

ツール・アプリ

2019.12.27

【初心者向け】Gitとは何なのか。基本用語やその仕組みをまとめています。



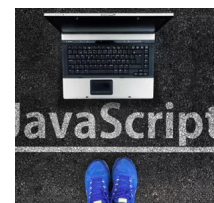
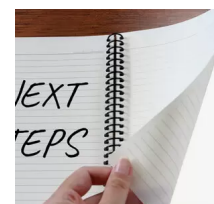
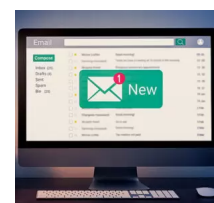
便利なツールのアップデートが頻繁に行われているWEB業界において、Gitを活用して業務を効率化させることは一般的になってきています。しかし、初心者の方や業界に馴染みのない方は、Gitが何のことかわからない方もきっといらっしゃるでしょう。多くの開発者に利用されている印象のあるGitです



最新記事

オススメ

人気記事

【Ja
の
応
2021【20
年
版】
2021長い
入
2021【超
簡
単】
2021

が、実はエンジニアだけでなく、WEBデザイナーやライターにとっても便利なツールのひとつなのです。

Gitを活用することで、チームでの作業効率を高めたり、多くのファイルを管理することが簡単になります。というわけで今回は、その使い方の前に「Git」にまつわる基本的な用語やその仕様について細かく解説いたします。

そもそもGitとは何なのか。

Gitとは、分散型バージョン管理システムです。といっても分かりにくいのですが、ざっくりいうとファイルのバージョン管理が簡単にできるツールといえます。バージョンとはそのファイルをアップデートしたり更新した時に変化するアレですね。iPhoneなんかでも最新バージョンにアップデートするように通知が来たりしますよね。ただ、アップデートしてしまうと基本的には、古いバージョンに戻すことはできませんし、するとしても手間がかかります。

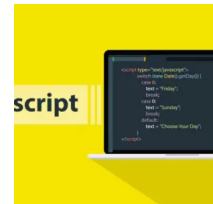
しかし、Gitで管理しているファイルであれば、コンピュータ上でファイルの編集履歴を管理できるので、編集前のファイルを残したまま、新しく編集したファイルを保存することができます。なので古いバージョンから新しいバージョンまでの管理が簡単です。

ア
フ
イ

2021

ペ
ー
ジ

2021

【Ja
の
応

2021

RANKING

今月の売れ筋ランキング

1位

SOLARIS
(tcd088)

2位

CURE (tcd082)

3位

DROP (tcd087)

4位

NANO (tcd065)

5位

HEAL (tcd077)

6位

Switch (tcd063)

7位

FALCON
(tcd089)

CATEGORY

カテゴリー

WordPress – 投
稿・ページ

プログラマーにとっては、多くのコードを編集した上で何か不具合が起きたときに、元のバージョンへ戻すことは日常茶飯事です。かといって、ひとつひとつコードの編集の度に古いバージョンの日付や時刻ををつけて保存して、また新しい作業をするようなことをしては、時間はかかりますし、人的ミスも増えることは、いうまでもありません。そういった作業を無駄なく、効率的に行うためのツールがGitです。

Gitでできること

Gitが 分散型バージョン管理システムとよばれる所以は、以下のような特徴があるからです。

- 。古いバージョンに簡単に戻せる
- 。新旧のファイルを一元管理できる
- 。編集した履歴を複数人で共有できる
- 。複数人で修正した部分を一つに統合できる

以上の特徴からも、多くの開発者に利用されているツールだということはお分かりいただけましたでしょうか。複数人で開発を行ったり、デバッグ作業を行うときにミスを減らし効率化できる仕組みのことですね。

WEBデザイナーの作業にも活用できる

WordPress – 管理画面	>
WordPress – セキュリティ	>
WordPress – バックアップ	>
WordPress – 大集合	>
WordPress – 運用	>
TCD	>
SEO	>
WEB広告	>
ライティング	>
ブログ	>
メルマガ	>
アフィリエイト	>
SNS	>
YouTube	>
アクセス解析	>
プログラミング	>
Webデザイン	>
ツール・アプリ	へ

Gitは、プログラマーがコードを編集する際だけでなく、WEBデザイナーの方にも活用していただけるシステムになります。というのもWEBデザイナーやコーダーが作成するHTMLやCSSも決して一筆書きで制作するものではありません。何度も試行錯誤を重ねて修正したり、元に戻したりして完成に近づきます。

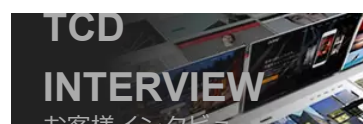
このような業務の特徴からもWEBデザイナーにとっても効率的に活用できるシステムといえます。

