

# 01

CHAPTER

# LLM 時代の開発を 体験する

LLM と一緒に学ぶ Web アプリ開発 — ゼロからデプロイまで

— 本書のサンプル章 — 全文無料公開

# この章で学ぶこと

**01** LLM が開発にもたらした変化  
プログラミング学習はどう変わったか

**02** 本書で作るアプリの全体像  
テキストアドベンチャーゲームを作る

**03** 学習の進め方  
全 13 章・約 23 時間の旅

**04** 必要な前提知識（ほぼゼロで OK）  
初心者でも安心して始められる

PART 1

# 何が変わったのか

ChatGPT が変えたプログラミング学習の風景

---

# プログラミング学習の革命

2022.11

## ChatGPT

### 登場

プログラミングの世界が一変した瞬間。  
学習の壁が、対話の対象に変わった。

### 学習体験の変化

BEFORE

#### 数時間の格闘

Stack Overflow 検索、英語ドキュメントとの戦い、  
そして時には諦めてしまう。

AFTER

#### 数秒で解決策

エラーを LLM に貼って質問するだけ。  
24 時間 365 日付き添う優秀なメンター。

# 「挫折の道」から「成功体験の積み重ね」へ

## 従来の学習フロー

疑問発生から挫折まで

- 1 Google 検索 10 分
- 2 Stack Overflow で似た質問を探す 20 分
- 3 英語のドキュメントを読む 30 分
- 4 試行錯誤 1 時間
- 5 うまくいかず挫折...

## LLM 時代の学習フロー

疑問から成功体験まで

- 1 LLM に質問 30 秒
- 2 即座に回答が返ってくる
- 3 分からない部分を対話で深掘り 5 分
- 4 実装して動作確認 10 分
- 5 成功体験を積み重ねる！

# 開発者に求められるスキルの変化

## 以前重視されたスキル

暗記とパターン認識の時代

- すべての文法を暗記
- フレームワークの API を細かく記憶
- エラーメッセージのパターンを覚える
- 大量のドキュメントを読破

## 今重視されるスキル

設計と対話の時代

- 1 適切な質問をする力  
*プロンプト力*
- 2 LLM の回答を評価する力  
*正しいか判断*
- 3 本質的な設計判断  
*アーキテクチャ・セキュリティ*
- 4 問題を分解して段階的に解決する力  
*問題分解力*

