一覧

🌏 超スモールステップアップ・エンジニアロードマップ（全150ステップ）

～小さな一歩を、確実な成長に変える～

TechField コーチ リュウセイ監修

🟩 Chapter 1｜成長の準備とマインドセット（STEP 1～10）

1. 学ぶ時間を15～30分で固定する（例：夜20:00～20:30）
2. 学習用PCを1台決める
3. ネット環境とChromeを整える
4. Visual Studio Code（VSCode）をインストール
5. 「Hello, World!」を出力して動作確認する
6. デスクトップに「techfield」フォルダを作成
7. Gmail・Google DriveなどGoogle環境を整える
8. Chromeのブックマークを整理する（学習／資料／成果）
9. 「3ヶ月後にどうなりたいか」を1行で書き出す
10. 今日の「できた！」を1つ書き残す

🟨 Chapter 2｜HTML・CSSでWebの土台を作る（STEP 11～40）

1. HTMLとは何かを調べる
2. index.html ファイルを作成する
3. <!DOCTYPE html> と <html> 構造を理解する
4. <h1> タグでタイトルを書く
5. <p> タグで文章を追加する
6. <a> タグでリンクを貼る
7. <img> タグで画像を表示する
8.   でリストを作る
9. <table> で表を作成する
10. CSSとは何かを知る
11. style.css を作りHTMLに読み込む
12. 文字色を変更する
13. 背景色を変更する
14. フォントサイズを変更する
15. 余白（margin/padding）を調整する
16. 画像サイズを変更する
17. classを使って複数要素を装飾
18. idを使って特定要素を変更
19. ページ全体の背景を設定
20. ボタン風デザインを作る
21. ナビゲーションメニューを作る
22. Flexboxでレイアウトする
23. ページ下にフッターを追加
24. ホバーアクションを設定
25. ファビコンを設定
26. 自己紹介ページを作る
27. 作品紹介セクションを追加
28. CSSで全体を整える
29. フォルダ構造を整理する

30. ページをブラウザで確認する

🟦 Chapter 3｜JavaScriptで動きをつける（STEP 41〜60）

1. JavaScriptとは何かを調べる
2. script.js をHTMLに読み込む
3. console.log() で出力
4. alert() でメッセージを表示
5. 変数（let/const）を使ってみる
6. if文で条件分岐を学ぶ
7. for文で繰り返しを体験する
8. 配列を使ってデータをまとめる
9. 関数（function）を作って呼び出す
10. ボタンをクリックして関数を実行
11. document.getElementById() を使って要素操作
12. クリックでテキストを変更する
13. Math.random() でランダムな数字を出す
14. おみくじを作る（ランダム結果）
15. クイズ問題データを配列で作る
16. 次の問題が表示される仕組みを作る
17. 正解・不正解を判定する
18. 結果をカウントして表示する
19. 「もう一度遊ぶ」ボタンを追加
20. 3問クイズアプリを完成させる

🟧 Chapter 4｜PHPで“仕組み”を作る（STEP 61〜80）

1. PHPとは何かを調べる
2. サーバーとは何かを理解する
3. MAMPをインストール
4. MAMPを起動して localhost にアクセス
5. 「htdocs」フォルダを開く
6. index.php を新規作成
7. <?php echo "Hello, PHP!"; ?> を出力
8. HTMLとPHPを混ぜて書いてみる
9. PHPで変数 $name を使う
10. if文で条件を分岐する
11. for文で繰り返す
12. 配列を作ってデータをまとめる
13. フォームをHTMLで作成する
14. $_POST で送信データを受け取る
15. 入力内容を画面に表示する
16. fopen() と fwrite() でファイルに保存
17. ファイルを読み込んで一覧表示
18. メモ帳アプリを完成させる（入力→保存→表示）
19. エラーが出たら検索して解決する
20. コードを整理してフォルダに保存

🟥 Chapter 5｜データベースとCRUDの基本操作（STEP 81〜115）

1. MAMPでphpMyAdminを開く
2. phpMyAdminの役割を理解する
3. 新しいデータベースを作成（techfield_db）
4. テーブル memos を作成（id, content, created_at）
5. 手動で1件データを追加してみる
6. PHPでDBに接続する（mysqli_connect()）
7. 接続テストを行う
8. フォームから送信されたデータをINSERT
9. 登録完了メッセージを表示
10. phpMyAdminでデータが追加されたか確認
11. PHPでSELECT文を書いて全件取得
12. while文でデータを1行ずつ出力
13. HTMLテーブルに整形して表示
14. 一覧ページを完成させる
15. 登録→表示までの流れを整理
16. 各データに「編集」リンクを追加
17. 編集ページでデータを取得
18. UPDATEで内容を変更
19. 変更完了メッセージを表示
20. 一覧ページにリダイレクト
21. 各データに「削除」ボタンを追加
22. 削除確認メッセージを表示
23. DELETE文でデータを削除
24. 削除後にメッセージを表示
25. phpMyAdminでデータ削除確認
26. 登録・一覧・編集・削除を一連で動かす
27. isset / empty で安全対策をする
28. コメントを整理してコードを理解
29. CRUDメモアプリとして保存
30. アプリの仕組みをノートにまとめる
31. DB接続設定を整理して再利用可能にする
32. SQLインジェクションを簡単に調べる
33. prepare / bind_param を学ぶ
34. CRUD全体を関数化して再構築
35. ローカルで安定して動作するまで検証