WEBデザイナーだけでなく、ライターや、その他の作業にも活用できる

またGitは、コードだけを管理できるわけではありません。テキストデータや画像データ、Excelファイルなども管理することがので、WEBライターに限らず、多くのファイルを編集したり、修正する必要のあるありとあらゆる事務作業にも活用できるシステムなのです。

特に編集したファイルを共有できる機能は、複数人での作業を効率的に管理するのにピッタリですね。

Gitの仕様について

[スピノフ](#)[マーケティング](#)[月を選択](#)

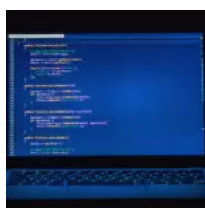
カスタマイズ事例満載で、
初心者にも安心
TCDLABO



ここまででGitとは何なのか。何ができるツールなのかはご理解いただけたと思います。この章ではGitの仕様について解説していきます。まず前提として、GitはCUIツールです。

CUI（キャラクタベースユーザインタフェース）ツールとは、キーボードで入力するコマンドによって操作するツールです。マウスの動きやクリックなどで直感的に操作できるGUI（グラフィカルユーザインタフェース）ツールの対義語として使われます。

それぞれインターフェースのより具体的な違いについては当記事では省略しますが、例えば、プログラマーは、文字だけの黒い背景の画面で仕事をしているイメージをお持ちかと思います。まさにあのような画面のことをCUI仕様の画面と認めていただければ大丈夫です。CUIについては、下記記事でも解説しています。



2019.12.25

**【初心者向け】CUIとGUIの違いとその特徴。
なぜ使い分ける必要があるのか。**

IT・WEB業界で働いている方や、プログラミングを勉強中の方は、CUIというワードを聞いたことがあるかと思います。ざっく

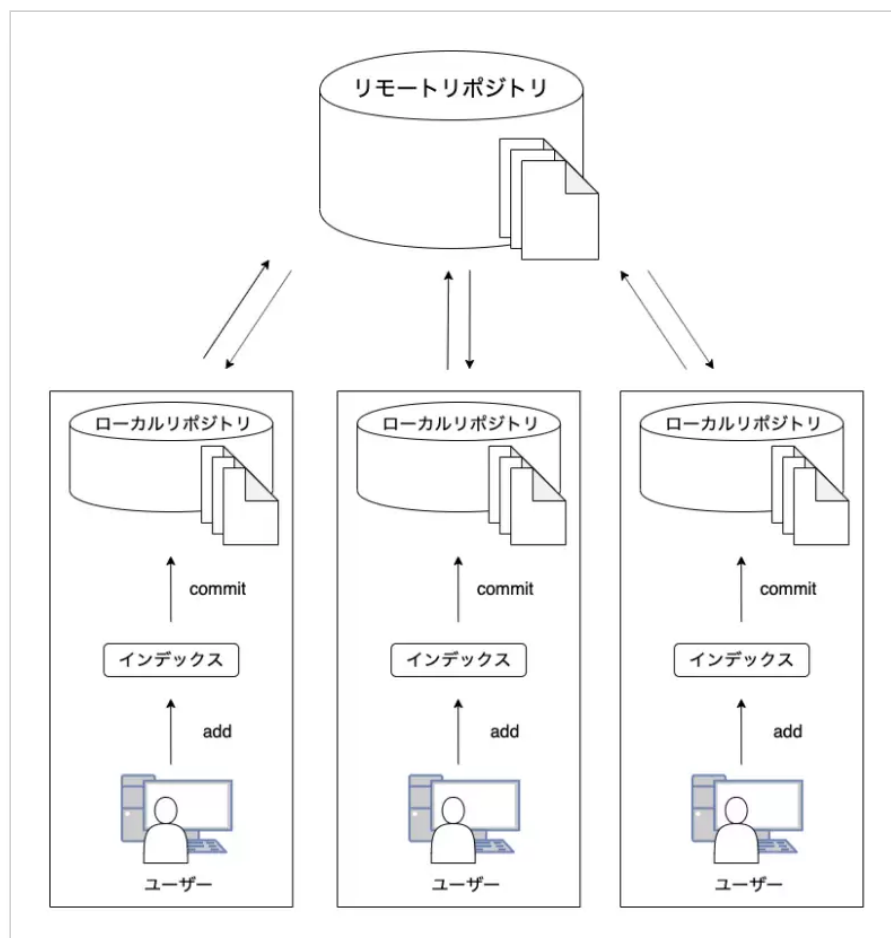
前述の通り、数多くのファイルやデータを複数人で共有するためのシステムであるGitでは、どの部分をいつ誰がどのように編集したのかを的確に共有する必要があります。ですので、もし仮にGitがGUIツールであれば「一番右端のボタンを操作したときの動きを〇〇のように編集しました」というよ

