2026 EDITION

FREE

AI
教養の教科書

ChatGPTを毎日使うあなたが
本当に知るべきAIのすべて

LLM原理・プロンプトエンジニアリング・バイブコーディング
エージェント・マルチモーダル・セキュリティ・キャリア戦略

バイブコーディング テイラー
@shuntailor

22 Chapters・無料配布

TABLE OF CONTENTS

目次

著者のことば

最近、こんな言葉をよく耳にする。

> これからの就職市場はAIと人間の競争になる。AIに仕事を全部奪われる。

本当にそうだろうか？

私の考えは違う。これからはAIと人間の競争ではなく、AIをうまく使いこなす人とそうでない人の競争になる。正直に言えば、競争という表現すら適切ではない。AIを巧みに操る人に、そうでない人が勝つ方法はほぼない。

同じ時間に10倍の成果を出す人の隣で、旧来のやり方に固執するのは、自転車で高速道路に入るようなものである。速度の問題ではなく、次元の問題だ。

だから問いを変えなければならない。「AIに仕事を奪われるだろうか？」ではなく、「AIで自分の生産性をどう高めるか？」を問うこと。この問いの転換が本書の出発点である。

Elon MuskとJensen Huangは同じ方向を指し示してこう語る。「これからはPhysical AIの時代だ」と。

AIがソフトウェアの世界を超え、物理的な世界に拡張するという意味だ。ロボット、自動運転、スマートファクトリー——AIが現実世界で直接動き始める。そのAIを動かすにはデータセンターが必要で、データセンターには天文学的な電力がかかる。だから二人とも冗談半分、本気半分でこう付け加える。

> AI時代に最後まで生き残る職業？ 配管工と電気工だ。
>
> — Jensen Huang, NVIDIA CEO

笑い話だが、核心を突いている。AIがどれだけ進化しても、物理的インフラには人の手が必要だ。同時にこの発言は、ホワイトカラーの知識労働こそAIによって最も大きく再編される領域だという警告でもある。まさに今大学に通っているあなたが入っていく、その世界の話だ。

この本で私が伝えたいことはシンプルだ。

AIを恐れるな、理解せよ。理解した分だけ使いこなせるし、使いこなせる分だけ生き残れる。

本書はプログラミングの教科書ではない。AIがどのように動くのか、なぜ嘘をつくのか、どんな質問をすればよい答えが返ってくるのか、どこまで信じてよくてどこから疑うべきかを扱う。26のチャプターを読み終えたとき、あなたはAIを「ただ使う人」から「理解し設計する人」に変わっているだろう。

AI時代の教養とは、コーディング能力ではない。AIの構造を理解し、限界を知り、自分の判断で適切に活用できる能力のことだ。本書がその第一歩となることを願う。

一つ告白すると、本書のかなりの部分はAIの助けを借りて作成した。リサーチ、初稿の執筆、データ収集、PDF生成まで——AIにできるすべての工程でAIを活用した。しかし構造を設計し、何が重要かを判断し、どの順序で説明すれば読者が理解できるかを決めたのは人間である。本書自体が「AIを道具として使い、判断は人間がする」方式の実践事例だ。

2026年3月、ソウルにて

バイブコーディング テイラー (@shuntailor)

プロローグ：
あなたはAIを「使う」だけでいいのか？

今この瞬間、9億人がChatGPTを使っている

2026年2月時点、ChatGPTの週間アクティブユーザー数は9億人である。毎週、9億人。世界人口8人に1人の割合だ。

あなたもその一人である可能性が高い。課題の下書きをするとき、知らない単語を調べるとき、アイデアに詰まったときにChatGPTを開く。もしかすると本書を開く直前にもそうしていたかもしれない。

しかしちょっと考えてみよう。ChatGPTがなぜその答えを出すのか説明できるだろうか？その答えが正しいか間違っているか、どう判断しているだろうか？1年前のChatGPTと今のChatGPTがどこでどう変わったか知っているだろうか？

ほとんどの人は首を横に振る。それが普通であり、間違ってもいない。しかしその状態は「電卓の使い方は知っているが電卓の仕組みは知らない」のと同じだ。電卓ならそれでも十分だ。しかしAIは電卓とはまったく異なる代物である。

「使うこと」と「理解すること」の間の距離

スマートフォンが普及した2010年代、「デジタルリテラシー」という言葉が流行した。インターネットで情報を探せる、SNSでコミュニケーションできる——それが当時求められていた能力だった。今、同じことがAIで起きている。「AIリテラシー」という単語が求人情報に、授業の計画書に、ニュース記事にあふれている。

AIリテラシーとは何か。「ChatGPTを使えること」ではない。AIが何をしているのか、何が得意で何が苦手で、どこでどう間違えるのかを理解した上で適切に活用することである。その差は思った以上に大きい。

