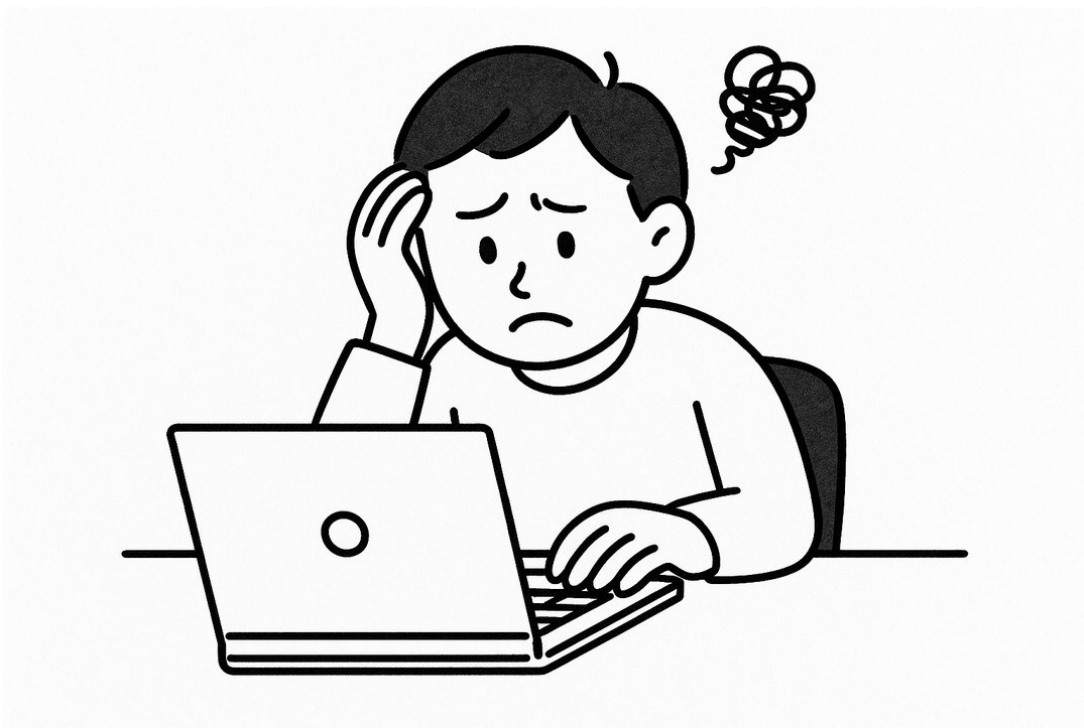
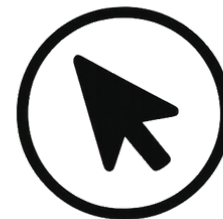


