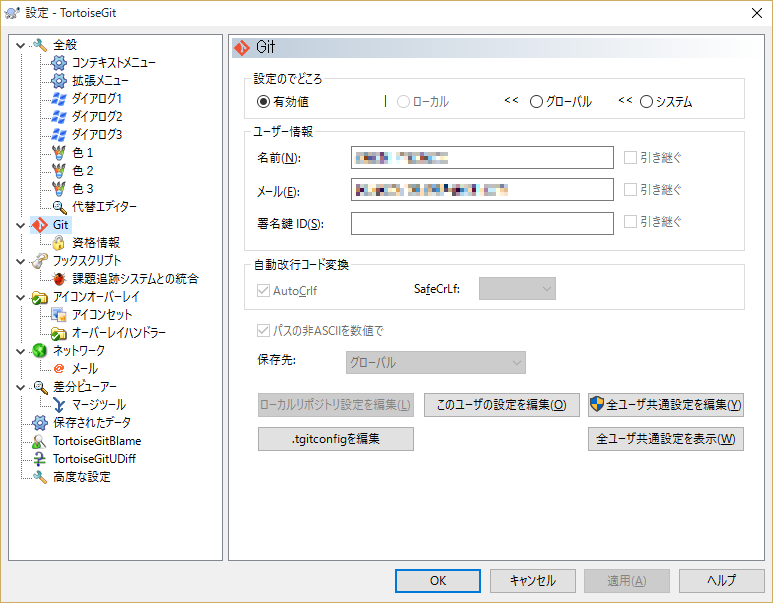
デスクトップで右クリックし、「TortoiseGit」→「設定」を選択します。

須藤さん

下記手順をGitの操作説明編のように整形をお願いできますでしょうか。



「Git」の設定項目でGitHubに登録したユーザー名とメールアドレスを入力して「OK」ボタンをクリックしてください。



以上でGitHubへのアクセス設定が完了しました。

## **GitHubを使った文書管理**

### 文書変更のステップバイステップ

ここまでの手順でGitHubを利用して文書管理を行う準備が整いました。  
それでは早速文書の変更を行ってみましょう。

大まかな流れは下記のように自分の書棚の文書への改変を行い、その内容をまとめて原本への改変依頼を出す流れとなります。

原本のコピーを自分の書棚に取得

原本への改変採用依頼を出す

自分の書棚の文書を改変

初回のみ

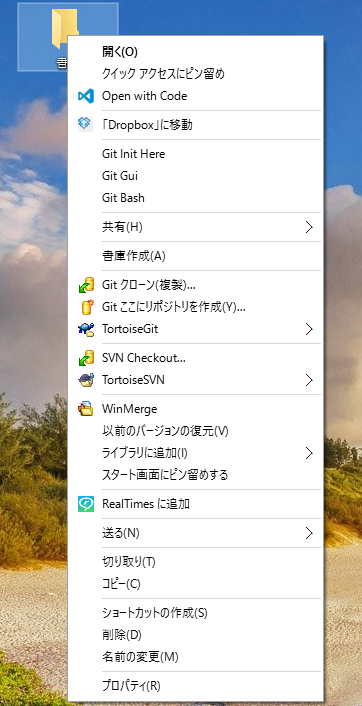
繰り返し

### 原本のコピーを自分の書棚に取得

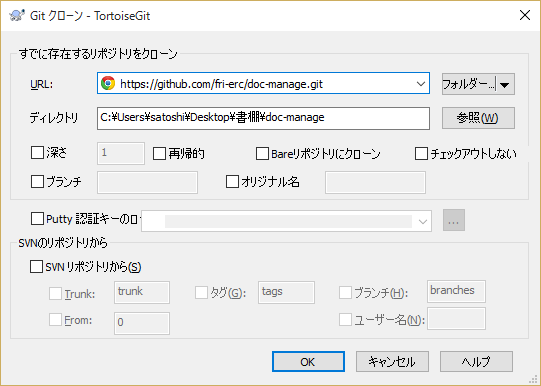
まず原本のコピーを自分の書棚に取得しましょう。  
そのためには自分の書棚を保管する任意のフォルダーを用意する必要があります。  
ここではデスクトップに「書棚」フォルダーを作成してください。



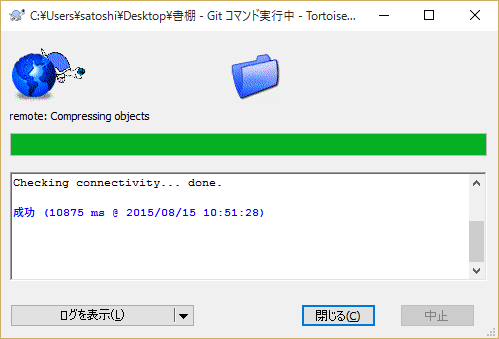
「書棚」フォルダーを右クリックしてメニューから「Gitクローン(複製)…」を選択してください。



「Gitクローン」の画面が表示されますので、URLに「https://github.com/fri-erc/doc-manage.git」と入力して「OK」ボタンをクリックしてください。  
このURLは原本を指すURLなので正確に打ち込んでください。