本書はあなたを怖がらせるために書いたのではない

AIに関する本には二種類ある。「AIで仕事がなくなる」と不安を煽るものと、「AIで何でもできる」と夢を売るもの。どちらも半分だけ正しい。

本書はそのどちらでもない。動作原理を理解してよりうまく使うための本である。大学生として社会に出る前の今、AIの本質を理解しておくことはこれからの時代を切り拓くための基盤となる。

プログラミング経験はまったく不要だ。数式も出てこない。難しい英語の専門用語も登場するたびに丁寧に説明する。文系でも理系でも、AIを日常的に使う人であれば誰でも読めるよう設計した。

> **本書が初めての読者へ**
>
> 本書は開発者向けの本ではない。「コードが読めない」「数学が苦手だ」という人ほどむしろ役に立つ。原理の説明はすべて日常の言葉とたとえで行う。難しいと感じたらその章を飛ばして次に進んでも構わない。

本書を読み終えると、あなたは何が変わるのか

本書を読み終えると、以下のことが可能になる。

- ChatGPTとClaudeが「なぜそう答えるのか」を概念レベルで説明できる
- ハルシネーションを見抜き、AI回答を適切に検証できる
- プロンプトの書き方を変えてAI出力の品質を大幅に高められる
- 2026年現在の主要AIモデルの違いと用途別の選択基準を理解する
- 「バイブコーディング」でプログラミング経験なしにアプリのプロトタイプを作れる
- AIエージェントとは何かを理解し、自分の学業・仕事にどう応用するか構想できる

これらはすべて今日から使える実用的な能力だ。自己紹介文の一行を埋めるための飾りではない。授業、課題、インターン、プロジェクトで今すぐ差を生む力である。

本書の構成

パート	タイトル	核心内容
1	AIの動作原理	機械学習・LLM・生成AI・ハルシネーション
2	AIの舞台裏	データセンター・GPU・ローカルLLM・6大企業ストーリー
3	2026年AI勢力図	モデル比較・ベンチマークの読み方
4	プロンプトエンジニアリング	基礎〜上級 10の技法・コンテキスト・ハーネス
5	バイブコーディング	Lovable・Claude Code・Genspark
6	AIエージェント	エージェント・マルチエージェント・企業事例
7	AI実践リテラシー	学業倫理・AI検索・マルチモーダル・セキュリティ
8	サバイバル戦略	キャリア戦略・ファクトチェックガイド

順番どおりに読む必要はない。興味のあるパートから始めてもよい。ただしパート1を先に読んでおくと、以降の章の理解度が大きく変わる。

読者タイプ別おすすめ読み順

1. 「AIの原理が気になる」→ パート1を最初に。4章のハルシネーションは特に重要。
2. 「今すぐAIをうまく使いたい」→
   パート3（プロンプトエンジニアリング）から開始。8章の10の技法は必読。

3. 「コードなしでアプリを作りたい」→ パート4 バイブコーディングへ直行。11章・12章から。
4. 「就職・キャリアにAIを活用したい」→
   パート2でAI業界の全体像を押さえ、パート7でAI時代のキャリア戦略を確認。
5. 「AI倫理とセキュリティが心配」→
   パート6へ直行。学業倫理、セキュリティ、バイアスの問題をすべて扱う。
6. 「最初から順番に全部読む」→
   最も深い理解につながる。1日1章ずつ24日で原理からサバイバル戦略まで完走。

2026年、AIは「使えるか使えないか」の段階を超えた。「どれだけ深く理解し活用するか」が求められる時代だ。週間9億人という数字が示すように、AIは今や空気のように当たり前の存在である。しかし空気の成分を知る人と知らない人は、酸素が不足した状況での判断力がまるで違う。

あなたはAIを「使う」だけでいいのか？

本書はその問いに対する一つの答えである。さあ、始めよう。

PART 1

AIの仕組みを理解する

01

機械学習・ディープラーニングとは何か？

Googleもこうして動いている

プログラミングと機械学習の根本的な違い

コンピュータに何かをさせるとき、従来のプログラミングでは人間がルールを直接書く。「メールに『無料』『今すぐ』という単語が含まれていたらスパムに分類せよ」——こうしたif文の集合がソフトウェアだった。

> 非開発者へ
>
> 機械学習は「コンピュータにルールを教える」のではなく、「データを見せてパターンを自分で見つけさせる」アプローチだ。子どもに猫の写真を何十枚も見せながら「これが猫だよ」と教えるのと同じ感覚である。

この方式には限界がある。スパム業者が「無料」を「無☒料」と書き始めたらルールを書き直さなければならない。猫の画像を認識させるには「耳が三角で、ヒゲがあって、毛がふわふわで…」というすべての条件を人間が言語で整理する必要があるが、それは事実上不可能だ。

