



基本設定





GitとGitHubの 概要

Git 歴史



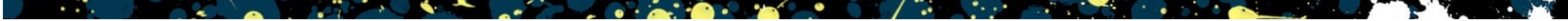
分散型バージョン管理システム

2005年に誕生

Linux(サーバーなどで使われるOS)の
コード管理のために
リーナス・トーバルズが開発

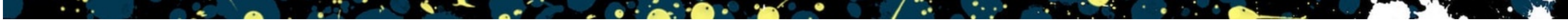
ソフトウェア業界の革命とも言われる

Git 特徴



- ・ 更新履歴の確認
- ・ 保存した(コミット)ポイントに戻せる ・ ・
セーブポイント
- ・ 保存場所の切替(ブランチ) ・ ・ セーブデータ
- ・ (GitHubなども活用する事で)共同作業しやすい(マージ・プルリク))

Git ざっくり年表



2005年 誕生

2019年8月 Git2.23 switchとrestore追加
(checkoutがわかりにくかったので分離)

2021年6月 Git2.32.0

2021年10月 Git2.33.0

2021年11月 Git2.34.0

2022年2月 Git2.35.0

2022年4月 Git2.36.0

GitとGitHub の違い

Git ・ ・ バージョン管理の仕組み、パソコンにソフトをインストール

GitHub ・ ・ Gitを使いweb上で確認する仕組み(webサイト)

Pull Request(プルリク)ができる

->コード差分を確認し問題なければ取り込みできる
(マージ)

他にも機能多数

GitHub ざっくり年表

クリス・ワンストラス、PJハイエット、トム・プレストン・ワーナー
にて開発

2008年 誕生

2010年 GitHub社による運営開始

2013年 GitHubコミュニティ 300万ユーザ

2016年 カンバン機能追加「Project」

2018年 マイクロソフト傘下に

2019年1月 プライベートリポジトリを無料化

2019年11月 GitHub Actions 公開

2021年8月13日 Git操作時のパスワード認証廃止 (SSHなどを使う)



Gitのインストール (mac)

Gitのインストール



Gitのインストール方法は複数

- ・ Gitのサイトからダウンロード
- ・ HomeBrewを使う

Gitのサイト

<https://git-scm.com>

最近のmacは最初からインストールされていますが、バージョンが古い場合もあるので今回はhomebrew経由でダウンロードします。

Gitのインストール(homebrew)

HomeBrewのサイト
(macOSのパッケージ管理ツール)
https://brew.sh/index_ja

```
brew update // homebrewを最新に  
brew -v // homebrewのバージョン確認  
brew install git // gitのインストール  
git --version // バージョン確認  
最新バージョンになっていればok
```


Gitのパスを通す

シェルの確認

(ターミナルに打ちこんだコマンドを実行してくれるプログラム)

2019年 macOS Catalina以降

デフォルトシェルがbash->zshに変更

ターミナルで使用中のシェルを確認

echo \$SHELL

Gitのパス

ターミナルにて

```
sudo vi ~/.zshrc // zshの場合
```

```
sudo vi ~/.bash_profile // bashの場合
```

(viエディタは 閲覧モードと挿入モードがある
Escで閲覧モード、aやiで挿入モード
閲覧モード時に :wq で保存終了)

パスの設定

```
export PATH=/usr/local/bin/git:$PATH
```

```
exec $SHELL -l // シェルの再起動
```

```
git --version // git バージョン確認
```