うな、人によって受け取り方が異なったり、主観が入る表現では同じ画面で確認するまでどの箇所をどう変更したのかを瞬時に把握することはできません。

ところがCUIツールであれば、何も考えずに共有されたコマンドを入力するだけで、個人の主観にかかわらずまったく同じ事象が再現されるので、複数人で同じ事象を共有するという面でもっとも適しているわけです。こういった理由からGitはCUI仕様になっているのですね。

Gitを理解するための基本用語

Git管理はは基本的に下記のような構造になっています。



Gitを使い始める前に上記の図を参考にして最低限おさえておきたい用語を解説いたします。

リポジトリ (repository)

リポジトリとは、ファイルやディレクトリを入れて保存しておく貯蔵庫のことです。Gitにおけるリポジトリは以下の2種類に分かれています。

- 。 リモートリポジトリ:特定のサーバー上に設置して複数人で共有するためのリポジトリです。
- 。 ローカルリポジトリ : ユーザーごとに配置される手元のマシンで編集できるリポジトリです。

2種類のリポジトリに分けることで、普段の作業はそれぞれのユーザーが手元のローカルリポジトリで行い、作業内容を共有するときにリモートリポジトリで公開するという使い方になります。リモートリポジトリを介して他のユーザーの作業内容を把握することも可能です。上の図の一番上の部分ですね。

コミット (commit)

コミットは、ファイルやディレクトリの編集作業をローカルリポジトリに記録するために必要な操作のことです。コミッ



トを実行するとファイルを編集した日時を記録したファイルが生成されます。コミットを実行するごとにファイルが生成され、時系列順にならんで格納されるので、ファイルを編集した履歴やその内容を確認することができるわけです。

ワークツリーとインデックス

ユーザーが編集している作業中のディレクトリのことをワークツリーといいます。また作業場所であるワークツリーと、保存場所であるローカルリポジトリの間には、インデックスという中間領域が設けられています。

Gitの仕様上、ワークツリーで編集したファイルをコミットしたい場合は、一度インデックスに登録しなければなりません。編集したファイルをリポジトリへコミットする前にインデックスへ登録して仮置き（add）しておくようなイメージです。

ワークツリーからリポジトリに直接保存するとミスが増えますし、効率が良くありません。というのも編集作業を行うファイルは、一つだけとは限りませんし、編集した複数のファイルを一つ一つ確認してコミットするのは、作業的にも非効率ですからね。

コミット予定のファイルをインデックスに仮置きしておけば、後からまとめて確認した上でコミットできるので、編集



したファイルのコミットし忘れなどを防ぐことができますし、余分なファイルを含めずにコミットできるというわけです。

クローン (clone)

一言でいうとダウンロードに近いものと思っていただければ結構です。複数人で共有しているファイル（リモートリポジトリ）をまるごと自分のローカル環境（ローカルリポジトリ）に保存する機能ですので、ほとんどの場合Gitで最初に行う作業になります。

プッシュ (push)

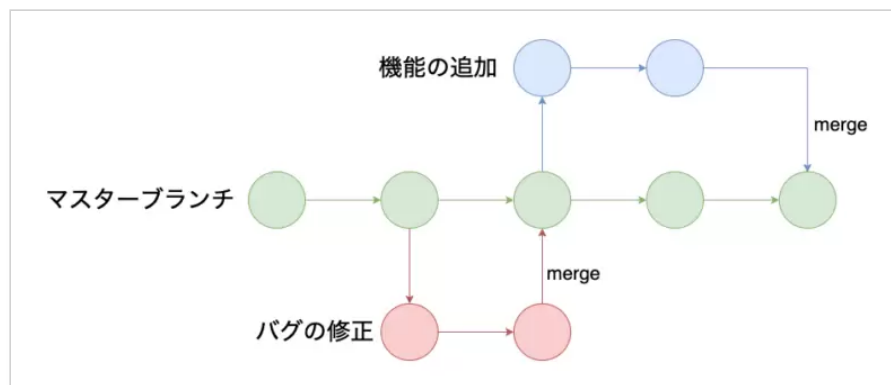
プッシュとは、ローカルリポジトリにあるファイルをリモートリポジトリに送信して保存する機能です。いわゆるアップロードに近い感覚ですね。

誰かが共有しているファイルをクローンして、ワークツリーで作業したファイルをインデックスに一度仮置きし、まとめてローカルリポジトリに登録（コミット）する。ローカルリポジトリにコミットしたファイルを共有するためにリモートリポジトリにプッシュするのが基本的な流れになります。

ブランチ (branch)

