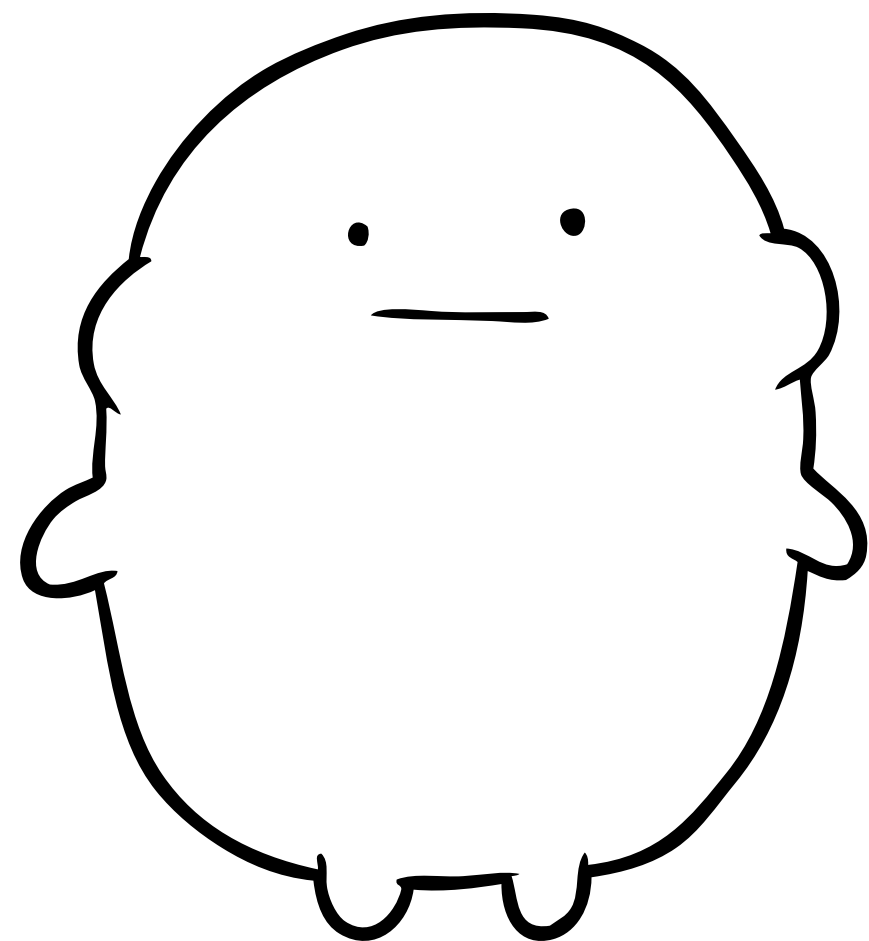




# HTML/CSS II

---

Introduction



# 講師紹介

About

## Profile

名前

紅 尚志(べに たかし)

出身

宮城県仙台市

専門

WEB開発や制作など

趣味

筋トレ、ゲーム(古い)など

メッセージ

初めまして。シンプルなようで意外と奥が深い分野だと思います。目的をもって取り組みましょう！

プログラム  
専門です！

みなさんの話を率直に  
聞かせてください！



学びたいこと、目標など

# 講義のルール

---

共通のルールにしましょう

【この授業のルール】

- ・ 私用以外の休みは素直に申告してください  
課題なら課題で構いません
- ・ 演習中は体をほぐしても大丈夫です
- ・ 自分でテーマ設定する課題は60～70%程度で終わらせられるくらいにしましょう  
高いモチベがあるなら自主製作がおすすめです
- ・ その日の課題が終わらなそうなときも教えてください



# 講義の流れ

---

予習・復習



講義



例題



解説



演習・課題



解説



まとめ

# 教科書

---



Mana, 1冊ですべて身につくHTML & CSSとWebデザイン入門講座,  
SBクリエイティブ(<https://www.sbcr.jp/product/4815618469/>)

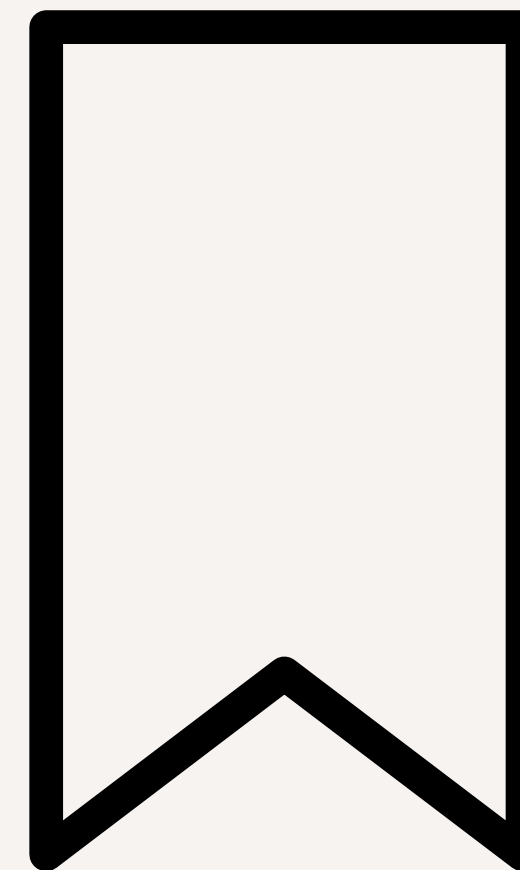


千貫りこ, プロのコーディングが身につくHTML/CSSスキルアップレッスン  
すぐに活かしてずっと役立つ現場のテクニック,  
翔泳社(<https://www.sbcr.jp/product/4815618469/>)

# 参考として



栢木厚，令和05年 イメージ&クレバー方式でよくわかる 栢木先生のITパスポート教室，  
技術評論社(<https://gihyo.jp/book/2022/978-4-297-13128-9>)



