|  |
| --- |
| NPO法人コミュニティリンク |
| SourceTreeとgitを用いたバージョン管理 |
| ワークショップ資料 |
|  |
| **梶田成祥** |
| **2015/01/28** |

|  |
| --- |
|  |

目次

[2. SourceTreeインストール 1](#_Toc410228102)

[3. Gitの操作 5](#_Toc410228104)

[4. GitHubと連携 11](#_Toc410228105)

[5. GitHubのFork Pull request体験 15](#_Toc410228106)

# SourceTreeインストール

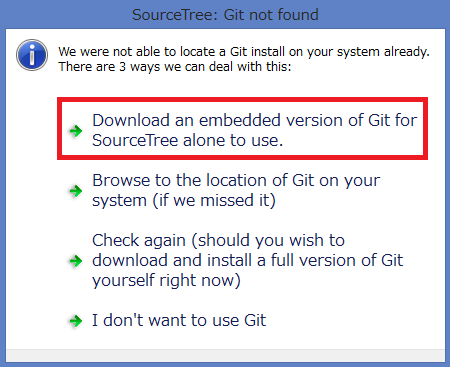
・SourceTreeのサイトにアクセスしてソフトをダウンロード



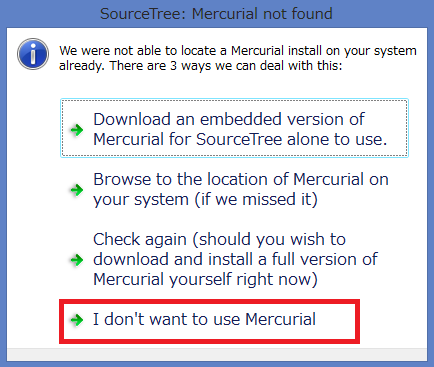
・手順に従ってインストール



・Git内臓版をインストールするので一番上をクリック。



・Mercurialは今回使わないので一番下をクリック

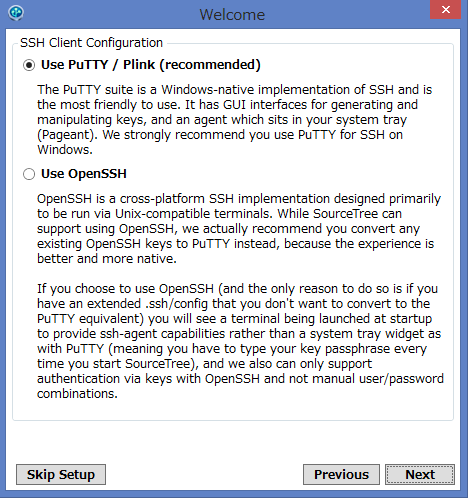


・名前とEメールアドレスの登録