処理が完了すると下記のように「成功」と表示されますので「閉じる」をクリックして画面を閉じましょう。



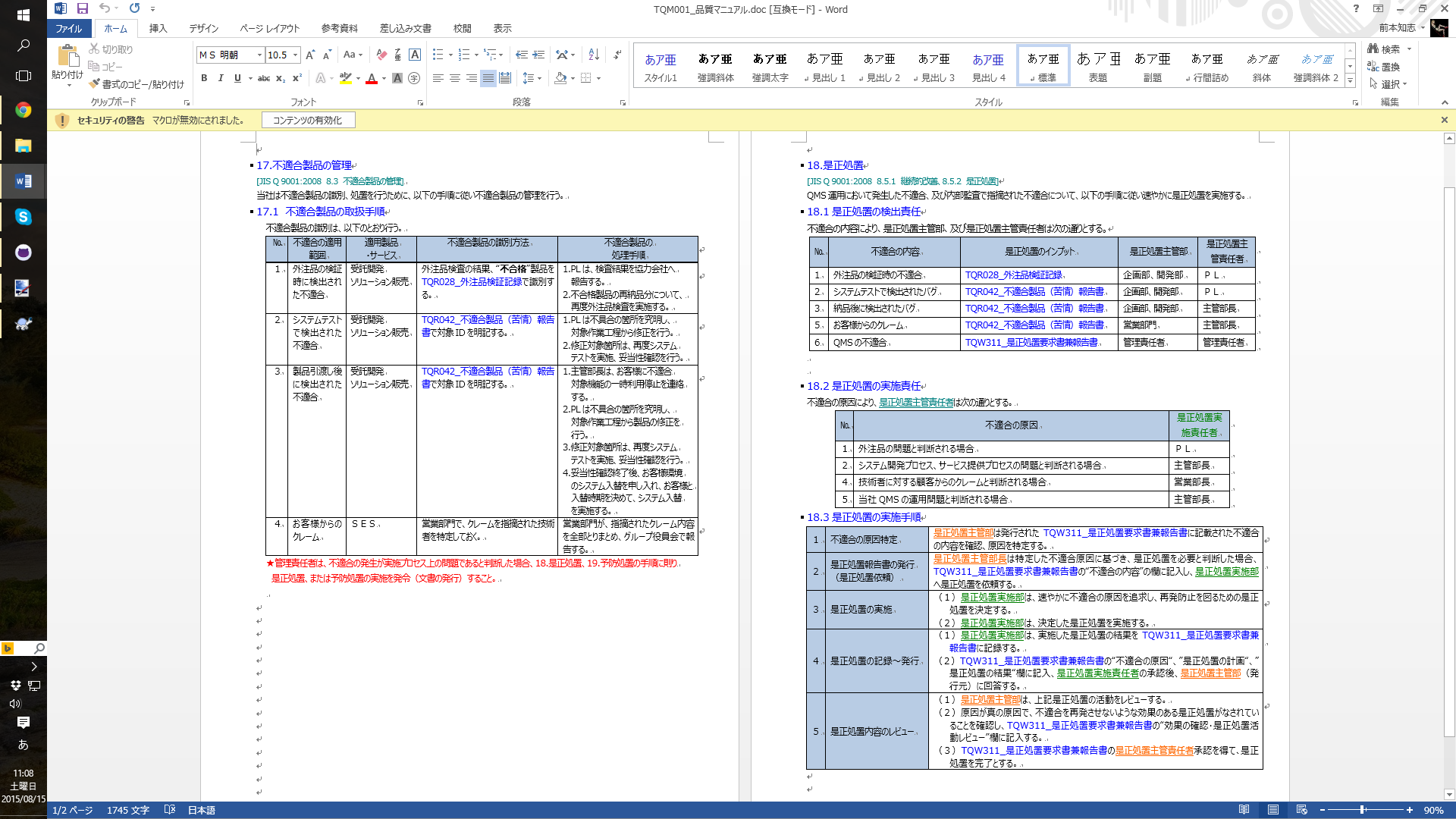
以上で原本のコピーを自分の書棚に取得することができました。

「書棚」フォルダーを開いて中を確認しましょう。  
doc-manage\サンプル文書 フォルダ内に下記のようにサンプル文書がダウンロードされています。

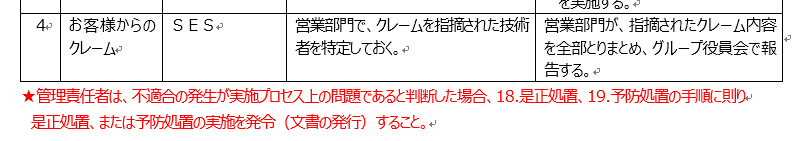


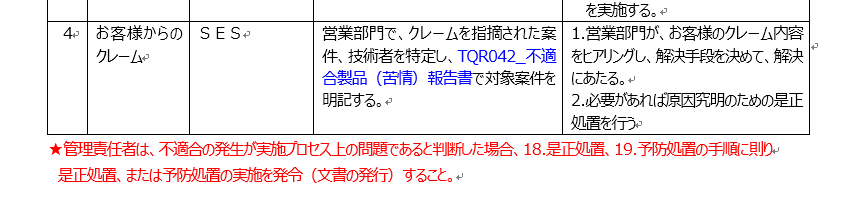
### 自分の書棚の文書を改変

それでは次にサンプル文書に改変を加えてみましょう。  
サンプル文書「TQM001\_品質マニュアル.doc」をダブルクリックしてMicrosoft Wordで開いてください。