最大目的: 納品できる！





# 到達目標

これから1年間をかけて身につけていく技術的な要件です。

正しく仕組みを理解して記述できる

技術を体系的に理解することを目指します。WEBに変更はつきものです。洗練された代替案を出せるようになると、技術がぐっとアップします。

自らの力で解決する能力を身につける

検索ワードで結果が異なることは自明でしょう。どんな時にどこを見るかで、結果が変わってきます。プログラムは楽をするために存在するものです。

前線で役立つスキルを身につける

授業やサンプルを作った後が大切です。「次、何をすればいいの？」と。しっかり目的を持ち、且つすぐに応用できるような技術を習得しましょう。

# 年間の計画を立てよう



4月	5月	6月	7月
<ul style="list-style-type: none"><li>• 前期授業開始</li></ul>		<ul style="list-style-type: none"><li>• LP制作完了(PF可)</li></ul>	
8月	9月	10月	11月
	<ul style="list-style-type: none"><li>• 後期授業開始</li></ul>		
12月	1月	2月	3月
	<ul style="list-style-type: none"><li>• WordPress制作完了</li><li>• 講義終了</li></ul>		



ここまでで  
質問はありますか？

---

遠慮せずに挙手・発言してください！

# 小休止



# 去年の授業はどのような内容でしたか

---

レスポンシブウェブデザインを説明してください

どんなデザインなのか、利点は何かを考えてみましょう。

スキルを教えてください

HTMLマークアップの基本  
(sectionやhタグ、メタなど)を把握していますか。cssのメディア  
クエリや詳細度は説明できます  
か。

不明のままのものはありますか

言語化できそうな疑問があれば教  
えてください。難しければイメー  
ジでも構いません。

# 本日の内容

---

1. ウェブサイトとは何かを振り返ろう
2. Git系ツールをインストールしよう
3. DNSとは何かを振り返ろう
4. FTP・SFTPについて振り返ろう
5. サーバーの役割や設定を確認しよう
6. 開発環境を整えよう
7. まとめ

ウェブ 사이트とは？

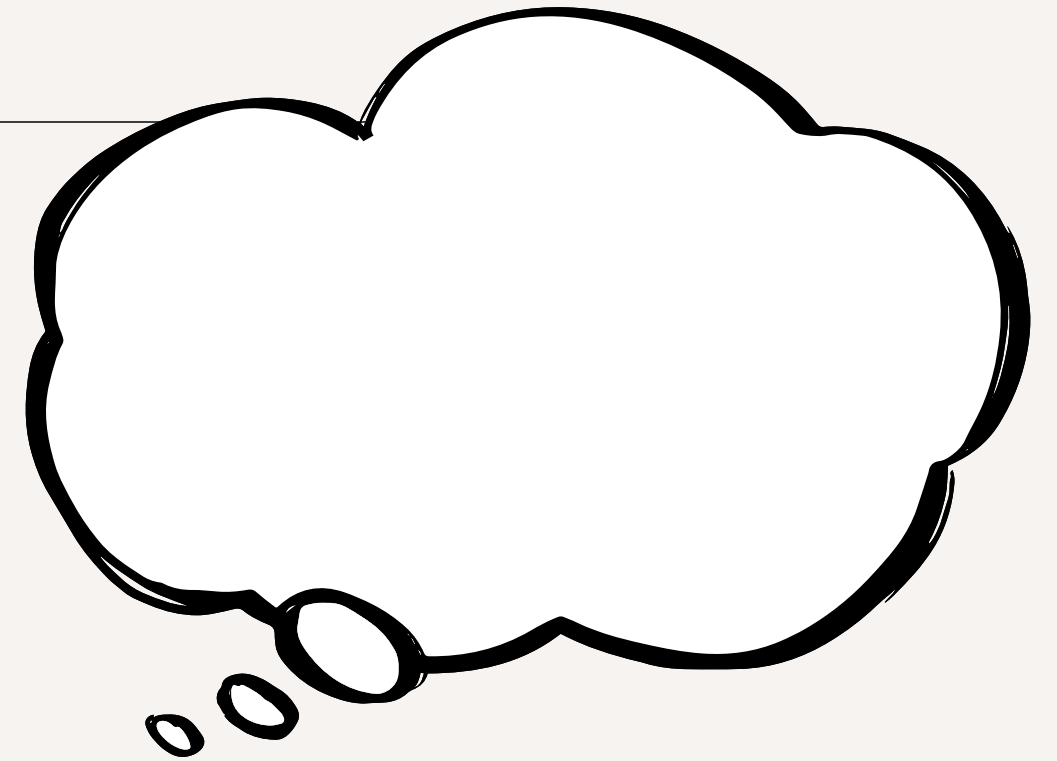




# 良いデザインとは何か

---

改めて考えてみよう！



正解はないが...