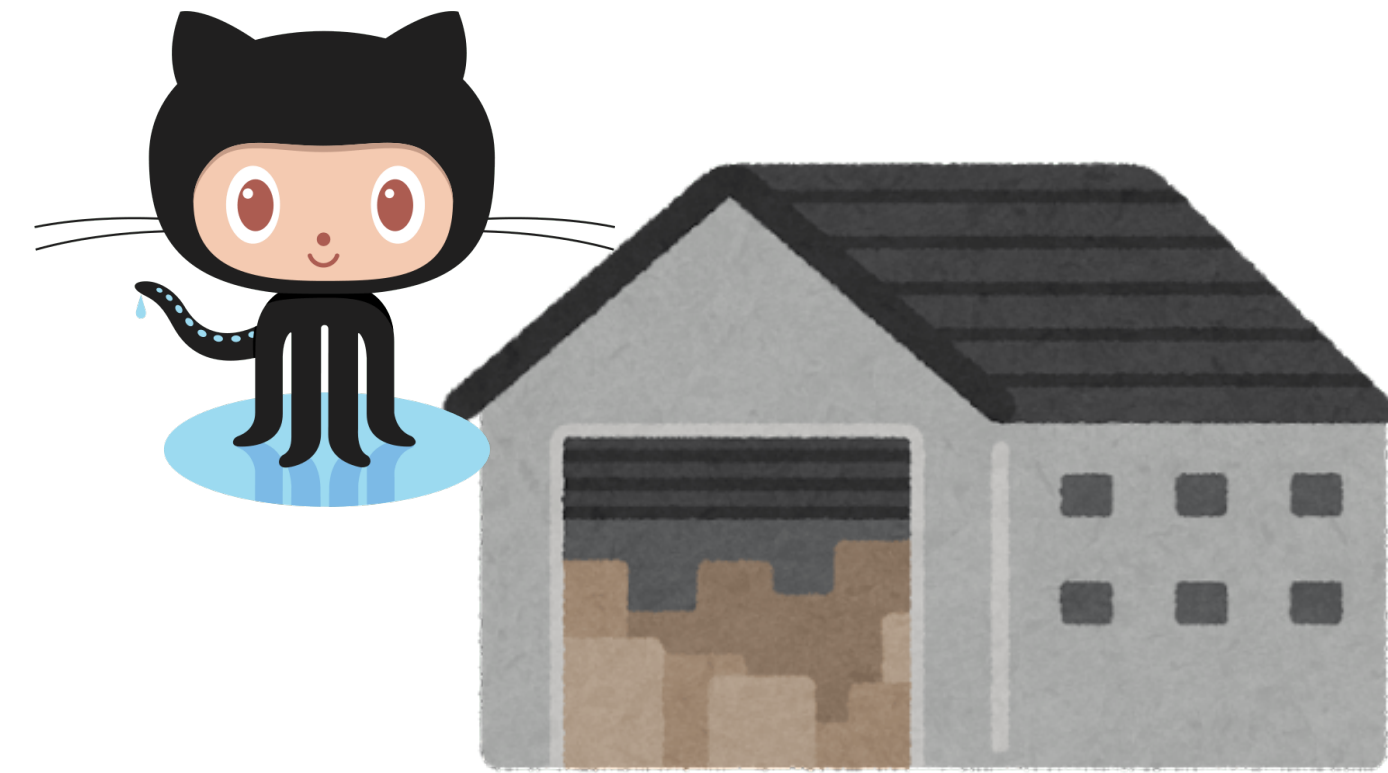



GitHub

アカウント登録

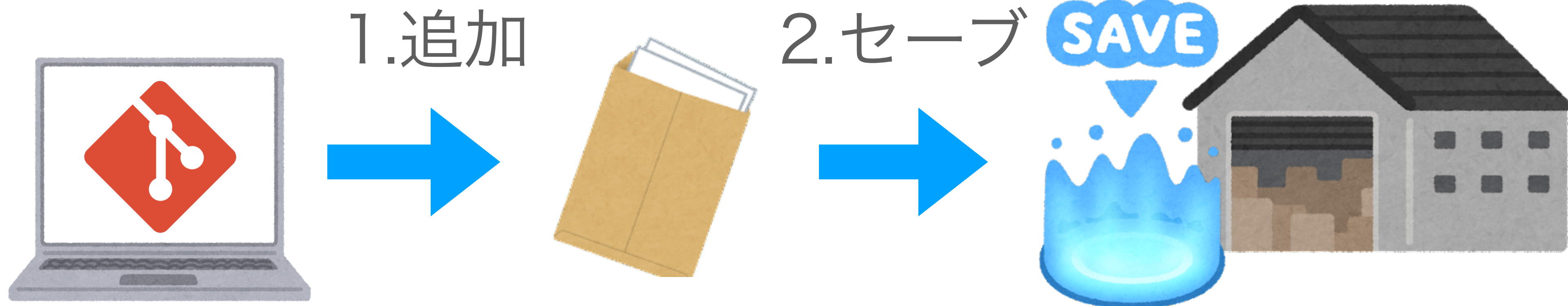
Git/GitHub ざっくり図

Remote



3. アップロード

Local



GitHub登録



GitHub

<https://github.com/>

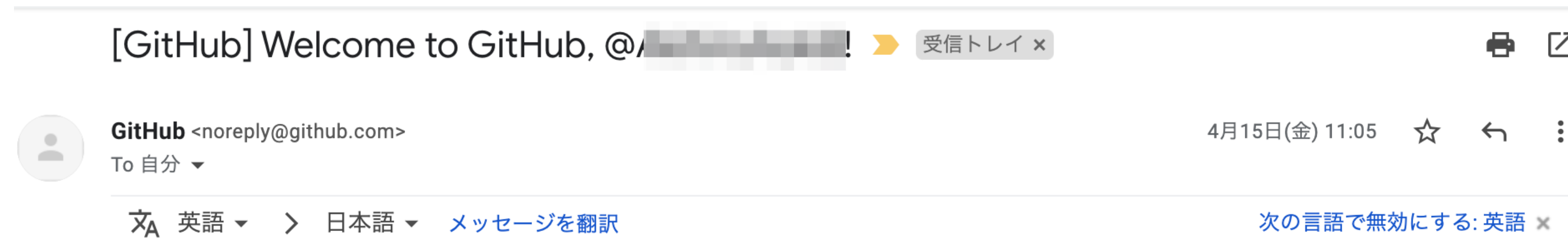
GitHub (日本語)

<https://github.co.jp/>

サインアップから情報入力
ユーザー名、メールアドレス、パスワード

GitHub

登録後こんなメールが届けばOKです



Welcome to GitHub, @[redacted]!



Welcome to GitHub

You're the newest member in this community of over 73 million people who use GitHub to host and review code, manage projects, and build software.



Git/GitHub 初期設定 (mac)

Git 初期設定 1 ユーザー情報

ユーザー名、パスワードの設定
ターミナルで実行

```
git config --global user.name 'xxx'
```

```
git config --global user.email 'xxx'
```

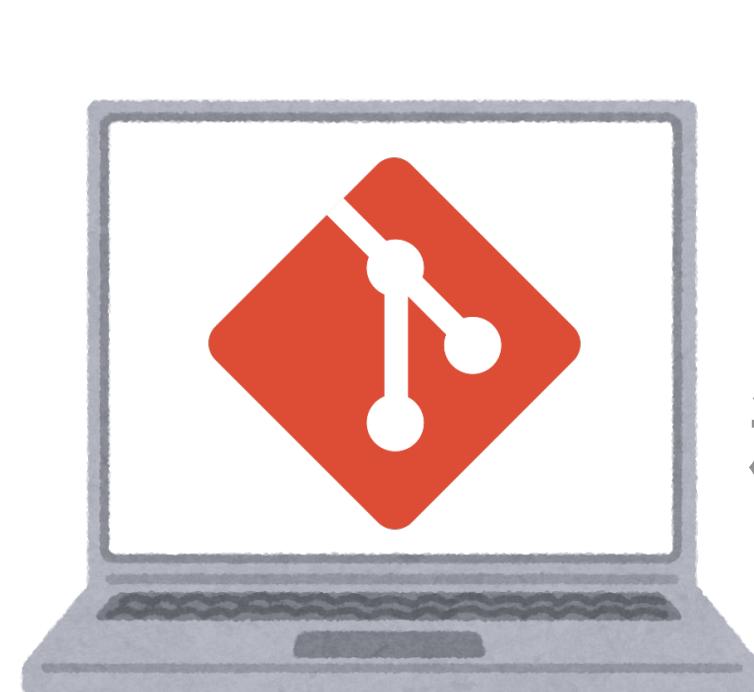
```
git config --list // 設定を見る
```


Git 初期設定 2 鍵のペア

SSH接続するために鍵のペアをつくる



公開鍵 (public key)



秘密鍵 (secret/private key)

Git 初期設定 2 鍵をつくる準備

ターミナル

`cd ~` // ホームフォルダに移動

`mkdir .ssh` // フォルダ作る

`ls -a` // .ssh フォルダができている事を確認