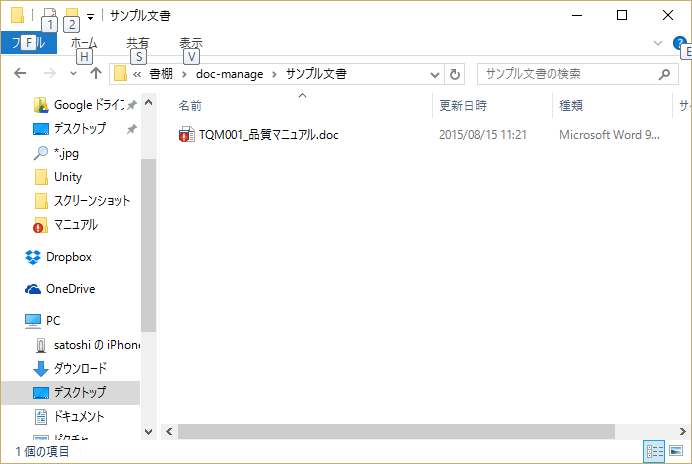


17.1部分の「不適合製品の取扱手順」-「４．お客様からのクレーム」の処理手順への改変が発生したこととして、文書を下記のように修正して上書き保存してください。



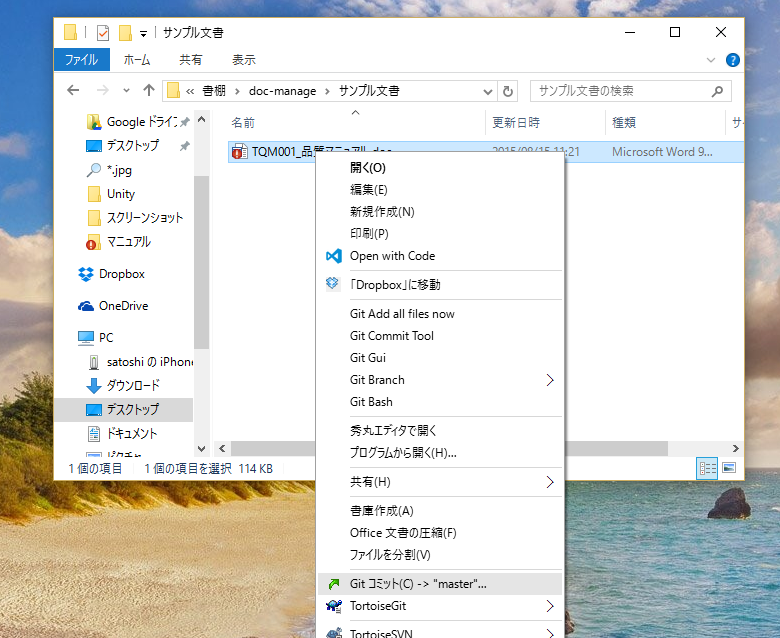


自分の書棚の文書に変更を加えたことはフォルダー内の表示でもアイコンが赤くなることで確認できます。



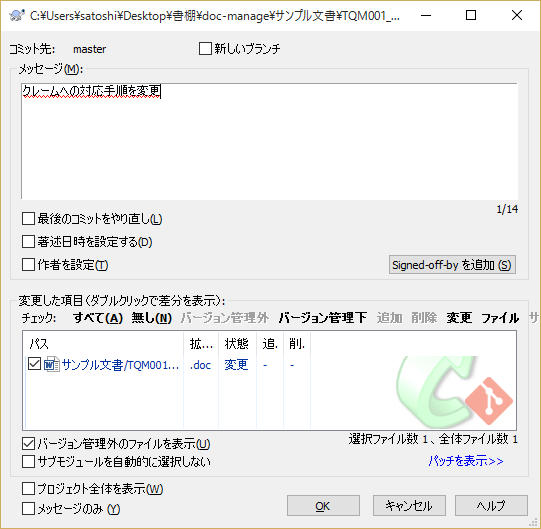
最後に文書の改変を確定（コミット）させます。この確定は自分の書棚に対するもので、この確定作業の単位で変更内容をキャンセルするような操作も可能となりますので区切りがよいところで確定をしておくことをおすすめします。