## デザインの目的

伝えるための「手段」であり、目的は別にあるということです。

∴ユーザーの利益や体験を最大化しよう。

美しい映像がAmazonで流れたらいいデザインなのか。ゲームサイトなら？

## デザインを具現化するのがコーディング作業

チラシなどとは違い、WEBで公開するためにはコードで書く必要があります。

その方法としてHTMLを記述し、CSSで装飾(デザイン)を組み上げていきます。

## デザインとコーディングは双方向

使いやすさとは重さ(ファイルサイズや実行スピード)や色味、ユーザーの特徴など実感が伴うものです。デザイナーとコーダーは、原則コミュニケーションが必須です。

# Git系ツールを インストールしよう



**git**

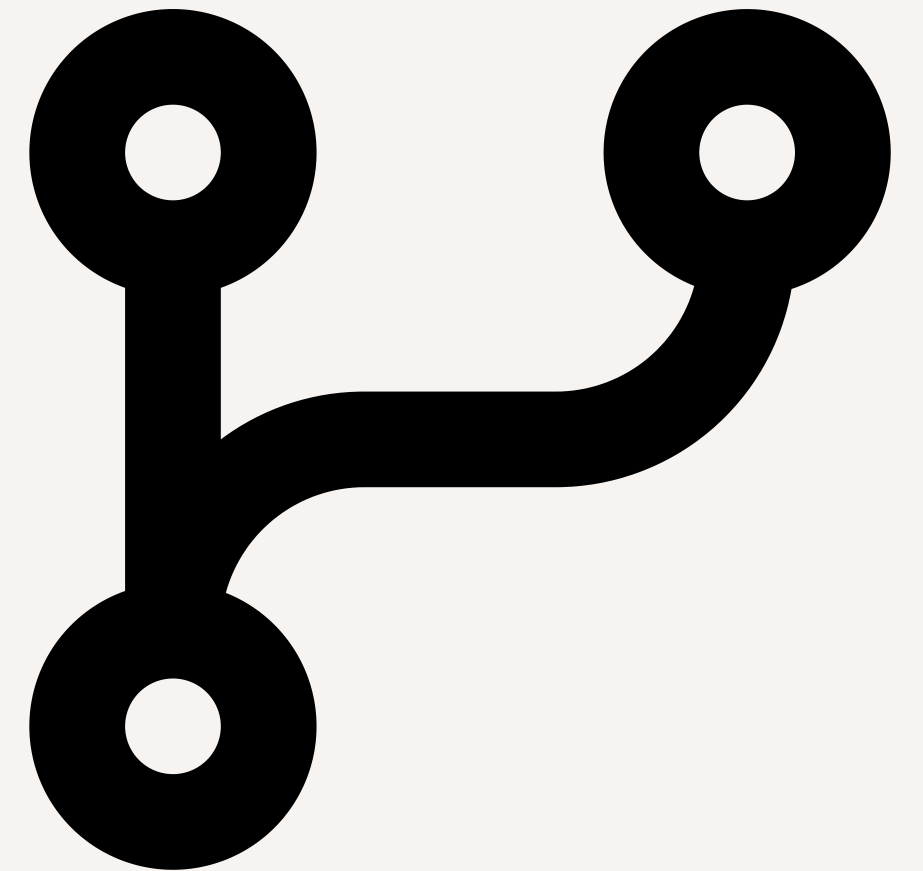
# Gitとは？

---

分散型のバージョン管理システム。各コンピュータにデータを保持しながら、サービスと組み合わせることでクラウドで複数人から参照が可能。

## 【Gitの利点】

- 複数人で作業がしやすい
  - レビューを置けばダブルチェックも可能
- いつ、だれが、何をしたかが一目で分かる
- 過去のデータを保持しており、すぐに戻すことが可能



# GitHubに登録しよう

---

無料でも使えるクラウド型のGit管理サービスです。Git自体は単体でも使えますが、クラウドで保存・参照させるためにはサービス契約が必要です。その筆頭が「GitHub」になります。

<https://github.com/>

# SourceTreeをインストールしよう

---

Gitをグラフィカル(GUI)に使うことのできるアプリケーションです。  
今回は特別な理由がない限り、SourceTreeに内蔵されたGitを使います。

※使う予定はありませんが、インストール時に必要であればBitBucketアカウントを作成します。

<https://www.sourcetreeapp.com/>



ここまでで  
質問はありますか？

---

遠慮せずに挙手・発言してください！



# DNSとは何かを振り返ろう



# DNSとは？

「ドメイン名」とインターネット上の住所にあたる「IPアドレス」を変換するための仕組みです。文字の羅列では人間には難しいため、具体的な名前を付けることで認識しやすくなります。

■ 入力情報	
対象ドメイン または IPアドレス	
google.com	
入力の逆引き または 正引き	
142.250.207.14	

CMAN 株式会社シーマン, 「ドメイン/IPアドレス サーチ  
【whois情報検索】」,  
(<https://www.cman.jp/network/support/ip.html>)

# FTP・SFTPについて 振り返ろう



# FTPとは？

---

ファイルを転送に使われる通信プロトコルです。具体的には、ローカルにあるファイルをサーバーにアップロードする際によく使われます。平文(暗号化されていない)ため取り扱いに注意が必要です。

❗FTP+SSLの暗号化機能を持った「FTPS」を使うようにしましょう。

# SFTPとは？

---

FTPと同じくファイルを転送に使われる通信プロトコルです。「SSH」という仕組みを利用したファイル転送で、FTPとは通信の方法が異なります。

- ✓ SFTPの方が安全性が高いです。
- ✓ SFTPに対応しているのであれば、優先的に使いましょう。
- ✓ ファイルに直接アップロードする方法は、減っていく可能性に考慮しましょう。