# 得意なこと、苦手なこと

LLM は強力な相棒。でも万能ではない。最終判断は人間の役目。

## 得意なこと

What LLMs do well

- ✓ 定型的なコードの生成
- ✓ エラーメッセージの解説
- ✓ 既存のベストプラクティスの提示
- ✓ コードのリファクタリング提案

## 苦手なこと

Where humans need to step in

- ! 最新すぎる情報（学習データの締切以降）
- ! プロジェクト全体の文脈理解
- ! 最適なアーキテクチャの判断
- ! セキュリティの深い考慮

PART 2

# 本書で作るもの

AI と会話しながら進めるテキストアドベンチャー

---

# テキストアドベンチャーゲーム

## AI が物語を生成する対話型アドベンチャー

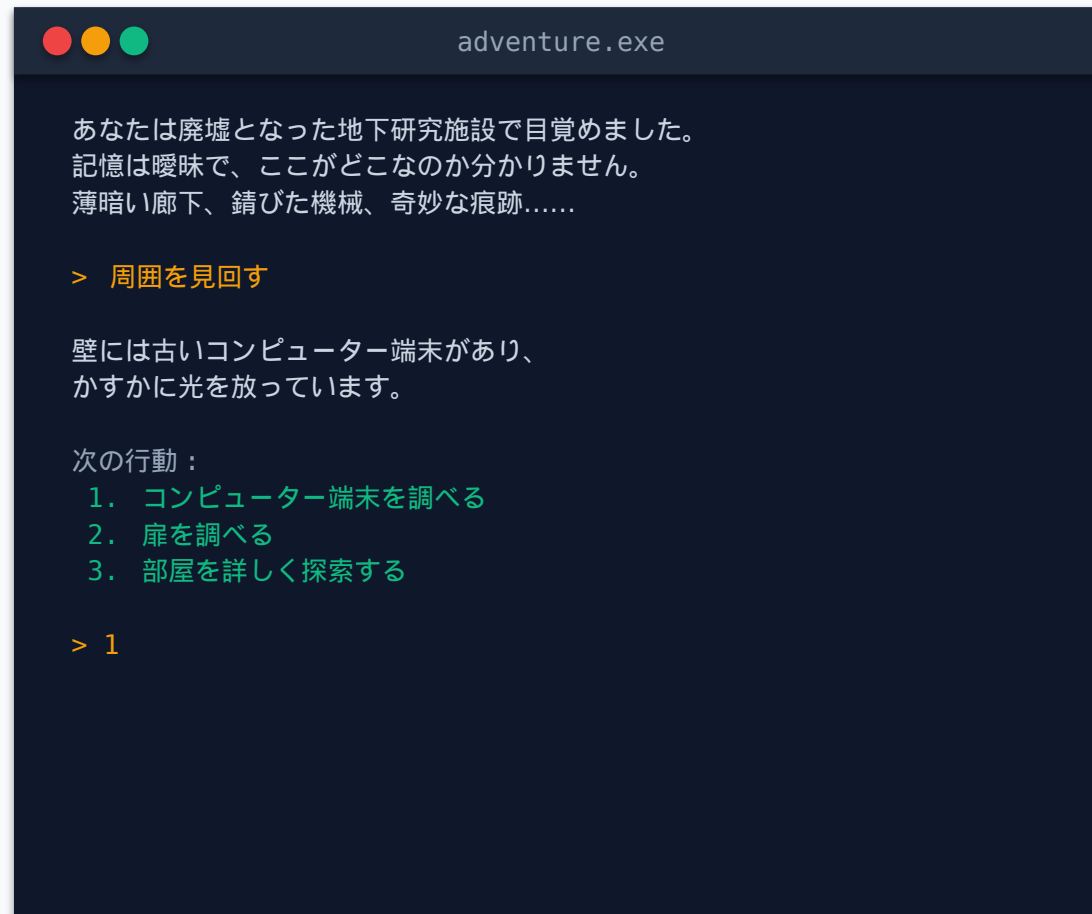
ストーリーは Gemini API によって生成され、  
プレイヤーの選択によって無限に分岐します。

本書のゴールは「動くサービスを公開する」こと。  
サンプルはシンプル、デプロイは手厚く。

完成形デモ

[adventure-dev-journey.com](https://adventure-dev-journey.com)