フォルダー内の改変文書を右クリックして「Gitコミット～」を選択してください。



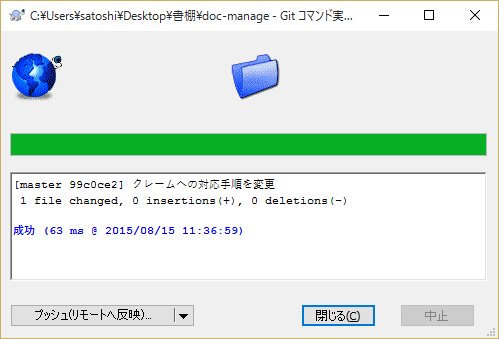
Gitコミットの確認画面が表示されます。

「メッセージ」欄には改変内容の簡単な説明を書いておきましょう。  
説明を書いておくことで後に変更履歴を表示した際に改変内容がわかりやすくなりますので必ず記述する習慣をつけましょう。



「メッセージ」を記述したら「OK」ボタンをクリックします。

下記のように「成功」と表示されれば自分の書棚への確定処理は完了です。



ここまでの手順で下記の図にあるローカルリポジトリへの改変が行えたことになります。

原本

リモートリポジトリ

（共有書棚）



Ｂさん

Ａさん

写本

写本

ローカルリポジトリ

（自分専用書棚）

ローカルリポジトリ

（自分専用書棚）

写本を基に

変更したファイル

写本を基に

変更したファイル



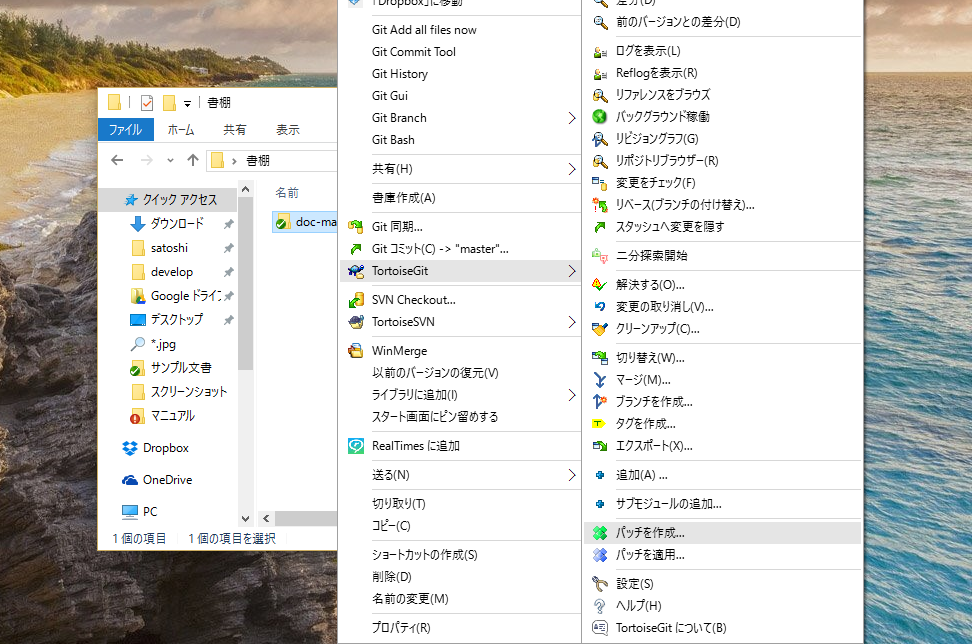
### 原本への改変採用依頼を出す

前項までで自分の書棚の文書を改変することはできました、しかしこの状態では他の方にその改変内容を見てもらうことができません。

それで自分の書棚の文書への改変を原本へも取り込んでもらえるように原本の管理者に改変採用依頼を出す必要があります。

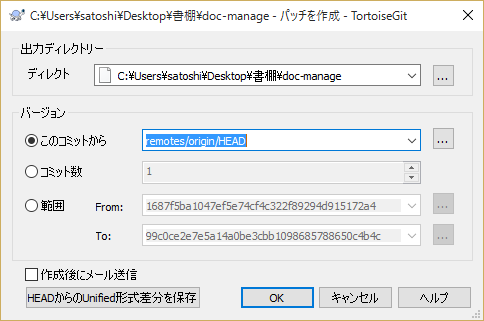
この作業には色々な方法がありますが、ここでは最もシンプルな改変内容をまとめたファイルを原本の管理者にメールで送付する方法を紹介します。