🟦 Chapter 6｜XserverでWeb公開とポートフォリオ化（STEP 116〜150）

1. Xserverにログイン
2. 契約状況を確認
3. サーバーパネルで「ドメイン設定」を開く
4. 独自ドメインを登録（例：example.com）
5. 反映まで待機
6. ファイルマネージャーを開く

7. public_html にフォルダを作る（例：portfolio）
8. CRUDアプリをアップロード
9. ファイル権限を確認（644/755）
10. URLでアクセスして動作確認
11. 「SSL設定」で無料独自SSLを有効化
12. https:// にリダイレクトを設定
13. .htaccess でリダイレクト追加
14. 再度アクセスしてSSL確認
15. 公開完了！
16. フォルダを「app」「css」「img」「db」などに整理
17. 不要ファイルを削除
18. indexページに作品一覧を追加
19. ローカルにバックアップ保存
20. 修正→再アップロードで更新体験
21. 新フォルダ portfolio を作成
22. index.html に自己紹介を記載
23. 各作品のURL・概要・技術・期間をまとめる
24. CSSで見やすく整える
25. ページ全体を確認
26. SNS（Xなど）で公開報告ポストを作る
27. #ポートフォリオ公開 #プログラミング初心者 タグをつける
28. フィードバックを受けたら改善
29. 毎月1回はサイトを更新する
30. 新しい作品を追加して進化させる
31. 公開作品をGitHubに保存する
32. Gitをインストールして初期設定
33. commit/pushの流れを学ぶ
34. README.md に作品概要を書く
35. 次に挑戦したいテーマ（API連携／ログイン機能など）を決める

スプレッドシートはこちら
https://docs.google.com/spreadsheets/d/1QY9Kbnd-cloG0odapEU28KaloNPztkbcV90dXFIhPis/edit?usp=sharing
（コピーして使用してください。）

🎓 完走後に得られるスキル一覧

分野	できるようになること
HTML/CSS	自分でWebサイトをデザイン・構築できる
JavaScript	ページに動きをつけられる（クイズ・動作系アプリ）
PHP	サーバーサイドの仕組みを理解し、Webアプリが作れる
MySQL	データを扱い、CRUD操作で動的アプリを構築できる
サーバー運用	XserverでWebを公開し、SSL対応・更新管理ができる
実務基礎	“自走できるエンジニア”としての土台を完成させる

💬 リュウセイより
この150ステップを終えたあなたは、
「学ぶ人」ではなく「創る人」です。
誰に頼らず、自分の力で動くものを作り、公開できた。
それはもう、立派なエンジニアの始まりです。

序章

超スモールステップアップ・エンジニアロードマップ

～小さな一歩を、確実な成長に変える～

TechField コーチ リュウセイ

💬 はじめに

「何から始めたらいいか分からない」
「一度やってみたけど、続かなかった」

そんな人のために、このロードマップを作りました。

ここに書いてあるのは、“知識”ではなく“行動”。
1ステップずつ進むだけで、確実に成長できるように設計しています。

大切なのは、完璧にやることではなく、やめないこと。
今日できることを、小さく積み上げていきましょう。

🎯 このロードマップの目的

- 「できた！」という成功体験を積み上げる
- 学習のつまずきをなくし、継続を習慣化する
- 自分のペースで進めながら、確実にステップアップする
- **TechField**本講座でさらに深く学ぶ準備を整える

🧭 このPDFの使い方

- 各ステップには チェック欄（□）を用意しています。スプレッドシート参照（コピーして使ってください。）
- https://docs.google.com/spreadsheets/d/1QY9Kbnd-cloG0odapEU28KaloNPztkbcV90dXFIhPis/edit?usp=sharing

- 1日1ステップでもOK。空いた時間で進めよう

- わからない箇所はメモを残しておくと、オンライン相談で活用できます

💡リュウセイのポイント

完璧を目指す必要はありません。
大切なのは、今日も“やってみよう”と思えること。

📘 目次

CHEPTER1

🖤 Chapter 1｜成長の準備とマインドセット

——「完璧を目指すな、“始めた人”が勝つ」

💬 INTRODUCTION

多くの人が「プログラミングを始めたい」と言いながら、
実際に動けない理由は“やる気の問題”ではありません。
本当の原因は、どこからどう始めていいか分からないから。