本書で作るアプリを実際に体験できます



# なぜこのアプリを選んだのか

## SIMPLICITY

### シンプル

- 画像アップロード不要
- DB は SQLite
- 1 台の EC2 で完結

## LEARNING

### 学べる

- LLM API 連携
- チャット UI 実装
- セッション管理
- AWS デプロイ・HTTPS 化

## ENJOYMENT

### 面白い

- 実際に遊べる
- 友達に見せられる
- シナリオ変更で別ゲームに
- 機能追加しやすい

# 本書で使用する技術

## BACKEND

### バックエンド

**Python 3.13+**  
プログラミング言語

**SQLite**  
データベース

**Django 5.2**  
Web フレームワーク

**Gunicorn**  
WSGI サーバー

## FRONTEND / LLM

### フロント・LLM

**htmx**  
動的 UI ( JS 不要 )

**Gemini API**  
物語生成エンジン

## DEPLOY

### デプロイ

**AWS EC2**  
サーバー

**Let's Encrypt**  
SSL 証明書

**Nginx**  
リバースプロキシ

## DEV TOOLS

### 開発ツール

**Git**  
バージョン管理

**Cursor**  
IDE

PART 3

# 学習の進め方

全 13 章・約 23 時間の旅をどう走るか

---

# ハンズオン形式で 4 つのフェーズ

PART 1

## 準備と企画

第 1 ～ 4 章

環境構築・LLM で企画を固める・Django 基礎

PART 2

## 開発

第 5 ～ 7 章

Gemini API 連携・チャット UI・ゲームロジック

PART 3

## デプロイと運用

第 8 ～ 12 章

AWS 準備・EC2 デプロイ・HTTPS 化・運用

PART 4

## 発展

第 13 章

機能拡張のアイデア・次のステップ

# 学習時間の目安

13

章

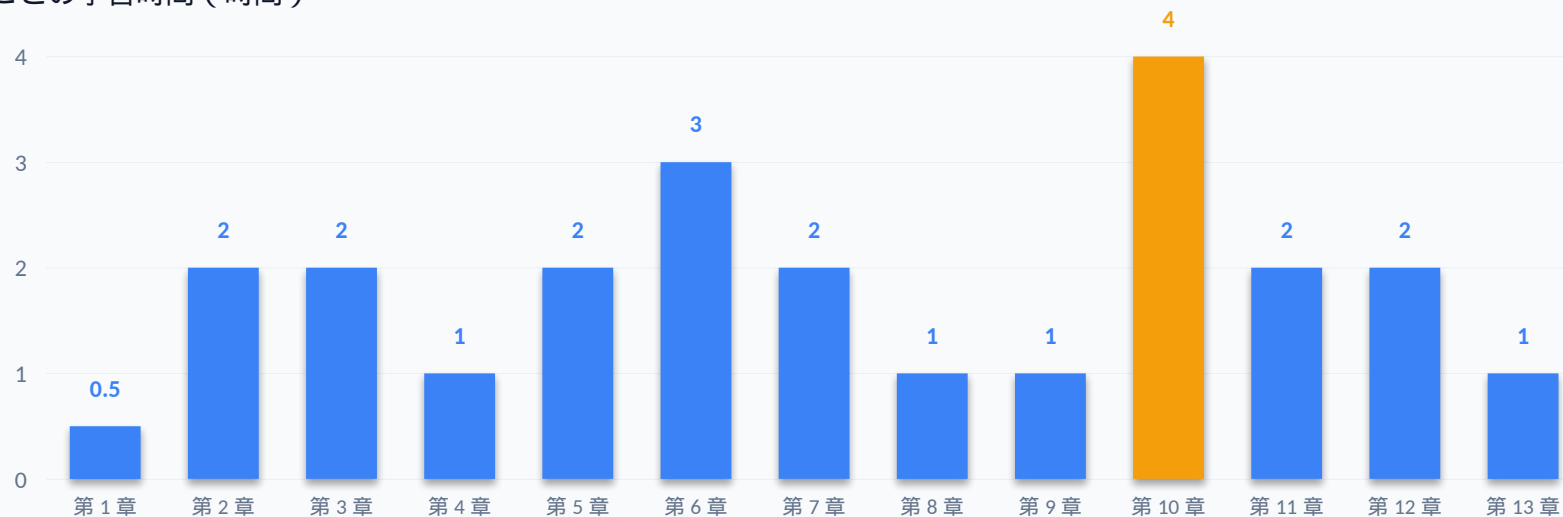
23h

合計学習時間

4h

最難関の第10章

章ごとの学習時間（時間）



おすすめペース

集中型

週末2日 × 3週間

平日型

平日2時間 × 12日

マイペース型

1日1章 × 約2週間

# 挫折しないための5つのコツ

01

必ず手を動かす

読むだけでなく、実際にコードを書く

02

エラーを恐れない

エラーは学習のチャンス

03

LLM に頼る

分からないことは即座に質問

04

完璧を目指さない

まず動かす、改善は後で

05

SNS で発信

学習記録のシェアでモチベーション維持

# 「良い質問」の構造

## 質問テンプレートの6要素

- |   |       |                         |
|---|-------|-------------------------|
| 1 | 背景・目的 | 何を作っていて、何がしたいのか         |
| 2 | 現状    | どこまで動いているか              |
| 3 | 問題    | エラーメッセージをそのまま貼り付け       |
| 4 | 試したこと | 何を試して、結果はどうだったか         |
| 5 | 環境    | OS / 言語 / フレームワークのバージョン |
| 6 | 質問    | 原因と解決方法を聞く              |

good-question.txt

**【背景・目的】**  
Django でチャット機能を実装中。

**【現状】**  
メッセージ送信までは動いている。

**【問題】**  
以下のエラーが出ました：  
TypeError: 'NoneType' object  
is not iterable

**【試したこと】**  
・ xx を試した → 結果は□□

**【環境】**  
macOS 14.2 / Python 3.13 / Django 5.2

**【質問】**  
原因と解決方法を教えてください。

# 前提知識ほぼゼロで OK ・ 本書で身につくスキル

## 必要な前提知識

(全部なくても大丈夫)

### LEVEL 1

これだけあれば十分

プログラミング経験 (言語問わず) / Web の基本概念

### LEVEL 2

あるとスムーズ

HTML/CSS の基礎 / 変数・関数・if 文

### LEVEL 3

あれば理解が早い

Python の基礎文法 / コマンドライン経験

## 本書で身につくスキル

完走後、あなたが手に入れるもの

### 技術スキル

Python 基礎

Django

LLM API 活用

HTML / CSS

Git

AWS

デプロイ実践

### 非技術スキル

プロンプトカ

エラーの読み方

問題分解力

ドキュメントの読み方

トラブルシューティング

READY?

# さあ、始めましょう。

プログラミングは、自転車に乗ることと同じ。  
最初は転ぶかもしれない。でも、転びながら学ぶのが一番の近道です。

本書の5つの約束

挫折ポイントを明示

実際のエラーを掲載

LLM との対話例

段階的な説明

動くものを作る

NEXT

## 第2章 — 開発環境を整える

Python ・ Git ・ エディタをインストールし、初めての LLM との対話を体験する。

コーヒーを用意して、深呼吸して、はじめましょう ☺