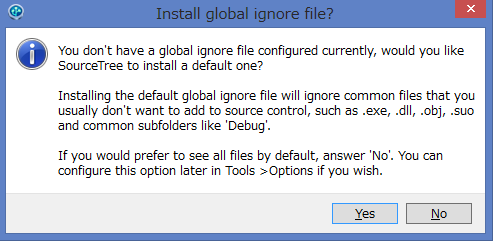
ここで登録した情報がコミット時に使用される



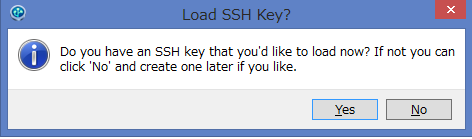
・sshクライアントの設定はデフォルトのPuTTYを使う設定で次へ



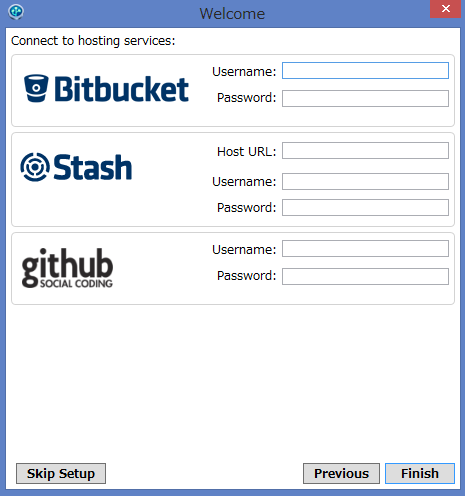
・無視ファイルをインストールするか聞かれるのでそのままYesをクリック



・SSH鍵をロードするか聞かれる。SSH鍵は後で作成するのでNoをクリック



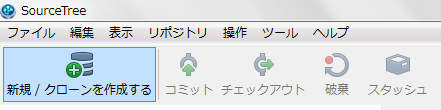
・ホスティングサービスとの連携画面。登録は後でも可能。



# Gitの操作

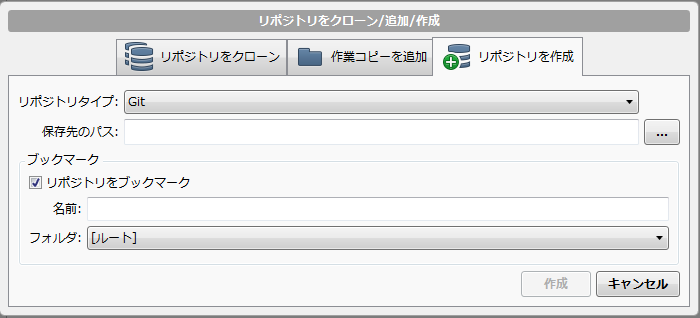
3.1新規リポジトリの作成

画面左上の「新規/クローンを作成する」アイコンをクリック。

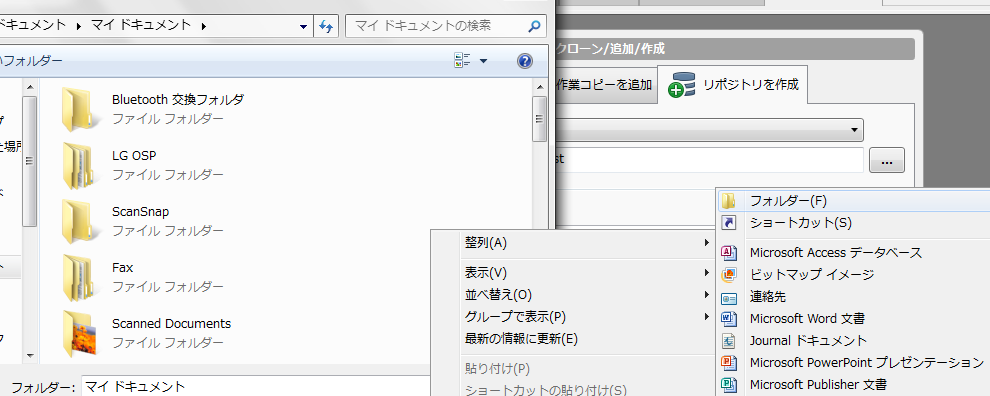


「リポジトリを作成タブ」をクリック

保存先のパスの欄の右の[…]をクリックするとファイルを選択するダイアログが表れる。

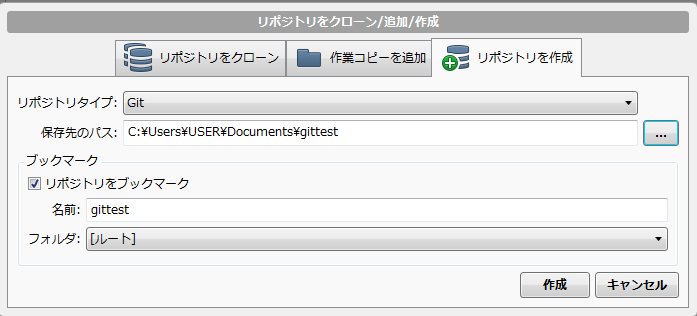