ここまでで  
質問はありますか？

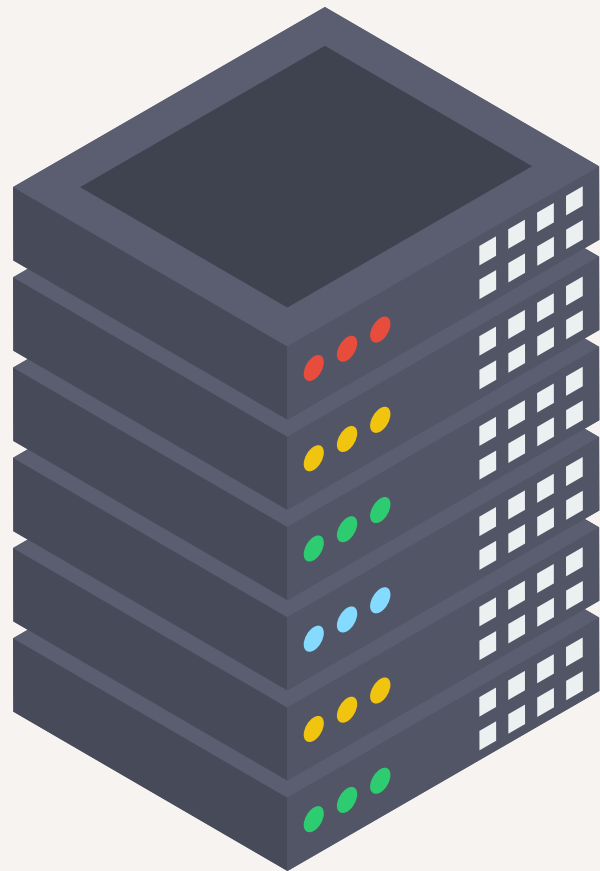
---

遠慮せずに挙手・発言してください！

# 小休止



# サーバーの役割や 設定を確認しよう



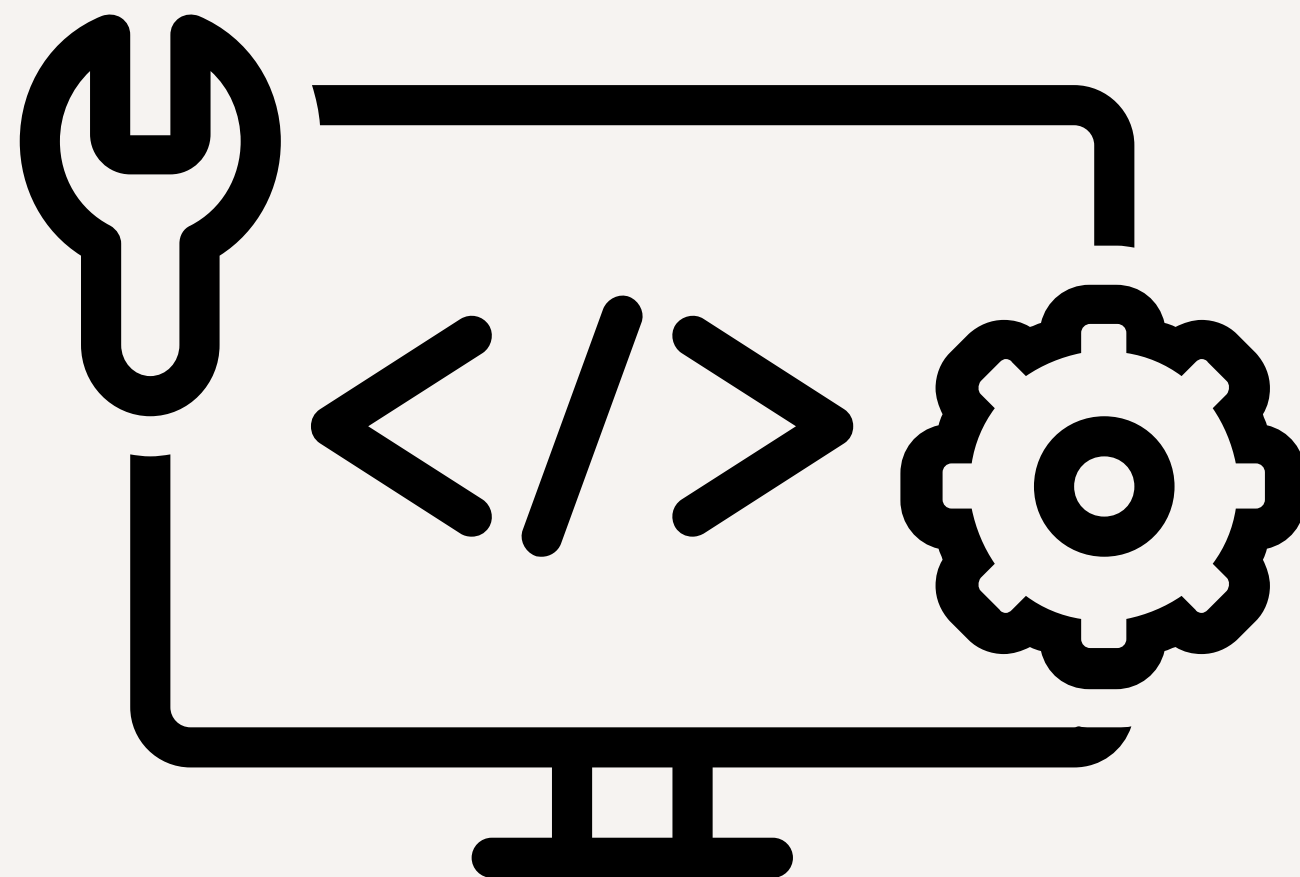
プランや契約状況を教えてください。

※クレジットカードなどの契約「情報」は不要です。

- ドメインは有効か
- メールは使えるか
- DBは使えるか
- PHPのバージョン,他



# 開発環境を整えよう



# 「VSCode」の準備

---

無料エディタとして有名な「VS Code」を使います。プラグインなどは必要になった時に追加していきます。今回は、以下のチェックリストを完遂してください。

- ✓ ソフトウェアは最新版になっていますか
- ✓ どのようなプラグインを入れていますか
- ✓ ショートカットは使えますか(コピペなどの汎用やファイル検索など)
- ✓ 日本語化やモードの選択は適切ですか