自分の書棚フォルダー（デスクトップ\書棚\doc-manage）を右クリックし、「TortoiseGit」→「パッチを作成…」を選択してください。

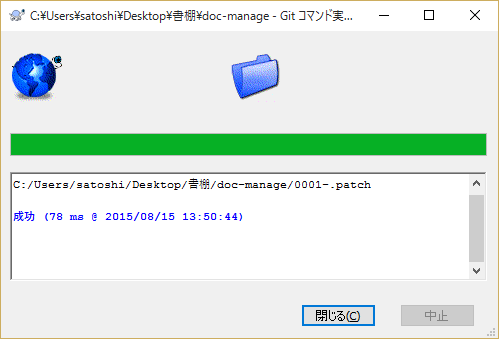


「パッチを作成」の画面が表示されます。

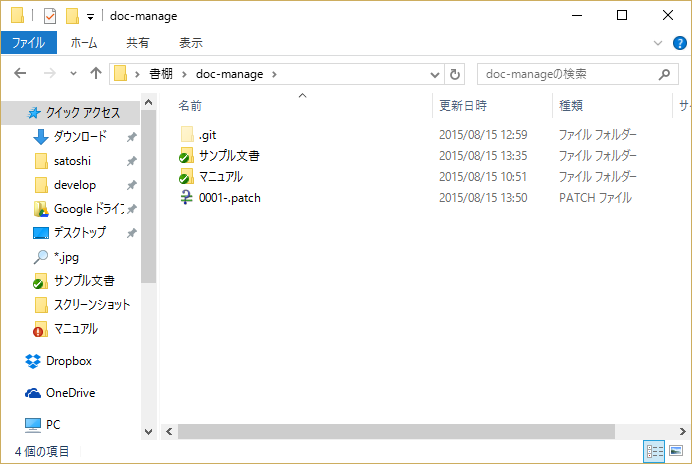
「バージョン」欄は「このコミットから」にチェックを付け「remorts/origine/HEAD」を選択して「OK」をクリックしてください。



下記のように「成功」と表示されれば変更内容のまとめファイルの作成は終了です。



自分の書棚フォルダーの中を見てみましょう。「0001-.patch」というような末尾に-patchという名前の付いたファイルが作成されています。



このファイルを電子メールに添付して原本の管理者へお送りください。

改変内容は管理者が確認の上妥当であると判断されれば原本に反映されて公開されます。

以上で原本への改変採用依頼の作業は完了です。

原本

リモートリポジトリ

（共有書棚）



Ｂさん

Ａさん

写本

写本

ローカルリポジトリ

（自分専用書棚）

ローカルリポジトリ

（自分専用書棚）

写本を基に

変更したファイル

写本を基に

変更したファイル

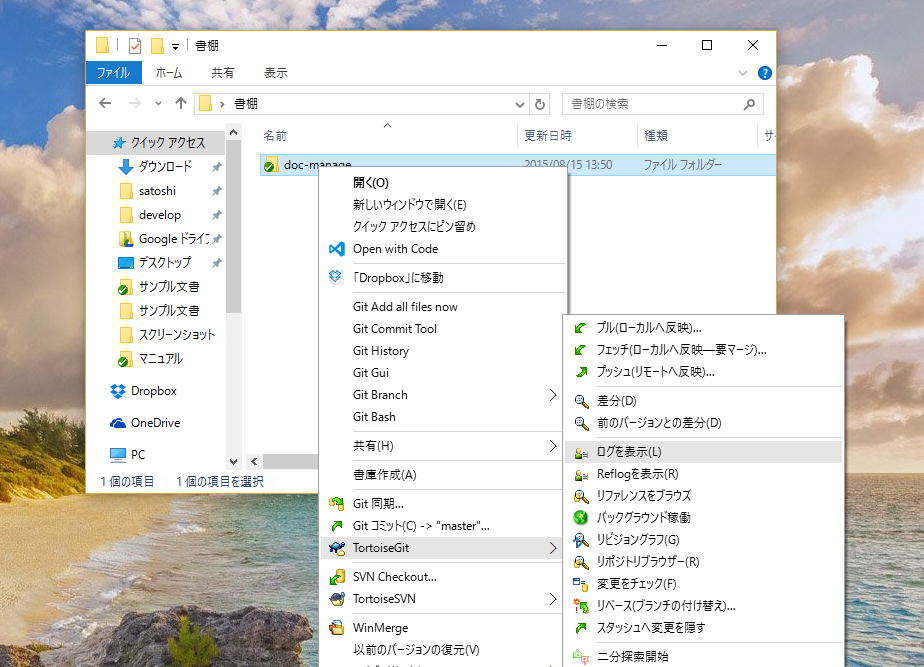


### 変更部分の確認

原本には管理者による更新が随時取り込まれて変更がされてゆきます。  
どのような変更が加わっているのか履歴を確認したいときにはGitの機能を利用して詳細まで確認を行うことができます。