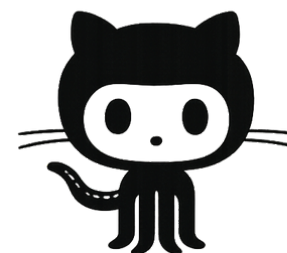
AIと一緒に開発をはじめよう



Replit



Cursor

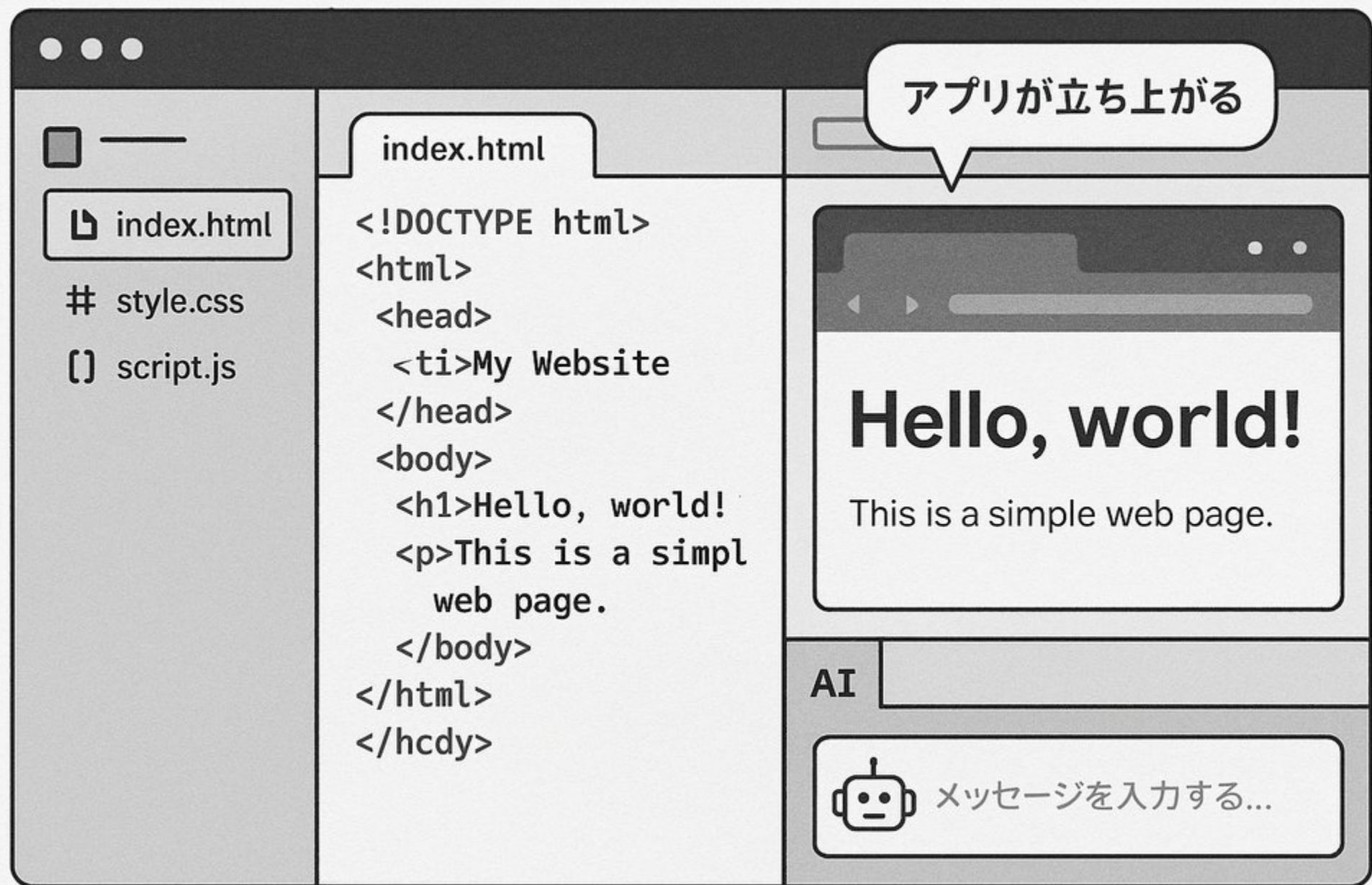


GitHub

日々の業務で「同じ作業を何度も繰り返して大変...」と感じることはありませんか？
今日はReplit・Cursor・Githubを使って、AIに助けてもらいながらアプリを作る体験を通じて、
効率化の第一歩を学びます。

Replitでアプリの大枠をつくる

Replitはブラウザ上でAIがコードを生成してくれるので、環境構築不要。
アプリの骨組みを素早く作り、すぐに動かして試せるのが魅力です。



特徴

AIがプロンプトに沿って
アプリを生成

メリット

環境構築不要
ワンクリックでデプロイ可能

留意点

細かいコード修正は苦手
→ Cursorと組み合わせる

Cursorでコードを調整する

CursorはAIエージェントと一緒にファイル単位でコードを修正できるエディタです。
Replitで作ったアプリをさらに使いやすく整えていきます。

```
1 def greet(name):  
2     print('Hello, ' + name + '!')  
3     greet('Alice')  
4
```

How can I fix
the error?



Replace "+" with
"," in the print
statement.

