

開発環境の準備

1. **GitHub Codespaces**の利用:
 - クラウド上で開発環境を提供するサービスで、ブラウザから直接開発可能
 - 環境構築不要で、どのPCからでも同じ開発環境を使用可能
2. **GitHub**リポジトリの作成:
 - GitHubにログインし、新規リポジトリを作成
 - READMEファイルを追加して、初期設定を完了
3. **Codespaces**の起動:
 - 作成したリポジトリから新しいCodespaceを作成
 - 環境準備には1〜3分程度かかる場合あり
4. 開発環境の構成:
 - VS Codeベースのインターフェースでファイルエクスプローラー、エディター、ターミナルを使用
5. **Gemini CLI**の起動:
 - ターミナルで`npx @google/gemini-cli`を実行
 - Googleアカウントでログインし、認証コードを使用して接続

Gemini CLIでWebサイトを作成する

主なポイント:

1. **Gemini CLI**のインストールと基本操作
 - Gemini CLIを使用してHTMLファイルを作成。
 - コマンド実行時はユーザーに許可を求める仕組み(例: `Yes, allow once` など)
2. **HTML**ファイルの確認
 - 簡易WebサーバーをPythonコマンド(`python3 -m http.server 5500`)で立ち上げ、ブラウザで確認
3. **Git**を使用したバージョン管理
 - `git add` → `git commit` → `git push`で変更内容をGitHubにアップロード
 - Gemini CLIを使えば、これらのGit操作も簡略化可能←おすすめ
4. **GitHub Pages**での公開
 - GitHubリポジトリにアップロードされたHTMLファイルをGitHub Pages経由で公開

GitHub Pagesを使ってデプロイ

1. GitHub Pagesの概要

GitHubが提供する無料のウェブサイトホスティングサービスで、HTML、CSS、JavaScriptファイルを公開できます

2. デプロイ手順

- リポジトリの準備: GitHub上にプロジェクト用のリポジトリを作成
- コードのpush: ローカルで作成したファイルをGitHubにアップロード
- **GitHub Pages**の設定: リポジトリ設定で「Pages」を選択し、公開ブランチを指定
- サイトの公開確認: 発行されたURLでサイト表示を確認

3. デプロイの確認方法

- **Actions**タブでデプロイの成功・失敗を確認
- エラー時はログを参照

4. よくあるトラブルと対処法

- 反映までに数分かかることがある
- `index.html`の配置場所の確認
- リポジトリがパブリックかプライベートかの設定確認

CSS、JavaScriptでWebサイトをアップグレード

1. Codespaceの再開

- 以前作成したGitHub Codespaceを再起動して作業を継続。GitHubリポジトリ画面から「Codespaces」タブを選び、ワンクリックで同じ環境を再利用できる

2. HTMLの編集

- **Gemini CLI**を使ってテキストや画像を追加・更新
- 自己紹介ページの作成やレイアウト変更が可能
- ファイルの差分表示を確認しながら修正を反映できる

3. 画像の追加

- GitHub Codespacesのファイルエクスプローラーに画像をドラッグ & ドロップしてアップロード
- HTML内にタグを追加して表示
- 画像ファイルはリポジトリ内に保存され、GitHub Pagesでもそのまま利用できる

4. CSSの追加

- **style.css**を作成し、色・フォント・レイアウトなど見た目を自由にカスタマイズ
- 「ハッカー風」や「レトロ風」などテーマを指定してデザインを整えることも可能
- 変更はブラウザのハードリロードで即時反映。

5. JavaScriptの追加

- ページに動きを与えるスクリプトを追加
- 例: タイピング風アニメーション、ボタン操作による背景色の変更、キー操作による演出
- 動作確認にはブラウザの開発者ツール(Console)を使用

6. HTML/CSS/JSの役割と連携

- HTML: ページの構造(骨組み)とコンテンツ
- CSS: 見た目のデザインや装飾
- JavaScript: 動的な動きやユーザー操作への反応

7. 外部ライブラリ活用

- より高度な表現のために外部ライブラリを使用可能
- p5.js: 2Dアニメーションやインタラクティブな演出
- three.js: 3D表現やWebGLを用いたリッチな演出

Gemini CLI Tips

1. Web検索機能

- Gemini CLIはインターネットを検索して情報を取得し、要約したりHTMLファイルを生成したりできる

2. ファイル／画像の添付

- Gemini CLIに画像ファイルを渡して、デザインや解析の参考にできる
- まずCodespaces上に画像をアップロード(ドラッグ & ドロップまたは「Upload」)
- Gemini CLI の入力欄で **@ファイルパス** を指定してそのファイルを参照可能
- パスには絶対パス／相対パスの概念がある(例: **/images/apple-hp.png** など)

3. Gemini CLI コマンド活用

- **/compress**:これまでの会話を要約して文脈を整理する
- **/init**:既存プロジェクトに Gemini CLI を導入する初期設定を自動化
 - フォルダ内のファイルを自動読み込み
 - プロジェクト構造を分析
 - **GEMINI.md** という仕様書ファイルを生成
- **/clear**:会話履歴をリセットして新しい話題をゼロから始める