

購入者特典

コードが書けなくても、アプリは作れる。
非エンジニアがClaude Codeで実際に1本作った話

- 特典1 最初に投げる依頼文のテンプレート
- 特典2 初心者がよく詰まるポイントのチェックリスト
- 特典3 困ったときの聞き方サンプル集

本記事のご購入、ありがとうございます。

この3点は、そのままコピーして使える実用ファイルです。

特典1 最初に投げる依頼文のテンプレート

Claude Codeに最初に話しかけるときの、この4点を埋めるだけで、返ってくるものの精度が大きく変わります。コピーして、【】の中を埋めて、そのまま貼り付けてください。

そのまま使えるテンプレート

【何を作りたいか】：
【誰が使うか】：
【最低限どう動けばいいか】：
【今回はどこまでやるか】：

上記の内容でアプリを作ってください。
分からない専門用語が出てきても、おすすめの形で進めてください。

記入例①：メモアプリ

【何を作りたいか】：日々のタスクを記録するメモアプリ
【誰が使うか】：自分1人だけが使う
【最低限どう動けばいいか】：メモを追加できる、消せる、それだけでいい
【今回はどこまでやるか】：まずは動くものが見たいので、見た目は後回しでいい

上記の内容でアプリを作ってください。
分からない専門用語が出てきても、おすすめの形で進めてください。

記入例②：家計簿アプリ

【何を作りたいか】：毎日の支出を記録して、月ごとの合計が見られる家計簿アプリ
【誰が使うか】：自分と配偶者の2人で使う
【最低限どう動けばいいか】：金額とカテゴリを入力して保存できる。月別の合計が見られる
【今回はどこまでやるか】：まずは入力と合計表示だけ。グラフ化は次の段階でいい

上記の内容でアプリを作ってください。
分からない専門用語が出てきても、おすすめの形で進めてください。

4点を埋めるときのコツ

項目	埋め方のコツ
何を作りたいか	一言で言えるレベルまで簡単にする（「便利なやつ」はNG）
誰が使うか	「自分だけ」か「他の人も」かだけでも十分

項目	埋め方のコツ
最低限どう動けばいいか	「これができれば合格」というラインを1〜2個に絞る
今回はどこまでやるか	見た目・機能追加など、後回しにしていいものを先に決めておく

この4点さえ埋まっていれば、題材が変わっても同じ型で応用できます。

特典2

初心者がよく詰まるポイントのチェックリスト

本記事・第4章の内容を、手元に置いておける1枚チェックリストにまとめました。依頼を出す前・作業中・完成後の3タイミングで、それぞれ確認してください。

依頼を出す前のチェック

- ☒ 「何を・誰が・最低限どう動けば・どこまでやるか」の4点を埋めたか（特典1のテンプレート参照）
- ☒ 一度に全部を頼もうとしていないか（ログイン・デザイン・通知など盛り込みすぎ注意）
- ☒ まず一番小さい動く形から頼もうとしているか

※ 曖昧な依頼で一番多い失敗は「いい感じのアプリを作って」だけで投げること。ここに使う3分が、後の30分の手戻りを防ぎます。

作業中のチェック

- ☒ Claude Codeからの作業報告を、読まずに次へ進めていないか
- ☒ 1つの作業が終わるたびに、実際に自分の目で動かして確認したか
- ☒ 分からない専門用語が出てきても、慌てずに「おすすめで進めて」と伝えられているか

※ 確認を飛ばして「次お願いします」と進めると、間違いに気づくのが数ステップ後になり、直すのがかえて面倒になります。地味でも、都度確認が一番の近道です。

完成後のチェック（エラーが出たとき）

- ☒ 赤いエラーメッセージを見て、固まっていないか
- ☒ エラーの中身を自分で理解しようとしすぎていないか
- ☒ 表示された文言をそのままコピーして「これが出た」と伝えられているか

※ エラーの原因分析はClaude Code側がやってくれます。怖がらず、そのまま貼って渡してください。

主なつまずき4パターン（早見表）

つまずき	避け方
最初の依頼が曖昧すぎる	4点（何を・誰が・最低限どう動けば・どこまでやるか）を先に埋める
一度に全部を頼もうとする	一番小さい動く形から作り、機能を1つずつ足す
確認せずに次へ進む	1作業ごとに実際に動かして確認する
エラーを見て固まる	文言をそのままコピーして「これが出た」と伝える

小さく作って確認し、1つずつ足していくやり方が、結局は一番速い近道です。

特典3 困ったときの聞き方サンプル集

「何て伝えればいいかわからない」となったときに、そのままコピーして使える言い回し集です。3つの場面別に、パターンをまとめました。

パターン①：エラーが出たとき

基本の伝え方

こんなエラーが出ました。直してください。

(ここにエラーメッセージをそのまま貼り付け)

状況が分かるとき用

〇〇をしようとしたら、次のエラーが出ました。原因を調べて直してください。

(ここにエラーメッセージをそのまま貼り付け)

繰り返し出るとき用

さっき直してもらった後、また同じようなエラーが出ました。
今度は根本的な原因まで調べて直してもらえますか。

(ここにエラーメッセージをそのまま貼り付け)

※ エラーメッセージは理解しようとしなくて大丈夫です。文言をそのまま貼るだけで十分です。

パターン②：進行に迷ったとき

現状を整理してほしいとき

ここまでの話を整理して、次に何をすればいいか教えてください。

進め方を任せたいとき

ここから先、どう進めるのがおすすめですか。おすすめの順番で進めてください。

一旦立ち止まりたいとき

今の状態で一旦止めて、ここまで何ができていて、何が残っているかを教えてください。

※ Claude Code自身に進行を仕切ってもらうことができます。無理に自分で判断しなくて大丈夫です。

パターン③：仕上がりが思っていたのと違うとき

見た目が違うとき

動きはよさそうですが、見た目が思っていたのと違います。
（ここに「こうなってほしい」を具体的に書く）に変更してください。

動きが違うとき

〇〇のボタンを押したときの動きが、想定と違います。
本来は（こうなってほしい内容）という動きにしてほしいです。

そもそも要件が曖昧だったと気づいたとき

最初の依頼が曖昧だったかもしれません。
あらためて条件を整理しますので、一度リセットして次の内容で作り直してもらえますか。

（ここに特典1のテンプレートで整理した4点を貼り付け）

※ 完璧な依頼文である必要はありません。「思っていたのと違う」と感じたら、そのまま素直に伝え直すのが一番早い直し方です。

最後まで読んでいただき、ありがとうございます。

この3つの特典を使って、ぜひ1本目のアプリを作り切ってください。