|

特徴

AIとの対話でコードを改善

活用例

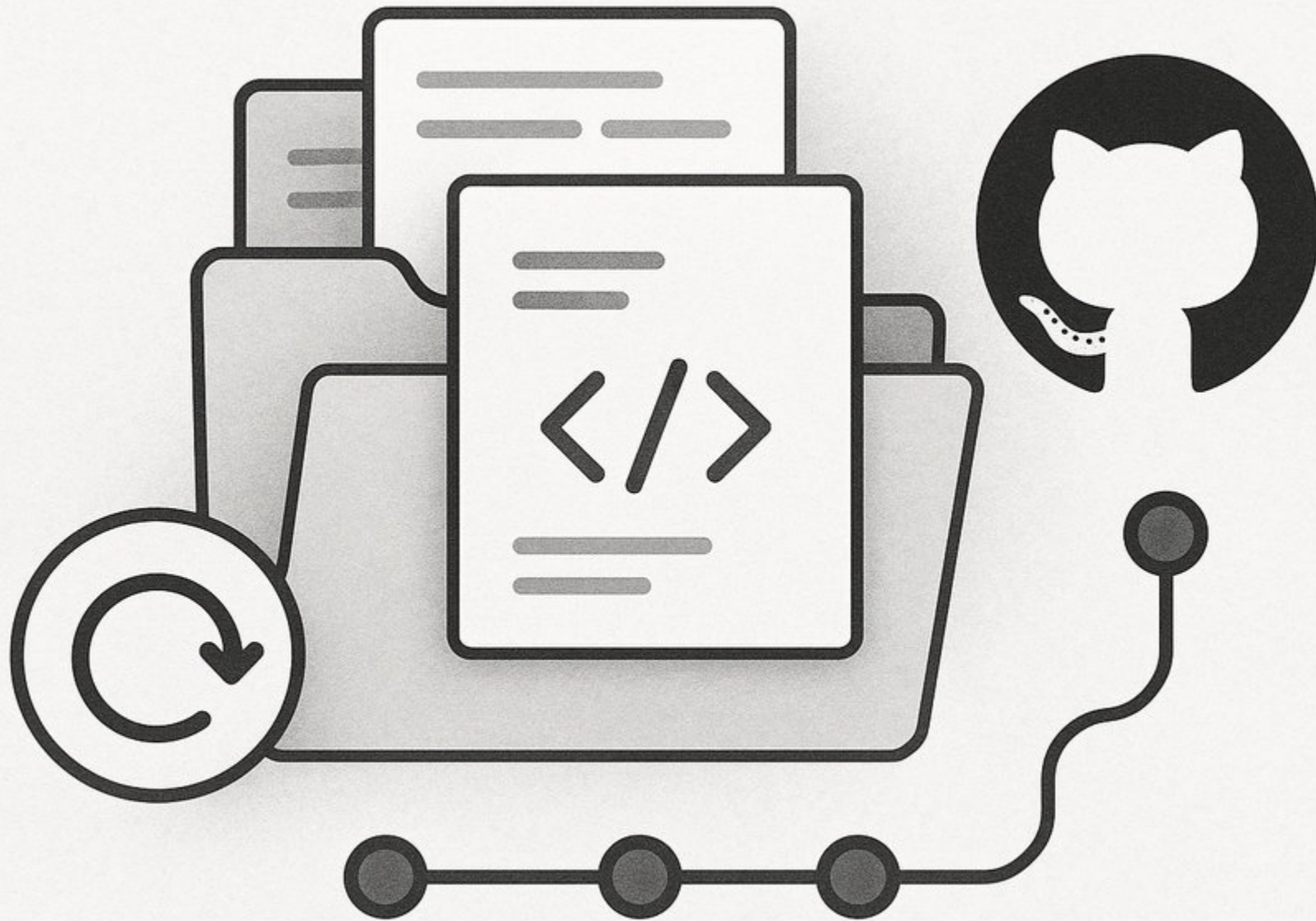
ボタンの動作変更
デザイン調整

メリット

非エンジニアでも
「質問しながら修正」できる

Githubでコードを管理する

Githubは作ったコードの保管庫。修正の履歴を残し、チームでも安心して開発を進められます。
ReplitやCursorから連携して、いつでも安全に管理しましょう。