TechFieldの学び方は、最初からすべてを理解しようとしません。
「小さく始めて、少しずつ積み上げる」
この“超スモールステップ”こそが、挫折しない学びの仕組みです。
ここから、あなたの物語が始まります。

⚡ STEP 1｜学ぶ時間を15〜30分で固定する

まずは、“時間を決める”ことから始めよう。
勉強する内容よりも、いつ机に向かうかを決める方が**100**倍大事。
1日たった15分でも、毎日続ければ1ヶ月で7時間半の差が生まれる。
ポイントは「時間」ではなく「リズム」。

“夜のコーヒーを飲んだら学ぶ”“出勤前の10分だけコードを打つ”——
習慣に組み込むと、学びが生活の一部になる。
💡 行動が続く人は「気合い」ではなく「仕組み」で動いている。

⚙️ STEP 2｜学習用PCを1台決め、環境を整える

「今日はMacで」「明日はiPadで」——これでは脳が混乱します。
学びを加速させるコツは、環境を固定すること。
1台のPCを“自分専用の学びの基地”にする。
そこに開発環境、資料、メモ、すべてをまとめる。
迷わない環境が、集中力を最大化します。

💻 STEP 3｜VSCodeをインストールし「Hello, World!」を出す

プログラミングの第一歩は、
世界で一番シンプルな成功体験を味わうこと。
黒い画面に “Hello, World!” の文字が出た瞬間、
それは「何もできなかった自分」から「創れる自分」に変わったサインです。
エラーが出てもOK。
うまくいかなくてもOK。
「出力できた」その事実が、スタートラインです。

🗂️ STEP 4｜「techfield」フォルダを作る

あなたの学びは、今日から“形”になります。
デスクトップに「techfield」というフォルダを作り、
そこにこれから作るすべてのファイルを入れていきましょう。
フォルダの中には、あなたの成長が積み重なっていきます。
1年後、それを見返したとき——
「何もなかった場所からここまで作ったんだ」と思えるはずです。
🧩 最初の小さなファイルが、未来の大きな成果につながる。

☁️ STEP 5｜Google環境を整える（Gmail・Drive・スプレッドシート）

“作る人”になるためには、情報を整理する力も必要です。
Googleアカウントを準備して、Driveやスプレッドシートを使えるようにしましょう。
TechFieldの講義や案件、開発管理も多くがGoogle連携で動いています。
あとでチーム開発をするときも、
「情報をすぐ共有できる」状態が大きな武器になります。

🧭 STEP 6｜Chromeのブックマークを整理する

ブックマークは“思考の地図”です。
学習サイト、リファレンス、YouTubeなど、
必要なページを「学習」「資料」「成果」にフォルダ分けしておきましょう。
混沌とした環境では、思考も散ります。
整理されたブラウザは、集中できる脳を作るツールです。

✍️ STEP 7｜「3ヶ月後にどうなりたいか」を1行で書く

書くことで、ぼんやりしていた夢が“目的”に変わります。
たとえば——
「3ヶ月後、Webサイトを自分で公開できるようになりたい」
「コードが読めるようになって、自信を持ちたい」
どんな小さな目標でもOK。
書くと、行動の方向が定まる。

💡 STEP 8｜「なぜ学びたいのか？」をひと言でまとめる

モチベーションは、目的が明確なほど強くなります。
「自由に働きたい」「自分のサービスを作りたい」「誰かを支えたい」
あなたの“理由”が、壁を乗り越える燃料になります。
他人と比べる必要はありません。
“あなた自身の動機”が、最も強いエネルギーです。

📔 STEP 9｜学習記録ノートを作る

勉強は“やりっぱなし”だと、成果が消えていきます。
ノートアプリでも、紙でもOK。
学んだこと・気づいたこと・詰まったことをメモしておこう。
「昨日より今日の自分が少し理解できている」
その感覚が積み重なることで、学びが「快感」になる。

🌙 STEP 10｜今日の「できた！」を1つ書き残す

たとえ5分でも、1行でもいい。
「できたこと」を書くことで、
脳は“成功体験”を覚えて、やる気を再生産します。
成長は、努力の総量ではなく、継続の回数で決まる。
毎日の小さな「できた」を記録していこう。

🧩 CHAPTERまとめ

ポイント	内容
学びのテーマ	「準備」は行動の一部
目的	習慣化と環境構築
ゴール	“やる気”ではなく“仕組み”で続けられる状態
キーアクション	1日15分×継続＝土台完成