ファイルの編集履歴を分岐させて記録していく機能のことです。WEBサービスやソフトウェアの開発において、バグの修正や、機能の追加などのファイル編集作業は複数のユーザーが同時に行うことも少なくありません。並行して同時に行われる作業を正確に管理するためにGitにはブランチという機能が用意されています。これがGitのバージョン管理を効率的にし、間違いを減らすためにもっとも活かされている機能ともいえるでしょう。

マスターブランチと呼ばれるメインのブランチと、そこから分岐してバグの修正や、機能の追加を行っているブランチを例に挙げると下記のようになります。



マージ (merge)

上記で説明した複数のブランチを一つにまとめて、完成形に近づけることをマージと呼びます。バグの修正や、機能の追

加を行ったブランチがマスターブランチに統合されている部分のことですね。

プル (pull)

共有されているリモートリポジトリに保存されているファイルの内、ローカルリポジトリ（あなたのローカル環境）に無いファイルや他のユーザーが更新したファイルのみをダウンロードする機能です。**リモートリポジトリのファイルをすべて丸ごとダウンロードするクローンに対して、ローカルリポジトリとの差分のみをダウンロードして更新するのがプルになります。**

フェッチ (fetch)

リモートリポジトリからファイルの最新情報を取得してくる操作のことです。共有されているファイル（リモートリポジトリ）の更新を確認したり、複数人の作業の擦り合わせのために使う機能といえます。**プルとは違い、ローカルのファイルを更新することはありません。**

リモートリポジトリのファイルをダウンロードし更新（プル）するのか、ファイルの更新があったかどうかを確認する

(フェッチ) という点が異なります。また、**前述のプルは、フェッチとマージを同時に行う機能といえるでしょう。**

何も確認せずにプル（フェッチ+マージ）してしまうと、あなたがローカルで編集したファイルと同じファイルが更新されていた場合、エラーが出たりします。フェッチは、それらを未然に防ぎ、**複数人で同じファイルを編集しているときでもお互い干渉しないようにするための機能**というわけですね。

GitとGithubの違い

初心者の方は、GitとGithubを混同されることが多いかと思いますが、実際は別物を指しています。Gitを活用されている多くの方は、Githubを活用されている印象ですが、以下のような違いがあります。

- Git : 誰がいつどのように編集したかを正確に把握できるバージョン管理システムのこと。
- Github:Gitの仕組みと連携して、他のユーザーとやりとりしやすくしているWEBサービスの名称。

端的に説明するとこのような感じですね。Githubは、文字通りGitのhub（拠点）であり、世界中のユーザーが編集したコードやデザインデータを保存・共有しやすくするためのWEBサービスになります。



Gitは先述の説明の通り、CUI仕様ですので、不慣れな方にとっては使いにくいです。一方でGithubはGUI仕様なので、画面上でマウスを使って操作できたり、複数のユーザーでコミュニケーションをとりやすいように機能が整備されています。このような点がGithubが活用されている要因の一つでしょう。

まとめ

本稿では、Gitを活用するためにおさえておくべき基本用語、その仕組みをご紹介します。Gitの仕組みを理解しておけば、プログラマーやWEBデザイナーだけでなく、事務職の方など、多くのファイルを編集したり、共有する必要のある業務を効率化できることがお分かりいただけたでしょうか。当記事がGitの仕組みや便利さを理解するための手助けになれば嬉しいです。



Tweet



Share



B! Hatena



RSS



TCD編集チーム

TCDテーマを開発する株式会社
デザインプラスのメディア編...



投稿者: TCD編集チーム



ツール・アプリ



git



【初心者向け】
＜ CUIとGUIの違いと
その特徴。なぜ使

不要なファイルが
紛れ込む問題！？ ＞
Macでzip圧縮する



RELATED POSTS 関連記事

Macの調
子が悪い
と感じる

意外と多
い！？見
逃せない

<p>ワードや エクセル の表をデ</p> <p>2020.02.10 </p>	<p>Webサイ トが何の CMSや技</p> <p>2019.07.20 </p>
<p>Instagram をサイト 上にオシ</p> <p>2020.12.28 </p>	<p>QRコード を生成し てPCとス</p> <p>2020.12.20 </p>

カスタマイズ事例満載で、初
心者にも安心

TCDLABO



TCDについて

テーマ一覧

テーマ比較ナビ

お客様の声

よくある質問

利用規約

会社案内

採用情報

メディア掲載

WordPress大全集

SEOまるわかり大全集

特別ライセンスのご案内

特別ライセンス取得者様一覧

TCD公認 Web制作会社

特定商取引法の表示

ライター募集

TCD MUSEUM

TCD GALLERY

TCD LABO

TCD AFFILIATE

フォトマルシェ

ロゴマルシェ

ボタンマルシェ

広告掲載について