特徴

バージョン管理
バックアップ

メリット

ミスしても元に戻せる
安心感

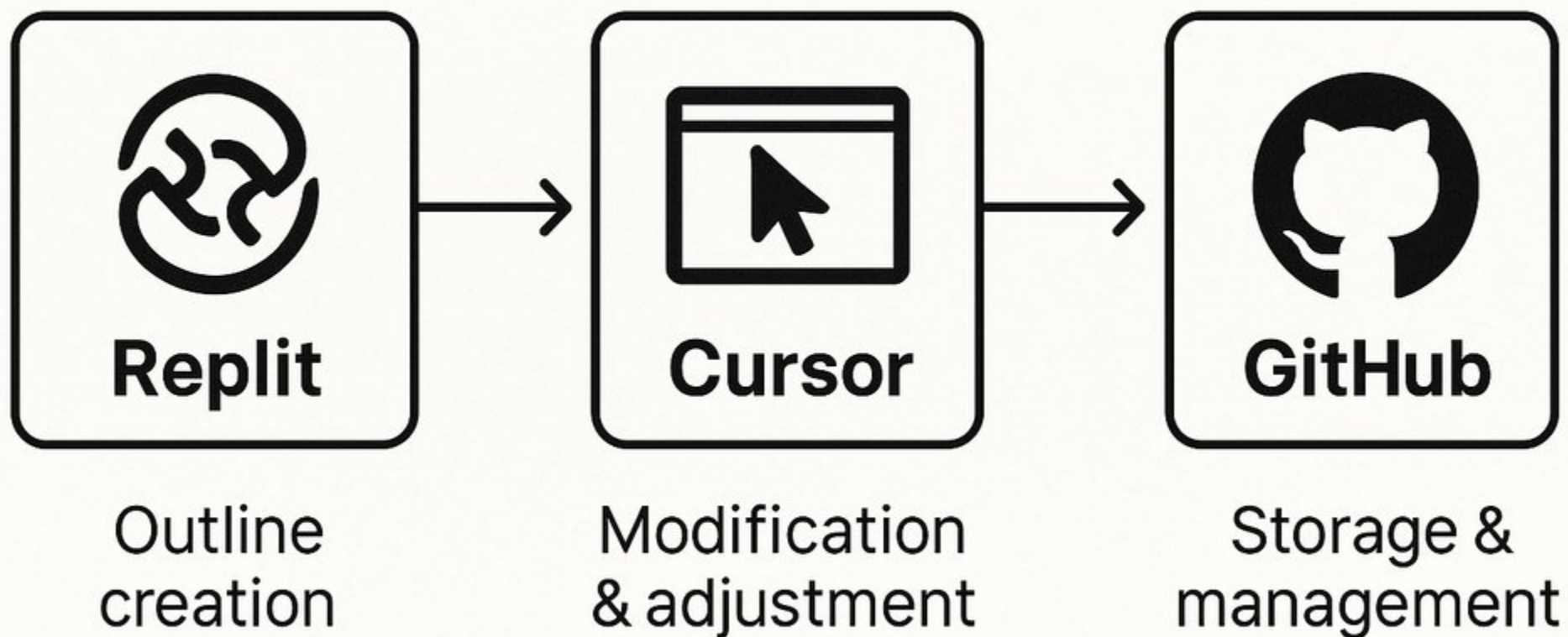
使い方

Cursorから「プッシュ」して
反映

Replit → Cursor → Github の流れ

実際の開発は「Replitで大枠を作成 → Cursorで修正 → Githubで保存」という流れです。
これらをつなげて使うと、効率的で安全に開発できます。

DEVELOPMENT FLOW



ReplitとCursorの連携手順

1. Replitでコードをエクスポート
2. Cursorでプロジェクトを開く
3. コードを修正・調整する
(無料プランでも基本機能は利用可能)

CursorとGithubの連携手順

1. Cursorで「Git」タブを開く
2. コミットメッセージを入力
3. 「Push」ボタンでGithubに反映
(簡単なコマンドで履歴管理が可能)

今日の学びと活用イメージ

Replit・Cursor・Githubを組み合わせると「アプリをつくって育てる」流れが実現できます。
例えば、社内の簡単なツール開発や業務自動化にすぐ活かします。

ツール	役割	メリット	使い方
Replit アプリの大枠作成	AIがコードを生成 環境構築不要	素早くプロトタイプ作成 すぐに動かして確認できる	アイデアを文章で入力 自動でコード生成
Cursor コードの調整・修正	AIとの対話で改善 細部の調整が可能	非エンジニアでも修正可能 質問しながら進められる	質問形式でコード修正 具体的な動作を指示
Github コードの保存・管理	バージョン管理 安全なバックアップ	ミスしても元に戻せる チーム開発に対応	定期的にPushで保存 変更履歴を管理

今日の要点整理

Replit = 大枠作成
Cursor = 細かい調整
Github = 安全な保存

応用例

社内FAQボット
入力補助アプリ
データ集計ツール

ポイント

小さく試してみる
成功体験を積み重ねる
自信につなげる

小さな一歩を試してみよう

Replitでサンプルアプリを立ち上げ、Cursorで文字を変更し、Githubに保存してみましょう。
「できた！」という実感が、次の一歩につながります。

- ☐ Replitで「Hello World」アプリを作成する
- ☐ Cursorで「Hello, AI!」に修正する
- ☐ Githubに保存（Push）して履歴を確認する

自己肯定感を育てる問い：

"これなら自分の業務でも使えそう？"



できた！この調子で次のステップへ