`cd .ssh` // .ssh フォルダに移動

Git 初期設定 2 鍵をつくる

```
ssh-keygen -t ed25519 -C "メールアドレス"
```

または

```
ssh-keygen -t rsa -b 4096 -C "メールアドレス"
```

鍵名はEnter

パスワード入力してEnter (パスワードは空でもok)

(ed25519やrsaというのは鍵の作り方)

Git 初期設定 2 鍵をつくる

ホームフォルダ/.ssh の中に
id_ed25519 // 秘密鍵
id_ed25519.pub // 公開鍵
が生成されていればOK

RSAの場合は
id_rsa // 秘密鍵
id_rsa.pub // 公開鍵

GitHubに公開鍵を設置する

GitHubの右上のアイコンから
Settings->SSH and GPG keys -> New SSH key

The screenshot illustrates the steps to add a new SSH key to a GitHub account. It shows the GitHub homepage with the user 'aokitashipro' signed in. A red arrow labeled '1' points to the profile icon in the top right corner. A second red arrow labeled '2' points to the 'Settings' option in the dropdown menu that appears. A third red arrow labeled '3' points to the 'New SSH key' button in the 'SSH keys' section of the settings page. The 'SSH keys' section lists two existing keys: 'id_git_rsa' and 'SourceTreePublicKey', each with a 'Delete' button.