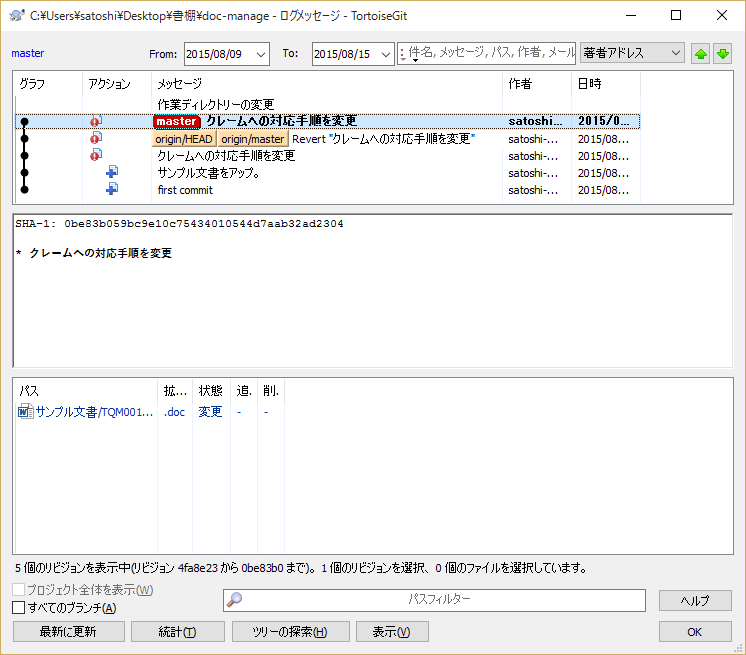
ここでは変更履歴の確認手順を説明します。

自分の書棚フォルダー（デスクトップ\書棚\doc-manage）を右クリックし、「TortoiseGit」→「ログを表示」を選択してください。



「ログメッセージ」画面が表示されます。

この画面の一覧には自分の書棚内に加えられたすべての変更が時系列で表示されています。



1. メッセージ詳細
2. 履歴リスト
3. 変更ファイルリスト
4. 履歴リスト  
   書棚への変更履歴が時系列で現在から過去への順番で表示されます。
5. メッセージ詳細  
   改変内容の詳細メッセージが表示されます。
6. 変更ファイルリスト  
   改変されたファイルのリストが表示されます。

上図の例でとラベル付けされている行は自分の書棚の改変状態を示しています。とラベル付けされている行は原本の改変状態を示しています。  
すなわち現在は原本に未反映の改変が自分の書棚に存在するという状態となっていることがわかります。

では、原本と自分の書棚の違いを確認してみましょう。

の行をクリックして選択し、「変更ファイルリスト」の確認をしたいファイルを右クリックしてください。



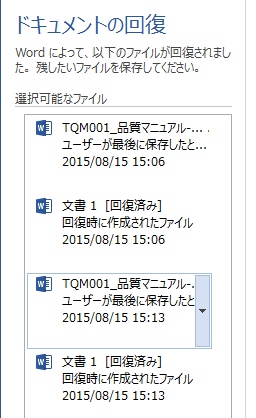
次いでメニューの「作業ベースと比較」を選択します。



するとMicrosoft Wordが自動起動し変更点が表示されます。  
前項で自分の書棚の文書に施した改変事項が正確に表示されていますね。



※環境によっては変更点表示時にMicrosoft Wordが異常終了することがあります。  
その際はMicrosoft Wordをもういちど起動しなおし、復元されたファイルを表示すれば改変内容を確認できます。