💬 リュウセイから

「完璧になったら始めよう」と思う人ほど、始められない。
小さくても、動いた人が勝つ。
今日、このページを読んで動き出したあなたは、
もう“学ぶ側”ではなく“創る側”です。

📘 次章予告：**Chapter2｜HTML・CSSでWebの土台を作る**
ここから、“手で作る楽しさ”が始まる。
コードが“形になる瞬間”を一緒に体験しよう。

CHEPTER2

⚡ Chapter 2｜HTML・CSSでWebの土台を作る

――「見える成果」が、学びを楽しくする。

💬 INTRODUCTION

人は、「できた！」という実感がないと続けられません。
この章では、最もシンプルな「形になる学び」を通して、
“学ぶ楽しさ”を体に覚え込ませます。
プログラミングとは「創造の言語」。
コードの一行一行が、あなたの考えを“目に見える形”に変えていく。
だからこそ、最初に作るのは“完璧なサイト”ではなく、
“動いた瞬間に心が動くサイト”でいいんです。

🧱 STEP 11〜40｜HTML & CSS 超スモールアップ

🧩 STEP 11｜HTMLとは何かを調べる

HTMLは「Webページの骨格」。
文章、画像、リンク――すべてはこの言語で形づくられます。
「これは何を意味しているのか？」を意識して読むことが、成長の第一歩です。

💻 STEP 12｜index.html ファイルを作成する

“最初の作品”の始まり。
名前は「index.html」にしましょう。Webの世界では、
「index」が“トップページ”を意味します。
💡 ファイルを作るという行為は、「何かを始めた」という証。

🧠 STEP 13〜15｜タグの基礎を触ってみよう

- <h1> … タイトル（見出し）
- <p> … 段落（文章）
- <a> … リンク（他ページやURLへ）

これらを自由に組み合わせるだけで、
“あなたの世界”がブラウザに映し出されます。
動かすたびに、自分の手で創っている感覚を味わってください。

🖼️ STEP 16〜19｜画像・表・リストを表示してみよう

見た目が変わると、学びが一気に楽しくなる。
画像1枚、表1つでも立派な作品。
「形になる＝手応え」
この成功体験の積み重ねが、次のエネルギーになります。

🎨 STEP 20〜21｜CSSで“デザイン”を始めよう

CSSは、HTMLに“命”を吹き込む存在。
色、文字、レイアウト──すべてを支配する「デザインの言語」です。
まずは **style.css** を作ってHTMLに読み込む。
color と **background-color** の2行を変えるだけで、世界が変わります。

💅 STEP 22〜28｜見た目を整える実験をしよう

- 文字サイズを変える
- 背景をグラデーションにする
- 画像を丸く切り取る
- ボタンを光らせる

細かな調整を繰り返すうちに、「美しさ」を感じ取る目が育ちます。
デザインはセンスではなく“観察力”。
💡 *1*ピクセルの違いが、印象を変える。

🧭 STEP 29〜35｜ページを構築する

ページ全体にヘッダー・メイン・フッターを設計して、
Webの基本構造を理解しましょう。
<header>, **<main>**, **<footer>** を使うと、
コードが読みやすくなり、検索エンジンにも好かれます。
“整えること”＝“伝える力”を鍛えることです。

💫 STEP 36〜40｜「自分のページ」を作る

- 自己紹介セクションを追加
- 趣味や目標を書く
- 作品のリンクを貼る
- スマホでも見やすいレイアウトに挑戦
- フォルダ構成を整理

完璧さより「自分らしさ」を優先しよう。
自分の言葉をコードに変えることが、エンジニアとしての最初の表現です。

✨ ここで得られるスキル

分野	できるようになること
HTML	Webページの構造を理解し、自分で設計できる
CSS	見た目を自由にコントロールできる
思考	「構成」や「視覚の伝え方」を意識できるようになる

💬 リュウセイから

最初のサイトは、ダサくていい。
100人が「微妙」と言っても、1人が「すごい」と思ってくれたら、それで十分。
コードを書いて、自分の想いを“形”にできた瞬間──
あなたはもう、創る側の人間です。

📘 次章予告：**Chapter3｜JavaScript**で“動く世界”をつくる
コードが反応し、ページが生き始める。
「仕組みを作る楽しさ」をここから体験していこう。

CHEPTER3

⚙️ Chapter 3｜JavaScriptで動きをつける

——「ページが動くと、学びが生き始める。」

💬 INTRODUCTION

あなたが書いたコードが“反応する”瞬間。
それが、プログラミングの本当の楽しさです。
この章では、**JavaScript（ジャバスクリプト）**を通して、
“Webが生きている”という感覚を体験します。
最初は「意味が分からない英語の羅列」に見えるかもしれません。
でも大丈夫。
JavaScriptは、人間の言葉に近い“会話の言語”。
ブラウザに「こうして」と命令するだけで、ページは動き出します。
最終的には、「クイズアプリ」を完成させます。
小さな仕組みを積み上げながら、動くものを作っていこう。