任意の場所を選択し、フォルダを新規作成して選択する。



今回はgittestという名前でフォルダを作成した。

作成をクリックする事でリポジトリが作成される。



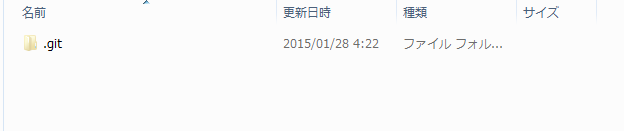
3.2ファイルの追加・コミット

リポジトリ作成後の画面。

まだ何も入っていない状態なので、「エクスプローラで開く」をクリックし、ファイルを追加する。

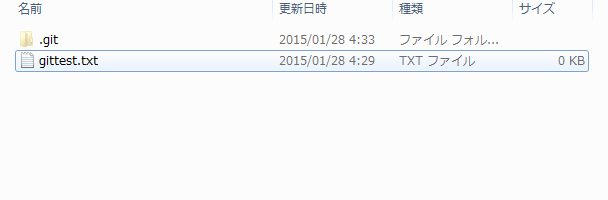


リポジトリ作成後の中身。「.git」フォルダが作成されている。更新内容等のデータはこのフォルダに保存される為、削除しない様に注意。



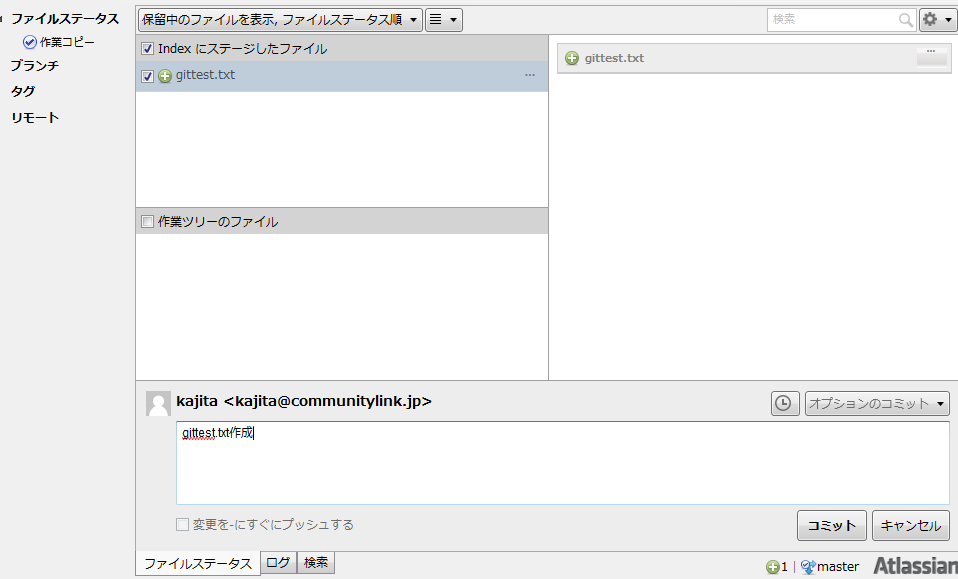
メモ帳などを起動し、適当に編集して保存する。

ひとまず、gittest.txtという名前で作成。

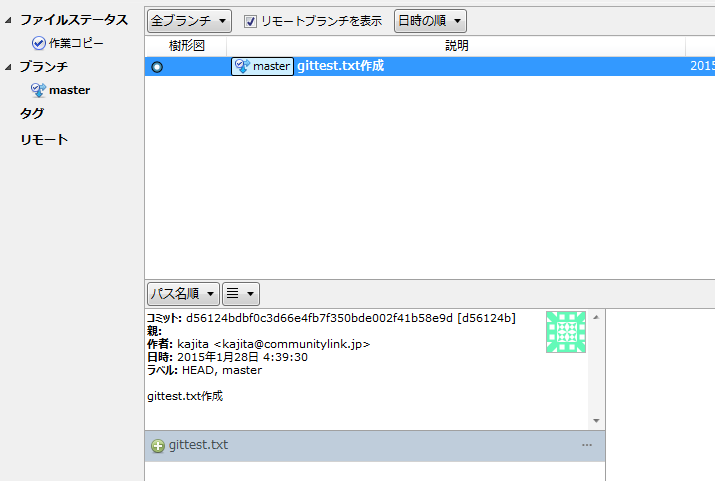


SourceTreeの画面に戻ると、ファイルが追加された事が反映されている。

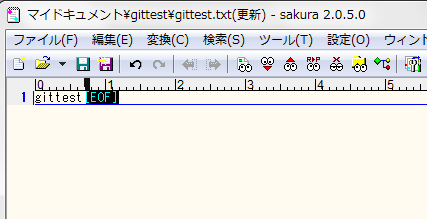
作業ツリーのファイルからgittest.txtにチェックを入れ、コメントを入れてコミットをクリックする。