### 変更された原本の再取得

原本に管理者自身からの改変や他の改変者からの改変が取り込まれて更新がされていることが分かった場合は原本を再取得して自分の書棚を更新しましょう。

更新

原本

リモートリポジトリ

（共有書棚）



Ｂさん

Ａさん

写本

写本

ローカルリポジトリ

（自分専用書棚）

ローカルリポジトリ

（自分専用書棚）

写本を基に

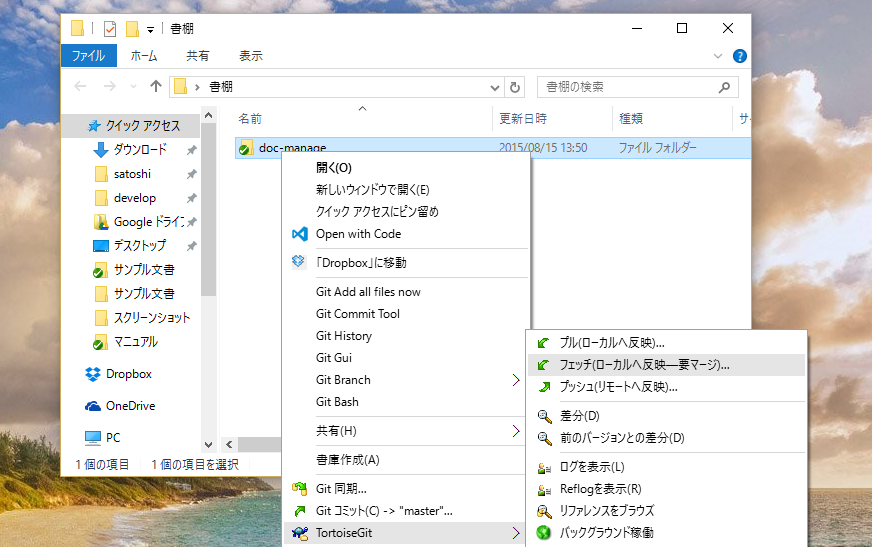
変更したファイル

写本を基に

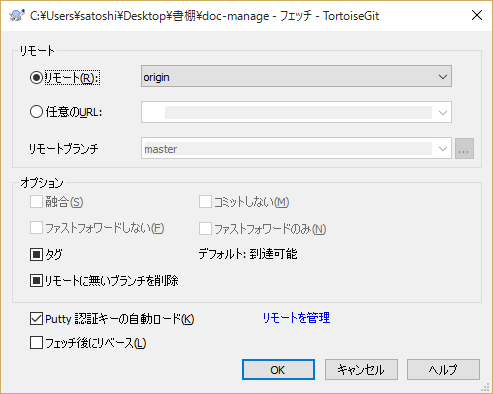
変更したファイル



自分の書棚フォルダー（デスクトップ\書棚\doc-manage）を右クリックし、「TortoiseGit」→「フェッチ～」を選択してください。



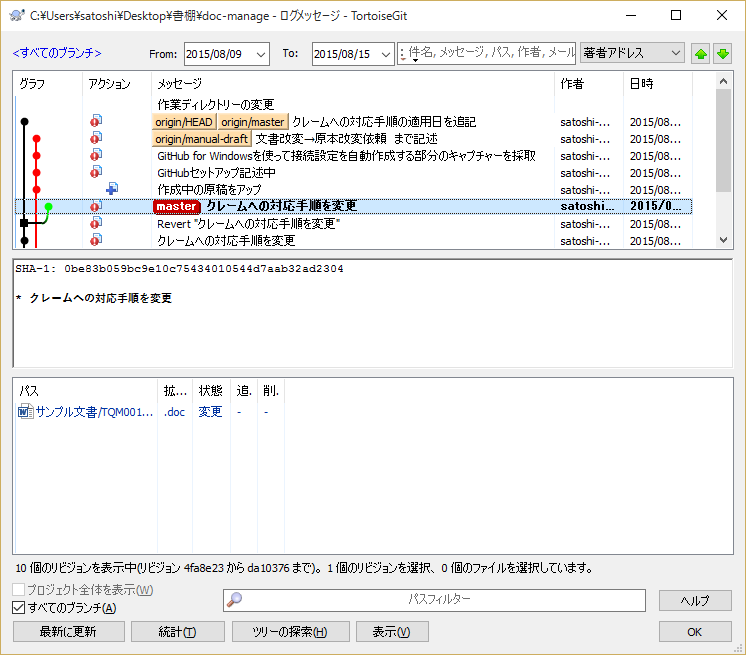
「フェッチ」画面が表示されますので「OK」ボタンをクリックしてください。



下記のように「成功」と表示されれば原本への変更履歴の取り込みは終了です。



次いで、自分の書棚の変更履歴を表示してみましょう。  
自分以外からの改変が原本に加わって、原本（）のバージョンが高くなっていることがわかります。



※もしが表示されない場合は画面左下の「すべてのブランチ」にチェックが入っているかご確認ください。

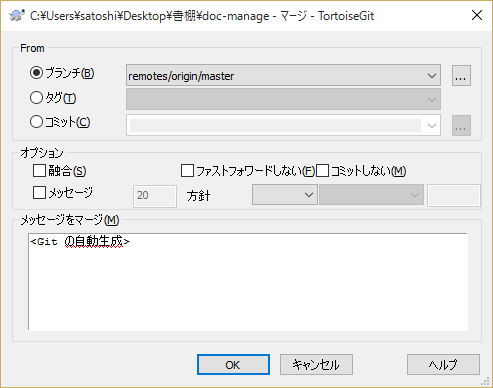
では、最新の原本への更新を取り込んでみましょう。

「ログメッセージ」画面の原本（）の行を選択して右クリックします。

メニューより「masterへマージする…」を選択します。



「マージ」画面が表示されますので「OK」ボタンをクリックします。

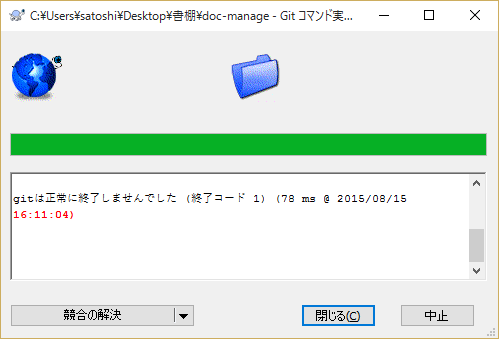


「成功」と表示されれば原本への変更反映は終了ですが、下記のように失敗する場合は少し手順が複雑なものになります。

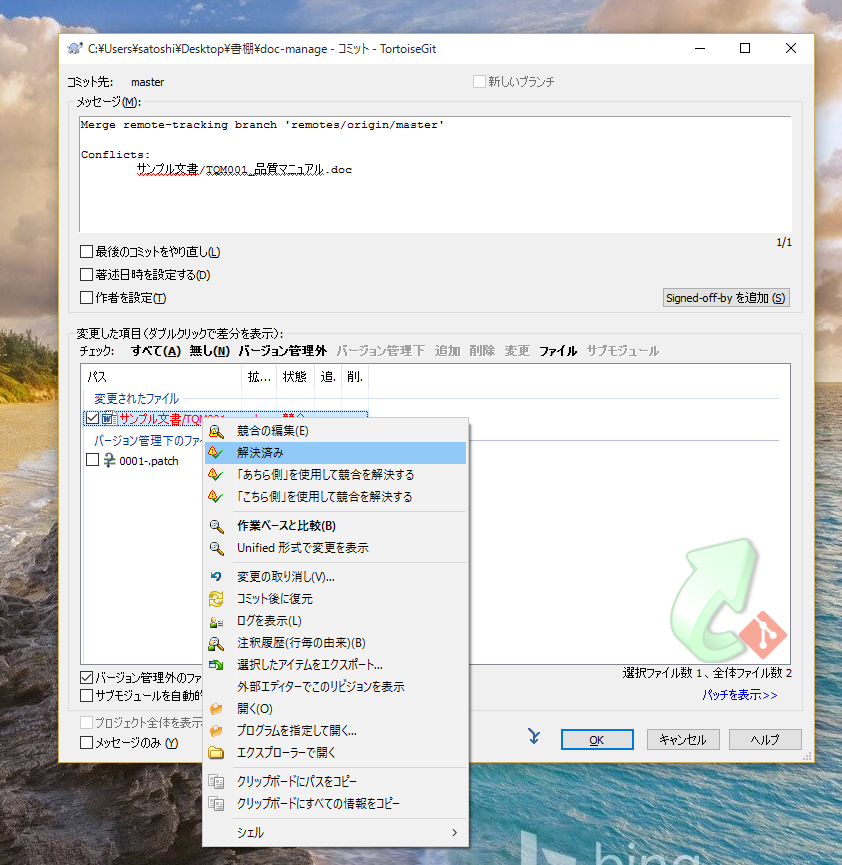
この失敗は複数の方が同じファイルの同じ個所を変更したため併合作業が自動ではできなかったことが原因です。

Microsoft Wordの文書ファイルなどテキスト形式ではないファイル類もこのように失敗することが多々あります。

その場合は「競合の解決」ボタンをクリックしてください。



「コミット」画面が表示されますので競合を起こしているファイルを右クリックします。



競合をどのように解決させるかを選択します。