💡 STEP 41〜60｜JavaScript 超スモールアップ

⚡ STEP 41｜JavaScriptとは何かを調べる

Webに「動き」を与える魔法の言語。
ボタンが押せる、画像が切り替わる、クイズが動く——
それらはすべてJavaScriptの力。
💬 “見える”世界を作るのが*HTML*、
*“*感じる”世界を作るのが*JavaScript*。

🧩 STEP 42〜44｜最初の一歩：ブラウザと対話してみよう

1. script.js ファイルを作ってHTMLに読み込む
2. console.log("Hello, JavaScript!") を出してみる
3. alert("こんにちは！") でメッセージを表示

目に見える反応が返ってくると、心が動く。
これが「プログラミングの報酬」です。

💡 STEP 45〜49｜考える力をコードで鍛える

JavaScriptは、頭の中のロジックを「形」にできる。

- let / const … 変数を使って情報を保存
- if文 … 条件を分けて判断する
- for文 … 繰り返しを自動化する
- 配列（Array） … 複数のデータをまとめる
- 関数（function） … 命令のまとまりを作る

💬 人間が「どう考えるか」を、そのままコードにすればいい。

🔘 STEP 50〜53｜クリックでページが反応する

「クリックしたら変わる」
——これだけで、ページは一気に“生きた”感覚になる。

1. ボタンをHTMLに設置
2. document.getElementById() で取得
3. addEventListener("click", ...) で動作を設定
4. innerText で文字を変更

💡 たった3行のコードが、“静”を“動”に変える。

🎲 STEP 54｜ランダムでおみくじを作る

```
const result = ["大吉", "中吉", "小吉", "凶"];
const num = Math.floor(Math.random() * result.length);
alert(result[num]);
```

これが、あなたが作った“最初のアプリ”。
シンプルでも、ちゃんと動く。
自分の指示でページが変わる——その瞬間が面白い。

🧠 STEP 55〜59｜「クイズアプリ」を作ってみよう

小さな仕組みを積み上げて、完成を目指します。

STEP 55｜クイズ問題を配列で作る

```
const quiz = [
  { q: "HTMLのHは何？", a: "Hyper" },
  { q: "CSSで色を指定するプロパティは？", a: "color" },
];
```

STEP 56｜1問ずつ表示する

ボタンを押すたびに次の問題を表示。

STEP 57｜正解・不正解を判定する

入力値をチェックして、「正解！」を表示。

STEP 58｜スコアをカウントして表示する

結果が“数値化”されると達成感が生まれる。

STEP 59｜「もう一度遊ぶ」ボタンを追加

学びを“体験”に変える最後の仕上げ。

🧩 STEP 60｜完成！3問クイズアプリ

あなたが書いたコードで、クイズが動く。

これが“エンジニアになる”ということ。

完成したアプリを見て、こう思ってほしい。

「あ、ちゃんと動いてる」

──それが、最高のご褒美です。

✨ ここで得られるスキル

分野	できるようになること
JavaScript基礎	条件分岐・繰り返し・関数・配列を理解
DOM操作	HTML要素を動的に操作できる
論理思考	“どう動かすか”を順序立てて考えられる

💬 リュウセイから

「動いた！」という感動は、すべての原点。

エラーが出ても、止まってもいい。
それは「頭の中が成長しているサイン」だから。
コードを書いて、反応を見て、また書き直す。
この繰り返しが、確実に“創る力”になる。

📘 次章予告：**Chapter4｜PHPで“仕組み”を作る**
「ページを動かす」から、「データを扱う」へ。
あなたが作るWebが、“サービス”に進化していきます。

CHEPTER4

⚙️ Chapter 4｜PHPで“仕組み”を作る
——「動く」から、「動かす」へ。

💬 INTRODUCTION
JavaScriptで“ページを動かす”ことができたあなたへ。
次に学ぶのは、データを扱うサーバーサイドの思考です。
HTMLとCSSとJSは、いわば「見える世界」。
でもWebの裏側では、常に誰かが「受け取り」「処理し」「記録して」います。
その裏方を担うのが、**PHP**（ピー・エイチ・ピー）。
この章では、

- フォームから情報を送信
- PHPで受け取って保存
- そして画面に表示
  この流れを“ローカル環境”で体験します。

つまり、ここでやることは——
Webアプリを「作る人」になるための第一歩。

💡 STEP 61〜80｜PHP 超スモールアップ
🧩 STEP 61｜PHPとは何かを調べる
PHPは「Webの裏側」を支える言語。
ユーザーが送ったデータを受け取り、
“その人専用の結果”を返す力を持っています。
💡 *JavaScript*が前線なら、*PHP*は司令塔。