・初コミット後の画面。masterブランチが作成されている。

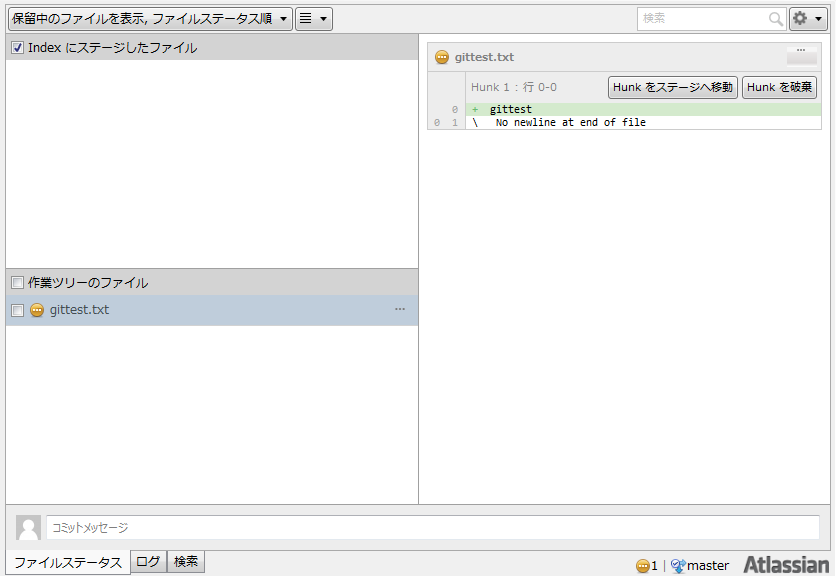


ファイルを適当に編集して保存する。

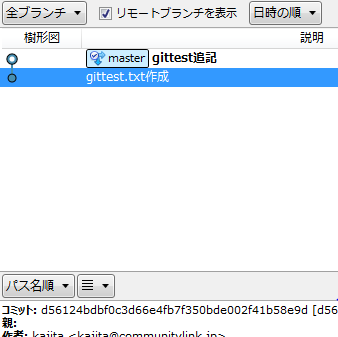


ファイルステータスをクリックし、最初のコミットと同じ要領でコミットを行う。

※右側の画面には変更箇所が表示されている。



編集後コミットした後の画面。コミットの流れはこの様に樹形図で表現される。



3.3ブランチ作成・チェックアウト

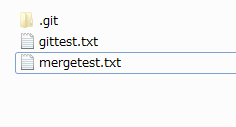
最初のコミットを選択し、ブランチアイコンをクリックする。

developという名前で新規ブランチを作成する。



この時、「新規ブランチを作成してチェックアウト」にチェックを入れていると、自動的にチェックアウトが行われる。チェックアウト後にgittest.txtを開くと、中身が一つ前の内容に戻っている。

続いてmergetest.txtという名前でテキストファイルを作成し、コミットを行う。



別ブランチでコミットを行うと、樹形図はこの様に枝分かれして表示される。



また、

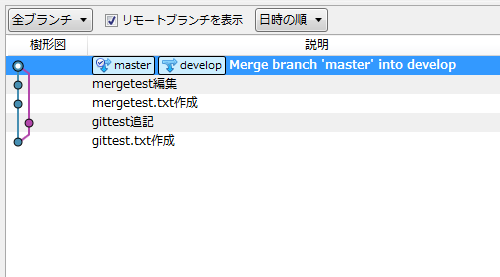
masterブランチを選択し、チェックアウトを選択し、実行するとdevelopブランチで作成したmergetest.txtがmasterブランチには存在しない事が確認できる。