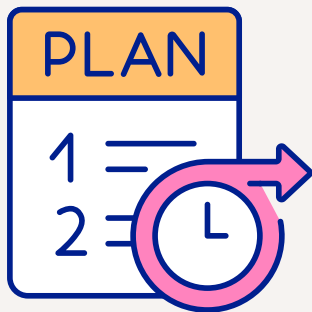


ここまでで  
質問はありますか？

---

遠慮せずに挙手・発言してください！

# 今日の講義のまとめ



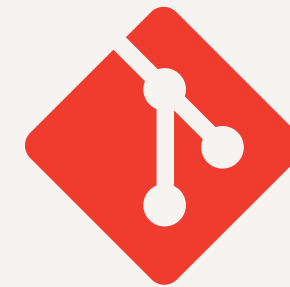
## 学習計画

大まかに今年度の取り組みについて紹介しました。今後も疑問点が出れば随時質問してください。



## 復習

WEB、DNS、FTPなど制作からサーバーに関する情報を復習しました。不明点は無くしておきましょう。



## Gitの整備

これから使っていくツールとして「Git」を導入しました。必須級の強力なツールなので、主な機能は使えるようにしていきます。



## 開発環境の確認

エディタの使いやすさは、開発速度やバグの発生に影響します。自分に最適な環境を作っていきます。

# WEBの知識確認など



- DartSassは何のために使ってますか(cssとの違い)
- コーディングルールなどは知っていますか
- jQueryについて説明してみてください
- 画像の形式について用途を説明してください
  - 写真、アイコン、ファビコンなど



# HTML/CSS II

---

実習2 Gitの使い方／前年の復習  
- HTMLマークアップとCSS構文の使い方

4月	5月	6月	7月
<ul style="list-style-type: none"><li>• 前期授業開始</li></ul>	※LP完了までJS止め	<ul style="list-style-type: none"><li>• LP制作完了(PF可)</li></ul>	
8月	9月	10月	11月
	<ul style="list-style-type: none"><li>• 後期授業開始</li></ul>		
12月	1月	2月	3月
	<ul style="list-style-type: none"><li>• WordPress制作完了</li><li>• 講義終了</li></ul>		