🖥️ STEP 62｜サーバーとは何かを理解する
Webサイトは「誰かのPCの中で動いている」。
そのPCこそが“サーバー”。
あなたのパソコンを小さなサーバーにして、
自分で動作を確かめるのが次のステップです。

⚙️ STEP 63〜65｜MAMPを使って“自分のサーバー”を作る

1. MAMPをインストール
2. 起動して「localhost」にアクセス
3. “It works!”が出たら準備完了
4. 「htdocs」フォルダを開いて、ここにファイルを置く

💬 「*localhost*」は、あなたの中にある“練習用の世界”。

🧱 STEP 66｜index.php を作る

PHPファイルは「.php」という拡張子を使います。
新しく **index.php** を作って、こう書いてみよう。

```
<?php echo "Hello, PHP!"; ?>
```

ブラウザで開いたら「Hello, PHP!」が出力されるはず。
これが、あなたの最初の“サーバーからの応答”です。

🔄 STEP 67〜70｜変数・条件・繰り返しを理解する

JavaScriptと似た考え方で動きます。

```
$name = "Ryusei";
if ($name == "Ryusei") {
  echo "ようこそ、TechFieldへ！";
}
```

PHPは「値を受け取り、判断して返す」言語。
つまり、“思考をロジックに変える”練習です。

💬 STEP 71〜74｜フォームからデータを送る

HTMLフォームを作り、PHPで受け取ってみよう。

```
<form method="POST" action="result.php">
  <input type="text" name="name">
  <button type="submit">送信</button>
</form>
// result.php
$name = $_POST["name"];
echo "こんにちは、" . $name . "さん！";
```

名前を入力して「こんにちは、〇〇さん！」と表示されたら大成功。
“人とプログラムが会話した瞬間”です。

📄 STEP 75〜78｜データを保存する仕組みを作る

情報を扱うには、「記録」が必要。
PHPではファイルにデータを書き込むことができます。

```
$file = fopen("data.txt", "a");
fwrite($file, $name . "\n");
fclose($file);
```

保存された文字が、あなたのアプリの記憶になります。
つまり、アプリに“命”が宿った瞬間です。

🗂️ STEP 79〜80｜読み込んで表示する

記録されたデータを一覧表示してみよう。

```
$lines = file("data.txt");
foreach ($lines as $line) {
  echo $line . "<br>";
}
```

保存 → 読み込み → 表示
この流れが、あらゆるWebサービスの基礎。
TwitterもYouTubeも、すべてこの仕組みの延長線上にある。

✨ ここで得られるスキル

分野	できるようになること
PHP基礎	変数・条件分岐・繰り返し構文の理解
入出力	フォームからデータを送信・保存・表示
思考力	「仕組みを作るとは何か」を体験的に理解

💬 リュウセイから

コードが動くのは、魔法じゃない。
そこには「意図」と「仕組み」がある。
あなたが作った小さな仕組みが、
いつか“誰かのために動く”日がくる。
PHPはそのための第一歩。
“仕組みで人を動かす”ことを、ここで体感してほしい。

📘 次章予告：**Chapter5**｜データベースと**CRUD**の基本操作

いよいよ、“本格的なWebアプリ開発”へ。
データを「登録・表示・編集・削除」する力が、ここで身につきます。

CHEPTER5

🧩 Chapter 5｜データベースとCRUDの基本操作

――「データが動く」とき、アプリは“命”を持つ。

💬 INTRODUCTION

あなたがこれまで学んできたHTML・CSS・JavaScript・PHP。
これらはすべて、「人とコンピュータをつなぐための表現」でした。
でも、今から扱うのは――
「記録」×「変化」×「再現」。
つまり、アプリが生きるための中枢：データベースです。

たとえばTwitterなら「ツイートを投稿」して「一覧で見る」。
メモアプリなら「メモを書く」「編集する」「消す」。
この一連の流れこそ、エンジニアが日常的に使う
CRUD（Create / Read / Update / Delete）です。
この章では、最もシンプルで実用的な題材として、
「メモアプリ」を軸に、CRUDの仕組みを丁寧に体験していきましょう。

🧭 CRUDとは？

操作	意味	例
C：Create	新しいデータを作る	メモを登録する
R：Read	データを読む	登録したメモを見る
U：Update	データを更新	内容を編集する
D：Delete	データを削除	不要なメモを消す

💡 この4つができれば、どんなアプリでも作れる。

🧠 データの流れ（テキスト図解）

[ユーザー入力]
↓
[PHPが受け取る]
↓
[DB（MySQL）が保存]
↓
[PHPが読み出す]
↓
[ブラウザで表示]