developブランチに戻り、mergetest.txtを編集して内容をコミットする。



3.4マージ

masterブランチを選択し、マージを行う。マージ後、developブランチで行った編集がmasterリポジトリにも反映される事を確認する。



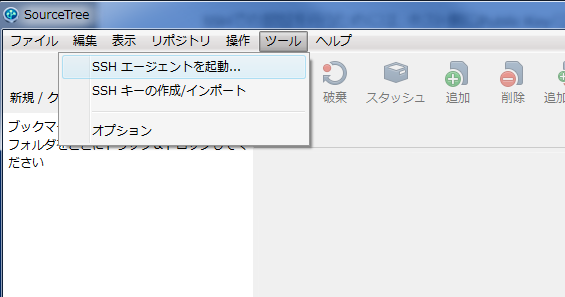
# GitHubと連携

　GitHubと連携する事で、SourceTreeを用いてGitHub上のリポジトリを操作できる。

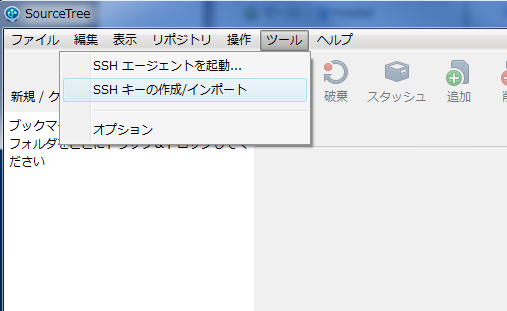
GitHubの認証にはSSHキーが使用されるので、SSHキーの設定を行う。

4.1 SSHエージェントの起動

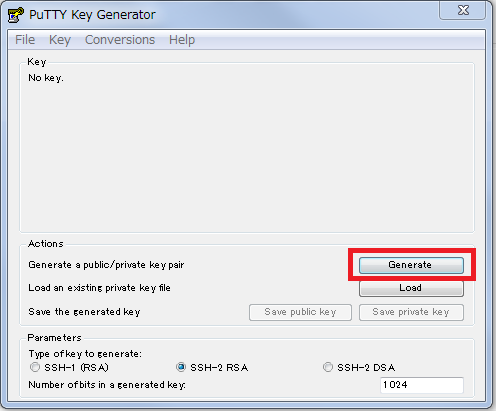
SSHキーの生成の準備の為、SSHエージェントを起動する。



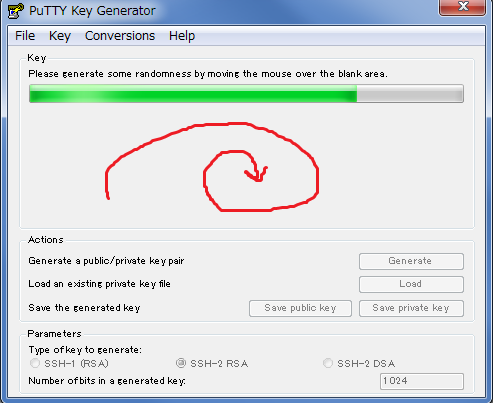