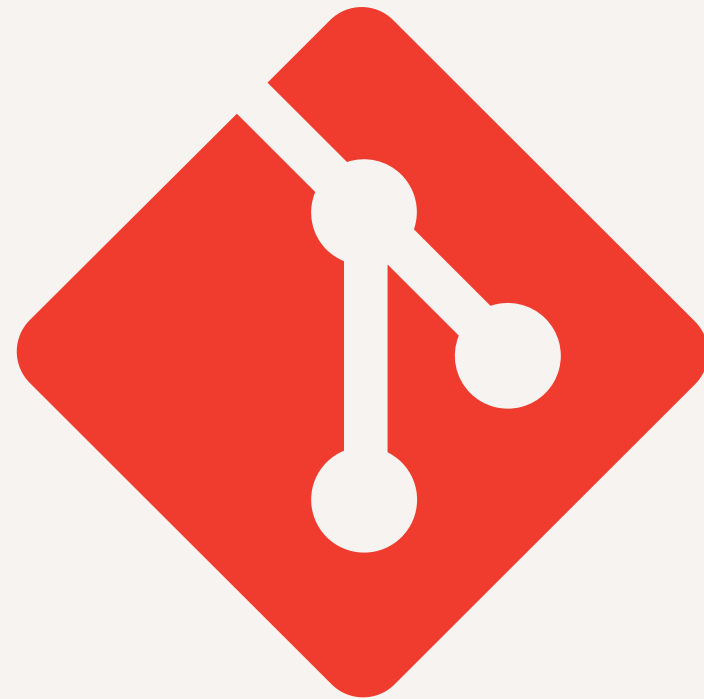
# 今回の内容

---

1. Gitの用語と使い方を学ぼう
2. Gitを実践してみよう
3. 今日の課題をgithubリポジトリから取得してみよう
4. プライベートブランチを作ってみよう
5. HTML／CSSの基礎問題を解いてみよう



# Gitの用語と 使い方を学ぼう

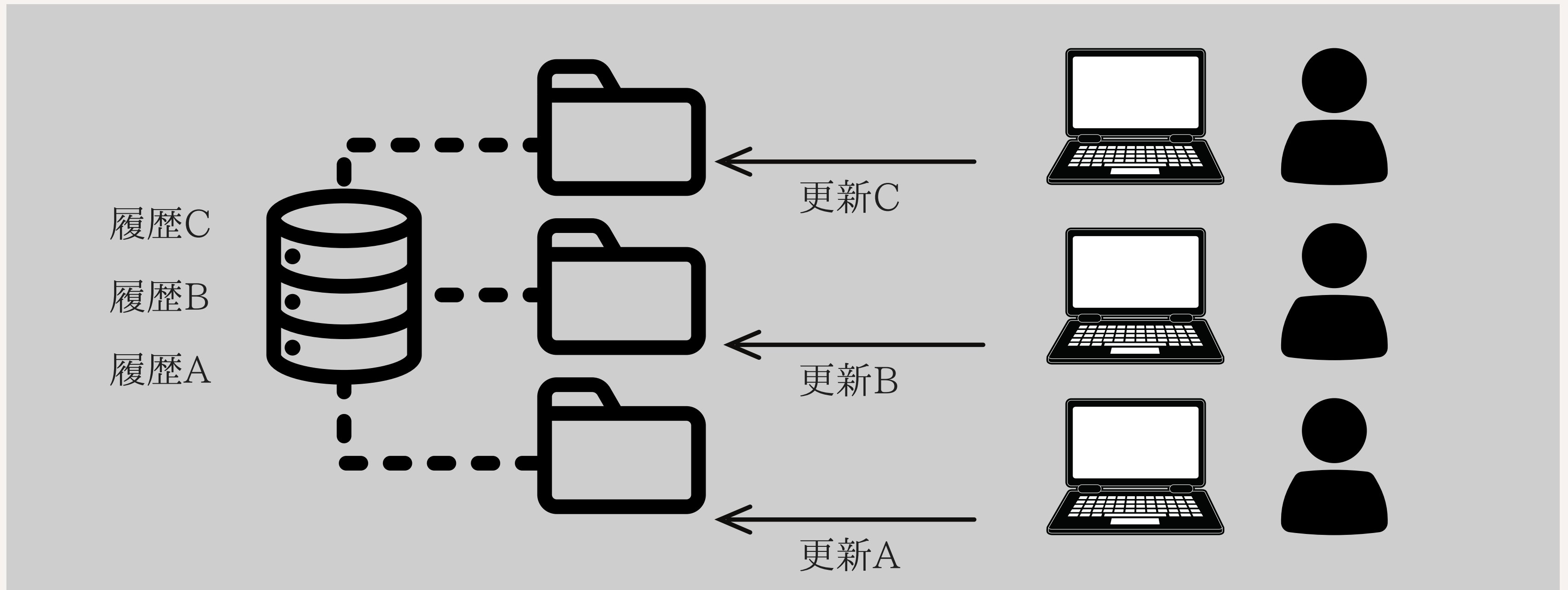


**git**

Gitの用語と使い方を学ぼう

# 分散型バージョン管理システム

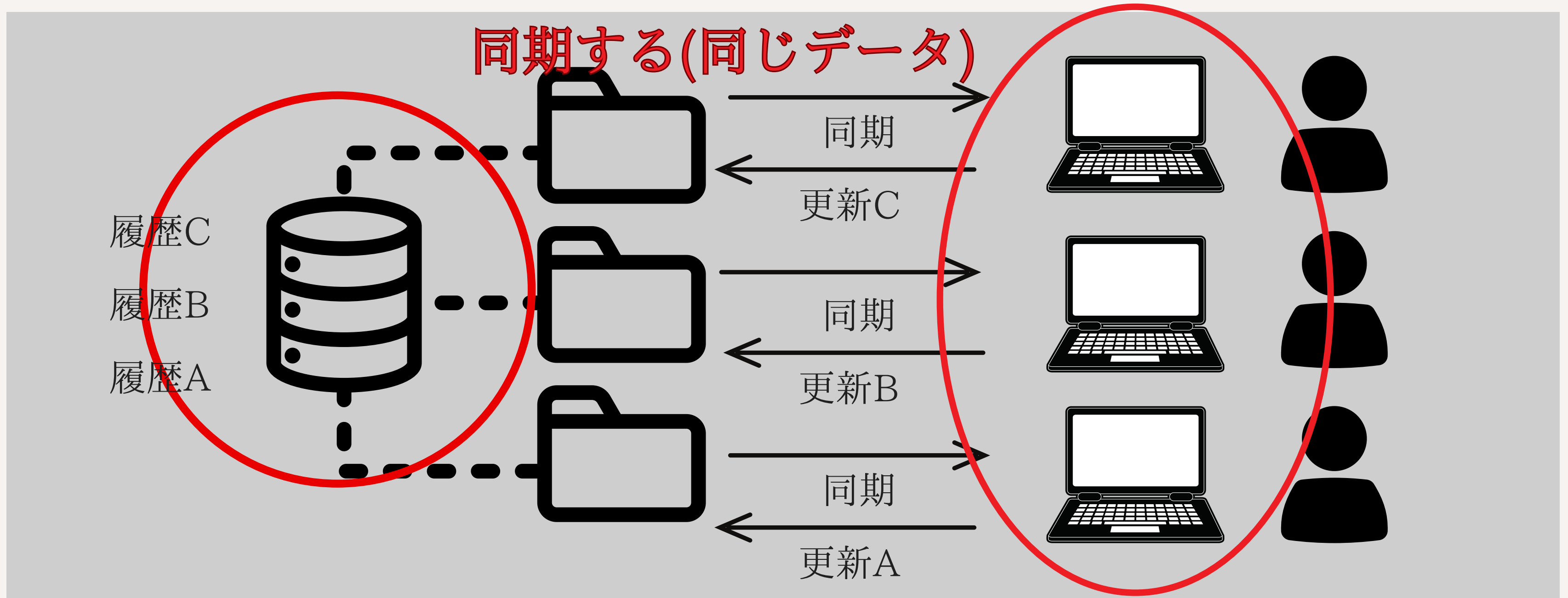
クラウドの共有ストレージと、各端末内の両方でデータを管理・共有するシステム。



Gitの用語と使い方を学ぼう

# 分散型バージョン管理システム

クラウドの共有ストレージと、各端末内の両方でデータを管理・共有するシステム。

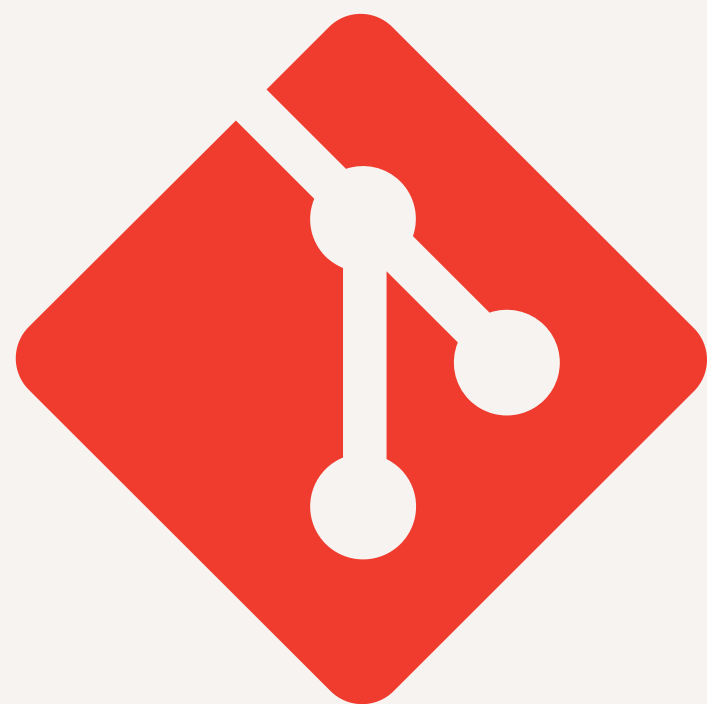


# 主な用語集

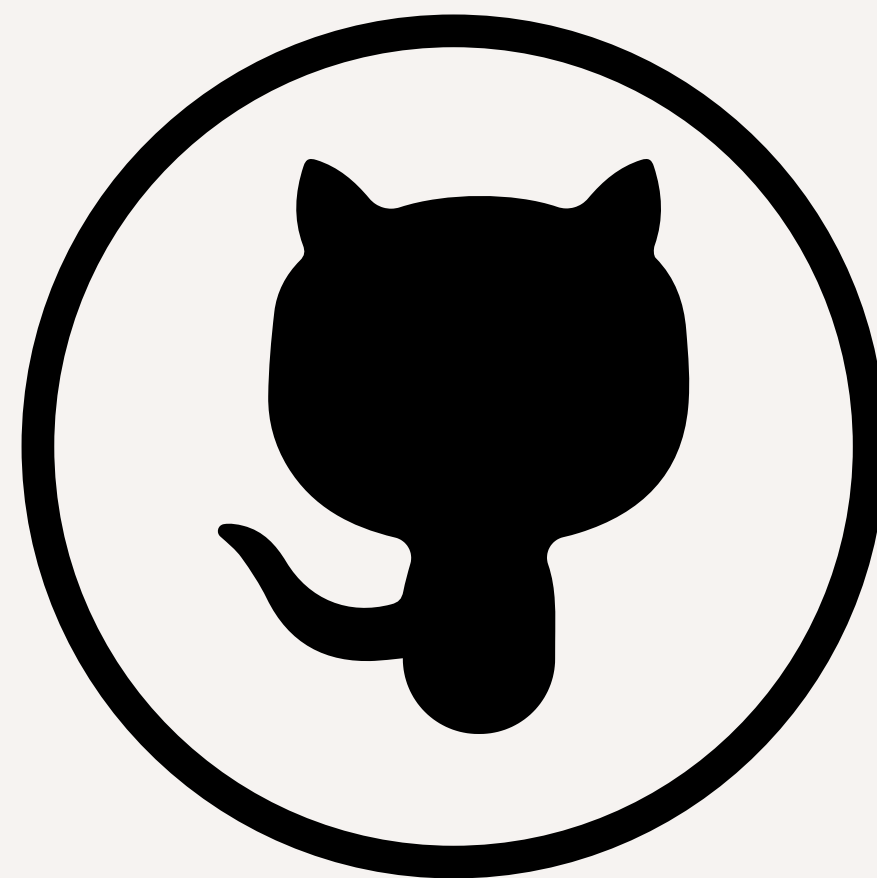
---

- リポジトリ: ファイル自体や変更履歴、設定情報などが含まれたメインとなるデータベース
- 作業コピー／作業ツリー: リポジトリで管理しているファイルまたはツリー(機密など管理しないファイルも設定可能)
- コミット: 作業コピーの変更を履歴として保存する機能で、いわゆる1世代のデータにあたる
- チェックアウト: ローカルリポジトリのある時点でのファイルを取り出す、過去／別バージョンへの切り替えが可能
- ブランチ: 独立した作業ツリーで、分岐と合流を繰り返しファイルを更新する単位
- マージ: ファイルやブランチを取り込む(合流)し更新する作業
- コンフリクト: マージした際に同じファイルを更新しどちらが正か機械的に判断できなくなる現象、この場合は手動で修正をする必要がある
- クローン／フォーク: クラウドにあるリモートリポジトリをローカルリポジトリにダウンロード作成する作業
- フェッチ: リモートリポジトリとローカルリポジトリ間で更新があったかを確認する
- プル: リモートリポジトリの内容をローカルリポジトリに取り込む、フェッチ+マージを合成した機能