この流れを作れれば、
あなたはもう「Webアプリを構築できる人」です。

💡 STEP 81～115｜CRUDメモアプリ構築スモールアップ

⚙️ STEP 81〜84｜データベースを作ろう

1. MAMPを起動 →「phpMyAdmin」を開く
2. 新規DBを作成（名前は techfield_db）
3. テーブル memos を作成
    - id（INT, AUTO_INCREMENT, PRIMARY KEY）
    - content（TEXT）
    - created_at（DATETIME）

💬 テーブルとは、アプリの“記憶ノート”のようなもの。

🧱 STEP 85〜90｜PHPでDBと接続する

DBとの通信を行うために、接続スクリプトを作成します。

```
$conn = mysqli_connect("localhost", "root", "root", "techfield_db");
if (!$conn) {
  die("接続失敗: " . mysqli_connect_error());
}
echo "接続成功！";
```

エラーが出ても焦らずに。
“接続成功”が出たら、あなたのアプリとDBが“会話を始めた”瞬間です。

🧩 STEP 91〜95｜Create（登録）を作る

1. フォームでメモ内容を入力
2. PHPで受け取り
3. SQL文でINSERT

```
$content = $_POST["content"];
$sql = "INSERT INTO memos (content, created_at) VALUES ('$content', NOW())";
mysqli_query($conn, $sql);
```

「登録ボタン」＝**C**（**Create**）
たった1行のINSERTが、“アプリが何かを覚える”最初の行動です。

🧭 STEP 96〜100｜Read（一覧表示）を作る

データを取り出して、一覧にします。

```
$result = mysqli_query($conn, "SELECT * FROM memos ORDER BY id DESC");
while ($row = mysqli_fetch_assoc($result)) {
  echo $row["id"] . ": " . htmlspecialchars($row["content"]) . "<br>";
}
```

目の前のページに、自分が登録したメモが並ぶ。
それは、**“コードが世界を動かす感覚”**を実感できる瞬間です。

✍️ STEP 101〜105｜Update（編集）を作る

既存のメモを変更できるようにします。

1. 一覧に「編集」リンクを設置
2. 編集ページで内容を取得
3. フォームから新しい内容を送信
4. UPDATE文で書き換え

```
$sql = "UPDATE memos SET content='$content' WHERE id=$id";
mysqli_query($conn, $sql);
```

💡 エラーが出たら、それは「理解を深めるチャンス」。

🗑️ STEP 106〜110｜Delete（削除）を作る

最後に「削除」機能を追加。

```
$sql = "DELETE FROM memos WHERE id=$id";
mysqli_query($conn, $sql);
```

一見地味だけど、この一行がアプリを“完成”させる。
不要なものを消せる＝管理できる。
エンジニアは“作る人”であると同時に、“整える人”です。

🔁 STEP 111〜115｜CRUD全体を整理・改善

4つの機能を一連で動かし、流れを再確認します。

- 登録 → 一覧 → 編集 → 削除
- 全ての操作がデータベースに反映される
- 画面を更新すると結果が変わる

💡 この一連が理解できれば、*SNS*・ブログ・*EC*サイトも同じ構造。

🧩 データフローの全体像

(1) ユーザーがメモを入力

↓

(2) PHPがPOSTで受け取る

↓

(3) INSERT文でDBに登録

↓

(4) SELECT文で一覧を表示

↓

(5) 編集/削除ボタンから再びPHPへ

↓

(6) UPDATE / DELETE 文で変更

この流れを作れた時点で、
あなたの中に「Webアプリの地図」が描かれています。

✨ ここで得られるスキル

分野	できるようになること
PHP×MySQL	DB連携の基本（接続・登録・取得・更新・削除）
SQL	基本的なSQL構文の理解
設計力	データフローを考えて構築できる
問題解決力	エラーを読み解き、原因を特定する

💬 リュウセイから

CRUDは「退屈」と思う人もいる。
でも、実は“創造”のすべての基礎なんだ。
世界中のWebサービスは、CRUDでできている。
あなたのコードが、誰かの人生の記録を扱う日も、そう遠くない。

📘 次章予告：**Chapter6｜XserverでWeb公開とポートフォリオ化**
あなたのアプリを、世界に出そう。
コードが“誰かの目に届く”瞬間が、エンジニアの誇りです。

CHEPTER6

🌍 Chapter 6｜XserverでWeb公開とポートフォリオ化
——あなたのコードが、世界に届く瞬間。

💬 INTRODUCTION
「動くようになった」——でも、まだ終わりじゃない。
本当の意味でのゴールは、**“誰かが使えるようにすること”**です。
アプリが自分のPCの中だけで動いているうちは、まだ練習。
それを“世界に出す”ことで、あなたの学びは「自己満足」から「価値提供」に変わります。
サーバーにアップし、URLを共有すれば、
誰でもアクセスできる、あなたの作品が完成します。
さあ、いよいよ外の世界へ。
“あなたのコードが、世界とつながる瞬間”です。