続いて、SSHキーの作成/インポートを選択しSSHキー作成画面を表示する



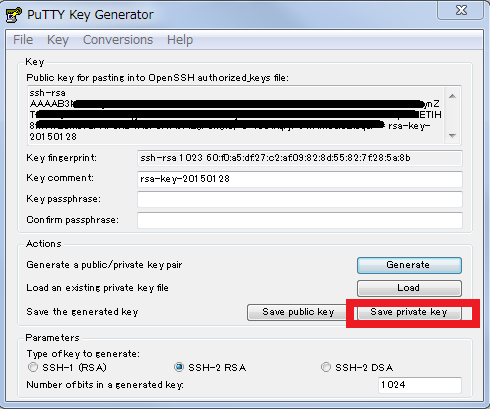
PuTTY Key Generatorが起動する為、Generateをクリック



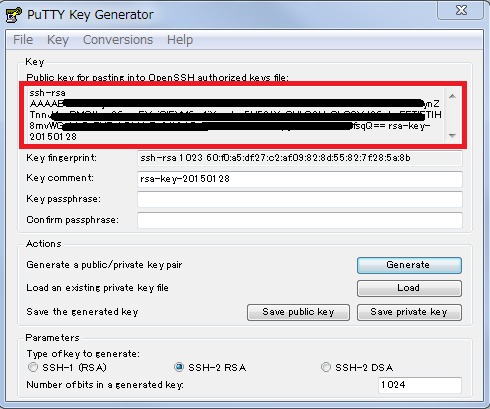
マウスのランダムな動きによってキーを生成する為、緑のバーがいっぱいになるまで、マウスを適当に動かす。



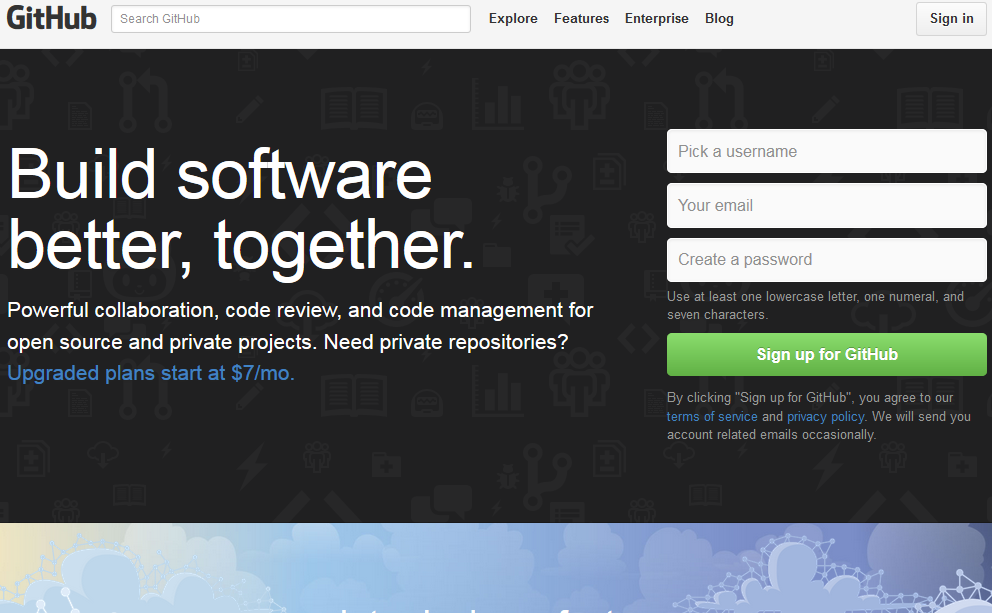
鍵が生成される為、[Save private Key]をクリックして、秘密鍵を任意の場所に保存する。



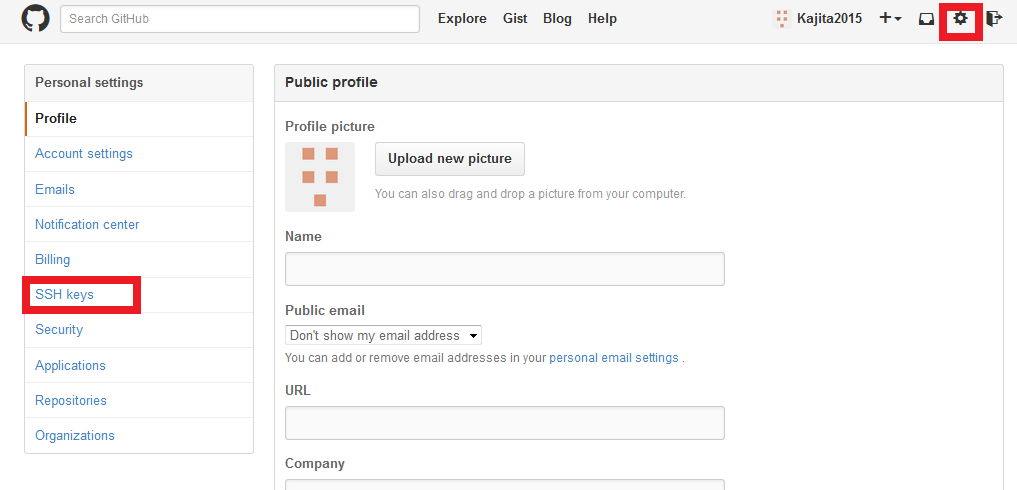
Public keyは赤枠で囲んだ部分の文字をコピーし、メモ帳に張り付けて任意の場所に保存する。