「あちら側」というのは原本、「こちら側」というのは自分の書庫を表しています。

大抵の場合は「あちら側」すなわち原本側を優先することにし、自分が行った改変をいったんクリアにしたのち、再度前項で説明した手順で自分の改変を文書に施し、変更確定（Gitコミット）を行い、改変採用依頼を原本の管理者へ出しなおす流れになります。

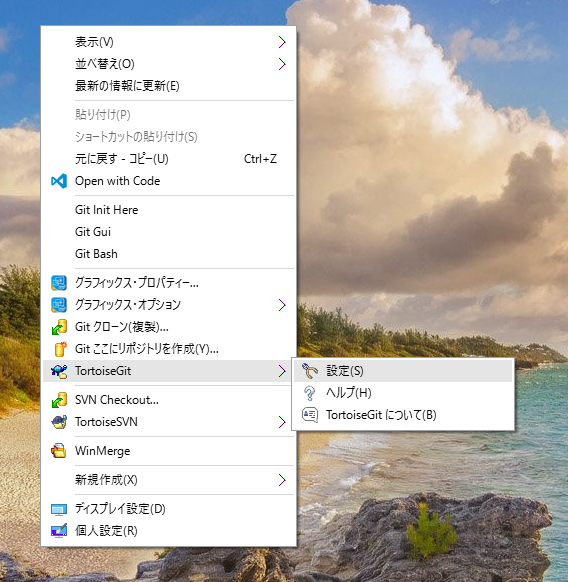
以上で原本の改変履歴の確認方法と、最新版の再取得の方法について説明しました。

## **Microsoft Word形式ファイル以外の差分表示**

TortoiseGitではMicrosoft Word形式ファイルの差分比較表示に対応していますが、設定変更でMicrosoft Excel形式ファイルやPowerPoint形式の差分表示も可能です。

設定は下記のように行います。

デスクトップで右クリックし、「TortoiseGit」→「設定」を選択します。



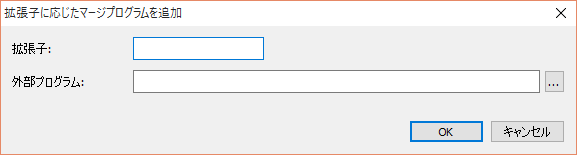
「マージツール」の設定項目で「高度な設定」ボタンをクリックしてください。



「マージの高度な設定」画面の「追加」ボタンをクリックしてファイル拡張子ごとの設定を追加します。



下記画面で設定を指定します。



指定する設定値は下記の通りとなります。

|  |  |
| --- | --- |
| 拡張子 | 外部プログラム |
| .xls | wscript.exe "C:\Program Files\TortoiseGit\Diff-Scripts\merge-xls.js" %merged %theirs %mine %base //E:javascript |
| .xlsx | wscript.exe "C:\Program Files\TortoiseGit\Diff-Scripts\merge-xls.js" %merged %theirs %mine %base //E:javascript |
| .ppt | wscript.exe "C:\Program Files\TortoiseGit\Diff-Scripts\merge-ppt.js" %merged %theirs %mine %base //E:javascript |
| .pptx | wscript.exe "C:\Program Files\TortoiseGit\Diff-Scripts\merge-ppt.js" %merged %theirs %mine %base //E:javascript |

実行結果イメージ