GitHubに公開鍵を設置する

鍵の名前(自由)、公開鍵をペーストし、
Add SSH key をクリック

SSH keys / Add new

Title

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Add SSH key



GitHub 初期設定 疎通確認

疎通確認

`ssh -T git@github.com` と入力

最初は接続するかと聞かれるので `yes` と入力
接続okになることを確認する



Git/GitHub

初期設定 (windows)

Git 初期設定(windows)

Gitをインストール時に
GitBash というツールがインストールされるので、

GitBashを起動
あとのコードはmacと同様です

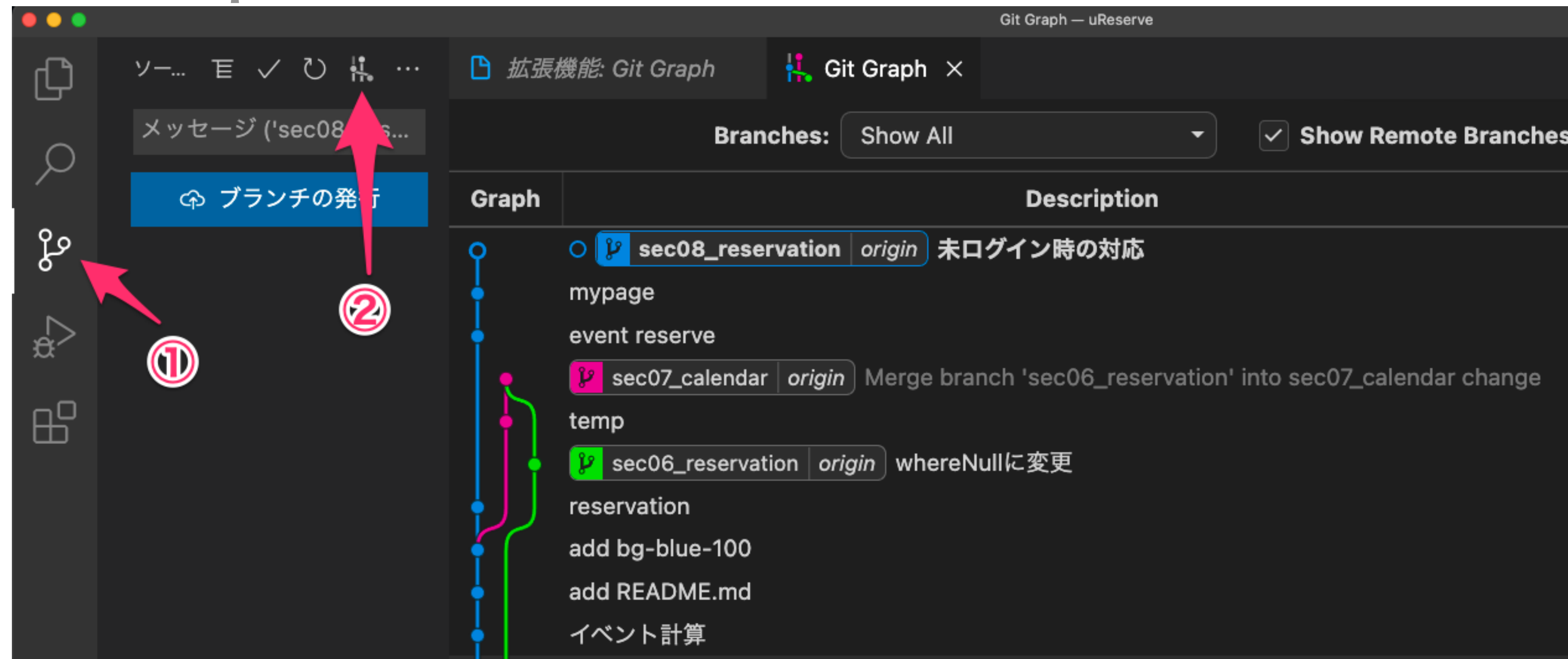
1. 鍵つくる
2. GitHubに公開鍵を保存する
3. 疎通確認



VSCode拡張機能

VSCodeの拡張機能

Git Graph コミット履歴を見やすく表示



表示->コマンドパレットで gitlog と入力でもOK
(他にも GitLens, GitHistoryなどの拡張機能もある)

VSCodeの拡張機能



表示->コマンドパレット

gitlog と入力しても表示されます

VSCode他の拡張機能

Dracula Theme エディタの色

Material Icon Theme アイコン