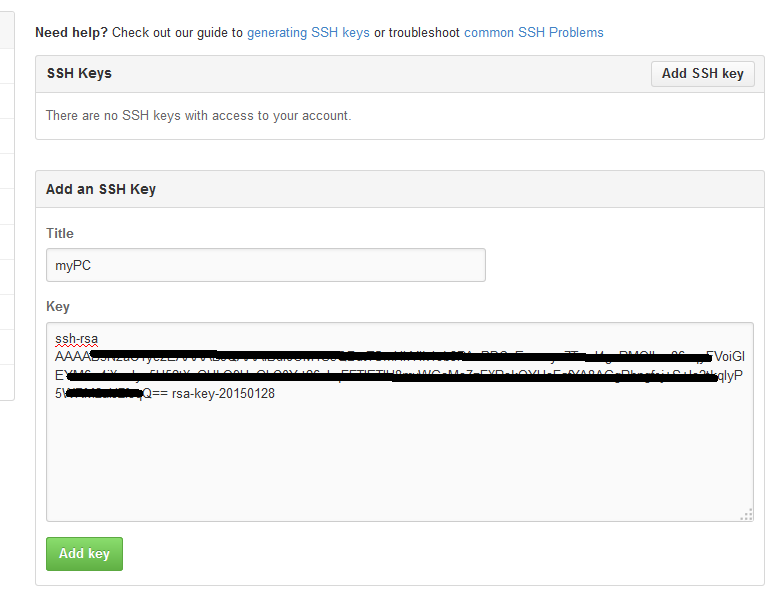
GitHubにアクセスする。ユーザー登録を行っていない場合は必要項目を記入して、登録する。ユーザーネームとパスワードは後で必要になるので記憶しておく事。



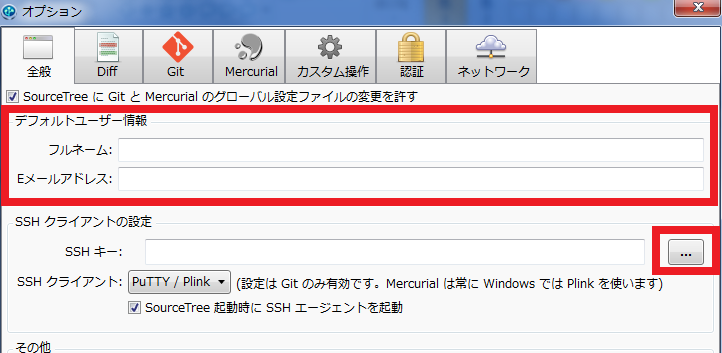
メニュー右上の歯車アイコンをクリックして、現れる左カラムのSSH Keysをクリック



Add SSH keyをクリックし、Titleに適当な名前を入力し、Keyに先ほど保存した公開鍵の内容を張り付けてAdd keyをクリックする。



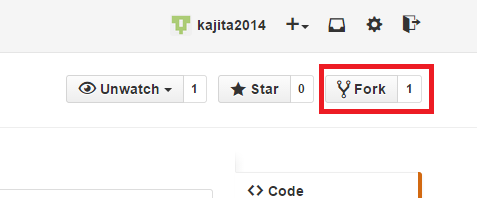
SourceTreeのオプション画面を開き、未入力であれば、デフォルトユーザー情報を入力し、SSHキークライアントの設定入力フォーム横の[…]をクリックし、先ほど保存した秘密鍵を指定する。