- プッシュ: ローカルリポジトリでコミットした履歴などをリモートリポジトリに反映させる作業
- プルリクエスト: ブランチ間のマージをクラウド上で発行する機能、これを通してコードレビューやドキュメント管理などとの連携がしやすくなる

Gitを実践してみよう



git



Gitを実践してみよう

# テスト用リポジトリの作成

---

1. 「github」で「プライベートリポジトリ」として「test」を作成してみよう
  - a. パブリック設定にした場合は公開設定で誰でもアクセスできてしまいます(OSSなどはこの形式)
  - b. その他のオプションはデフォルトで良い
2. 「sourcetree」を立ち上げ、作成したリポジトリのデータをHTTPSで「クローン」する
  - a. 資格情報が求められたらgithubのユーザー情報を入力するかブラウザで認証する
3. ローカルに作成したリポジトリ直下に「test.txt」を作成し、「テストです」と入力して保存
4. 「sourcetree」上で変更を確認し、「ステージ」に追加した後メッセージを入力して「コミット」→「プッシュ」
5. 「github」に戻ると、「プッシュ」した内容が反映される
6. 「sourcetree」にもどり、「ブランチ」から「feature/first」ブランチを作成&チェックアウト
7. 「merge.txt」を新規で作成。「マージをテストします」など入力して保存し、「4」に倣い「プッシュ」まで実行
  - a. 「sourcetree」の☒マークを付けること(リモートに初めて登録する場合に使用する)
8. 「github」を開きグロナビの「Compare & Pull request」(以後PR)→内容を確認してPRを作成、そのまま
  - a. PRの元が「feature/first」、先が「main」で、「main」ブランチに合流するような操作になる
9. PRを承認し、「sourcetree」の「フェッチ」でリモート側が更新されたことを確認する(mainに「2↓」が表示)
10. 「main」ブランチをダブルクリックしチェックアウト、「プル」でリモートのデータをローカルに同期
  - a. 「プッシュ」と「プル」はローカル・リモート間で逆に近い動作をする