💡 STEP 116〜150｜Web公開・ポートフォリオ構築ステップ
☁️ STEP 116〜120｜Xserverにログイン・準備
Xserverは、国内で最も安定したレンタルサーバーの一つ。
WordPressやPHPに対応しており、初心者でも簡単に公開ができます。

1. Xserverのアカウントにログイン
2. サーバー契約を確認
3. 管理パネルを開く
4. 「ドメイン設定」から独自ドメインを登録（例：techfield.dev）
5. 反映完了まで少し待つ

💡「自分のURL」が持てる瞬間、それは“デジタルの名刺”を手にしたようなもの。

🗂️ STEP 121〜130｜ファイルをアップロード

1. 「ファイルマネージャー」を開く
2. public_html フォルダを開く
3. portfolio フォルダを作成
4. ローカルで完成させたアプリ（CRUDメモなど）をアップロード
5. ブラウザで https://あなたのドメイン/portfolio にアクセス

もし画面にあなたのアプリが表示されたら——
それは“0から作ったものが世界に届いた”証拠。

🔒 STEP 126〜130｜SSL設定でセキュア化

公開したサイトを安全に保つために、SSL設定を行います。

1. 「SSL設定」を開く
2. 無料独自SSLを有効化
3. .htaccess にリダイレクト設定を追加

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
```

これでサイトは「https://」に対応。
あなたの作品が、より信頼されるWebサイトになります。

🧩 STEP 131〜135｜フォルダ構成と更新管理

アプリが増えてきたら、整理整頓が重要です。

```
portfolio/
├─ app/
├─ css/
├─ img/
├─ db/
└─ index.php
```

💡 整っているコードは、美しい文章と同じ。
「読まれる」ことを意識して整理しよう。

👩‍🎓 STEP 136〜140｜自己紹介ページを作ろう

エンジニアの価値は「何を作ったか」と「なぜ作ったか」で決まる。

1. index.html に自己紹介を書こう
2. これまで作ったアプリへのリンクを貼る
3. 技術一覧・制作期間・使用言語を明記する
4. スマホでも見やすいようCSSを調整
5. 最後に「学びのストーリー」を添える

「どんな想いで作ったのか」を言語化できる人は強い。
それはあなたの作品を、“単なるコード”から“物語”に変える。

📣 STEP 141〜145｜SNSで公開報告

あなたの挑戦は、必ず誰かの勇気になります。

- X（旧Twitter）に公開報告を投稿
- #ポートフォリオ公開 #駆け出しエンジニア などのタグを付ける
- 作品URLと簡単な説明を添える

💬「発信」は自己表現であり、次の出会いの入口。
あなたが発信したそのポストを見て、
「自分もやってみよう」と思う人が必ず現れます。

🗓️ STEP 146〜150｜継続と成長

Web公開はゴールではなく、スタートラインです。

1. 月に1回、ポートフォリオを更新
2. 改善・追加機能を計画
3. 新しいアプリを追加していく
4. GitHubに保存し、README.mdに詳細を記載
5. 「次に挑戦したいテーマ」を決める（例：API連携、ログイン機能など）

💡 継続するエンジニアは、“知識”ではなく“習慣”で成長する。

🧠 公開後のデータフロー（テキスト図解）

[あなたのアプリ]
↓
[Xserverのサーバー]
↓
[独自ドメイン]
↓
[ユーザーのブラウザ]
あなたのコードが、世界中のどこからでもアクセス可能に。
それは、努力が価値に変わった瞬間です。

✨ ここで得られるスキル

分野	できるようになること
Web公開	XserverでPHPアプリを公開できる
ドメイン	独自ドメイン取得とSSL設定の理解
発信	SNSでの自己ブランディング
継続	自分の成長を更新し続ける習慣化

💬 リュウセイから

コードを書くことは、未来を作ること。
あなたがここまで積み上げた力は、
“ただ学んだ”だけではなく、“形にした”ものです。
もうあなたは、“エンジニアを目指す人”ではない。
“自分の手で世界を動かす人”です。

🎓 **FINAL MESSAGE**｜ここからが、スタート。

150ステップを終えたあなたへ。
もしこのPDFを最後まで読み進めたなら、
それだけで“圧倒的に動ける人”です。
TechFieldでは、
あなたの行動を“成果”に変えるために、
一歩先のステップを一緒に設計していきます。
📩 無料相談 **or** 個別コーチングの案内
あなたの進捗・課題をもとに、
「本気で動きたい人のための設計図」を作ります。
🔗 TechFieldコーチ リュウセイ 公式LINEからご連絡ください。