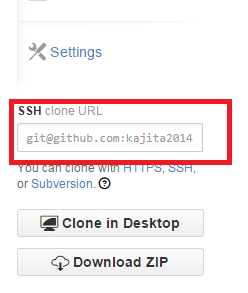
# GitHubのFork Pull request体験

<https://github.com/kajita2014/open_work_shop/>

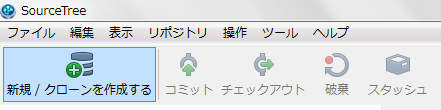
にアクセスし、画面右上のForkをクリック。



画面右下から　SSH clone URLをコピー

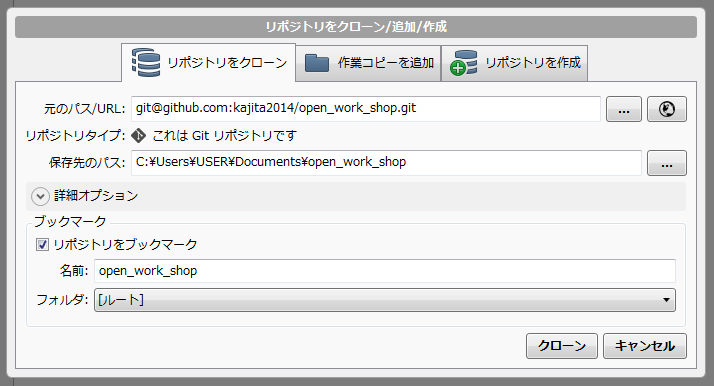


画面左上の「新規/クローンを作成する」をクリックする。

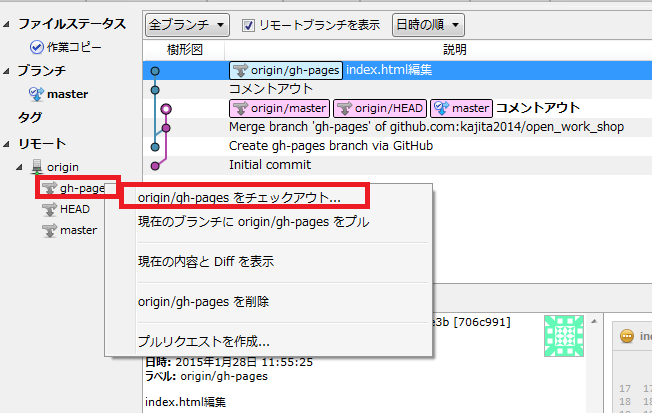


元のパス/URLの入力フォームにコピーしたURLを貼り、クローンボタンをクリックする。

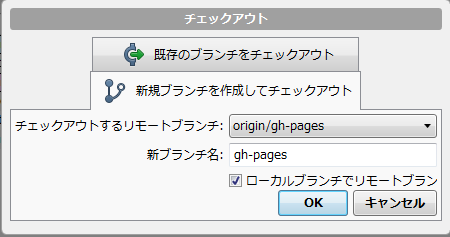
GitHub上のリポジトリがローカル上に複製される。



リモート欄のgh-pagesを右クリックし「origin/gh-pagesをチェックアウト」を選択



新規ブランチを作成してチェックアウトのままOKを押す。



チェックアウトが終わったら、index.htmlを編集する。

今回はプルリクエスト練習として

オープンデータカタログサイトを閲覧してもらい、活用方法について軽く考えて貰った物をプルリクエストで送信して頂きます。

オープンデータカタログサイト

神戸市のオープンデータ一覧

http://www.city.kobe.lg.jp/information/opendata/catalogue.html

ふじのくにオープンデータカタログサイト

http://open-data.pref.shizuoka.jp/

鯖江市オープンデータ一覧

http://www.city.sabae.fukui.jp/pageview.html?id=12768

コメントアウトしている部分が有るので、コピーして編集して下さい。

<!--

<ul type="circle">

<li>ハンドルネーム：</li>

<li>オープンデータリンク：<a href="リンク先のＵＲＬ">オープンデータタイトル</a></li>

<li>活用案：</li>

<li>有ったら良いなと思うオープンデータ：</li>

<li>今回のワークショップの感想：</li>

</ul>

-->

コミット後、リモートのgh-pagesを右クリックしてプルリクエストを作成をクリック。