Gitを実践してみよう

# コンフリクトを学ぶ

1. 「sourcetree」の「main」ブランチから新規で「ブランチ」「feature/conf1」を作成しチェックアウト
2. 先の項目で生成した「merge.txt」の2行目として「2行目のデータ」などを入力して保存
3. ステージに追加して「コミット」まで済ませる
4. 再度「**main**」**ブランチ**をチェックアウトし、新規で「ブランチ」「feature/conf2」を作成しチェックアウト
  - a. ※2行目の「データ1」が消えているいるので確認すること(=mainから分岐、枝分かれしたイメージ)
5. 先の項目で生成した「merge.txt」の2行目として「コンフリクトを確認」などを入力して保存
6. ステージに追加して「コミット」まで済ませる
7. 再再度「**main**」**ブランチ**をチェックアウトし、ここまでの一時停止
  - a. ※この時点の「merge.txt」は「マージをテストします」1行のみである
8. 今の「main」ブランチに、「feature/conf1」ブランチをマージ
  - a. サイドメニューの右クリック→「現在のブランチに〇〇ブランチをマージ」が分かりやすい、fast-fowardはOn
9. 続けて、「feature/conf2」ブランチをマージ
10. コンフリクトが起きるのでVSCodeで状況を確認してみる
11. 今回の場合は「Accept Both Changes」が適当で、両方の変更を取り込み内容の通りの行数になるよう調整
12. 保存が完了したら、「sourcetree
13. 「github」で画面を確認してみる **※このリポジトリは後で削除して構いません**



Gitを実践してみよう

# その他のポイント

---

- 作業コピーはファイルの他、自動で設定される変更ブロック、行単位で変更できる
- PR上のコンフリクトはローカルマージでも解決可能(上位工程のため省略)
- コミットメッセージはルールがあると統一しやすい
  - 参考: <https://gist.github.com/joshbучea/6f47e86d2510bce28f8e7f42ae84c716>
- コミットの単位(タイミング)はプロジェクトごとに設定する
  - 一般には機能ごとと言われるが、機能の捉え方による
- バイナリファイル(テキストで判別できないExcelやAdobe系など)は差分が人間には判断できない
- 初心者時点での「リベース」や「リセット」機能は要注意、ただし良い機能ではある
- 設定によって管理しないファイルを設定できる



今日の課題をgithubリポジトリ  
から取得してみよう





ここまでで  
質問はありますか？

---

遠慮せずに挙手・発言してください！

今日の課題をgithubリポジトリから取得してみよう

# ブランチの場所

---

- URL: <https://github.com/nichide-beni/html-css2>
- 上記URLからクローンまたはzipでダウンロードする
  - クローンの場合は今後の変更も追うことが可能
    - 書き換えのリスクがあるため、演習ファイルは各自分のディレクトリに管理すること
- クローンとは別で課題用のリポジトリとディレクトリを作ろう
  - リポジトリ: 「html-css-learn」で、「~/html-css-learn」に保存など(任意)

# 小休止



# HTML／CSSの基礎問題を解いてみよう

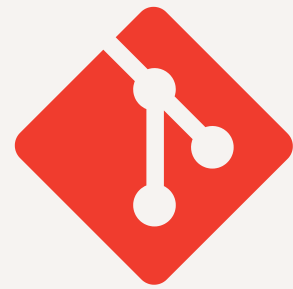


HTML／CSSの基礎問題を解いてみよう

# 設問項目について

- 
- URL: [https://docs.google.com/document/d/1AQxZZnv7ZgCb9fxHyT1qUUOF\\_-4ybNjIUIoc4nwoPD8/edit?usp=sharing](https://docs.google.com/document/d/1AQxZZnv7ZgCb9fxHyT1qUUOF_-4ybNjIUIoc4nwoPD8/edit?usp=sharing)
  - 権限のあるアカウントで閲覧した場合は、見ることが可能
  - 回答についてはリポジトリから取得したデータをコピーして使用すること

# 今回の講義のまとめ



## Gitの使い方

基礎的な使い方とリポジトリのサービスを学び、これから使っていくための下地を整える。



## 復習と知識の確認

問題形式の課題を解くことによって知識の確認を行う。また未学習のコンテンツがあればこの場で身につける。

# 今回の内容

---

1. HTML／CSSの発展問題を解いてみよう



HTML／CSSの応用問題を解い  
てみよう

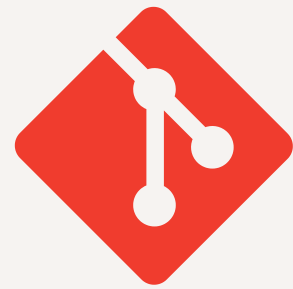


HTML／CSSの応用問題を解いてみよう

# 設問項目について

- 
- URL: [https://docs.google.com/document/d/1AQxZZnv7ZgCb9fxHyT1qUUOF\\_-4ybNjIUIoc4nwoPD8/edit?usp=sharing](https://docs.google.com/document/d/1AQxZZnv7ZgCb9fxHyT1qUUOF_-4ybNjIUIoc4nwoPD8/edit?usp=sharing)
  - 権限のあるアカウントで閲覧した場合は、見ることが可能
  - 回答についてはリポジトリから取得したデータをコピーして使用すること

# 今回の講義のまとめ



## Gitの使い方

実際に自分のコンテンツとして使うことで、疑問点や習得したことを整理する。



## 他人のコード修正

一部コーディング済みのファイルを修正する作業を通し、他人のコードを修正する技術を習得する。