機械学習は発想を逆転させた。ルールを書く代わりに、大量のデータと正解ラベルをコンピュータに渡す。コンピュータが自らパターンを見つけ、次の予測に使うモデルを作る。スパムフィルターなら数万通の「スパム」と「スパムではない」メールを見せれば、コンピュータが自ら判断基準を学習する。

区分	従来のプログラミング	機械学習
アプローチ	人間がルールを直接記述	コンピュータがデータからルールを学習
公式	データ＋ルール → 出力	データ＋出力 → ルール（モデル）
スパムフィルターの例	「バイアグラ」を含む → スパム処理	スパム10万件学習 → 自動判別
長所	ロジックが明確、デバッグが容易	複雑なパターンも自動発見
限界	ルールが数万個になると管理不能	なぜそう判断したか説明が難しい

機械学習の3つの種類

機械学習は学習方式によって大きく3種類に分かれる。違いを知れば、ChatGPT、Spotifyのレコメンドアルゴリズム、ゲームAIがそれぞれどの方式を使っているか見えてくる。

種類	学習データ	代表的な活用
教師あり学習 (Supervised Learning)	入力と正解ラベルのペア	スパムフィルター、画像分類、医療画像診断
教師なし学習 (Unsupervised Learning)	ラベルなしのデータのみ	顧客セグメント分析、異常検知、トピック抽出
強化学習 (Reinforcement Learning)	環境との試行錯誤	ゲームAI（囲碁・チェス）、ロボット制御、広告入札

教師あり学習が最も広く使われている。正解ラベル付きのデータを大量に用意して「この入力のときはこの出力が正解」というパターンを学ぶ。例えば猫の写真10万枚と「猫」/「猫ではない」ラベルを与えれば、モデルは新しい写真が猫かどうか判断できるようになる。

教師なし学習はラベルなしで構造を見つける。顧客100万人の購買データを渡すと「似た行動パターンのグループ」を自動で発見する。マーケターが直感では気づけなかった顧客群が浮かび上がる場合もある。

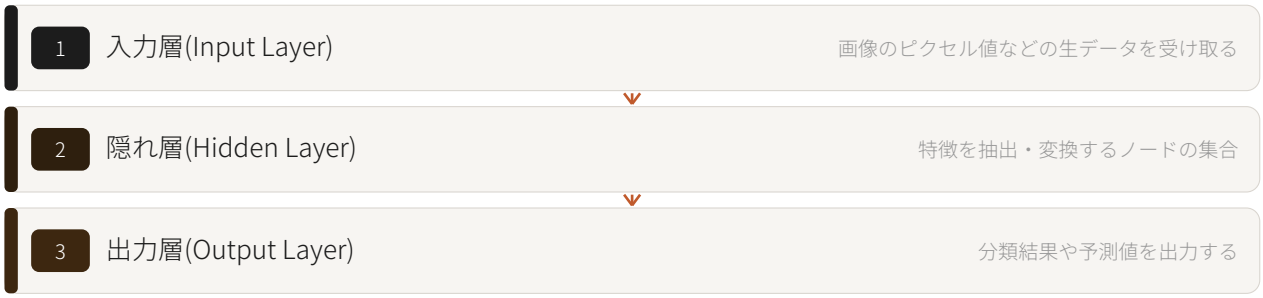
強化学習はゲームのルールだけ教えて無限に試行錯誤させる方式だ。2016年にAlphaGoが世界チャンピオンのイ・セドルに勝利したのも、数百万局の自己対局による強化学習の成果だった。

ニューラルネットワークとは何か

機械学習の中でも「ニューラルネットワーク（Neural Network）」という手法が現代AIの中心にある。名前だけ聞くと難しく感じるが、発想自体はシンプルだ。

人間の脳には約860億個のニューロン（神経細胞）がある。各ニューロンは複数の信号を受け取り、一定の閾値を超えると次のニューロンに信号を送る。この「受け取る→判断する→送る」という単純な動作の組み合わせが思考、感情、記憶を生み出す。

人工ニューラルネットワークはこの構造を数式で模倣する。各「ノード」は複数の数値（入力）を受け取り、それぞれに重み（weight）を掛けて足し合わせ、活性化関数というフィルターを通過させて次のノードに渡す。この重みが「学習」によって調整されるパラメータだ。



学習方式を「誤差逆伝播（Backpropagation）」と呼ぶ。モデルが出した答えと正解を比較し、誤差が小さくなるよう重みを少しずつ調整する。これを数万回繰り返しながらモデルの精度が向上する。

ディープラーニング ― 層を深く積むと何が起こるか

ディープラーニング（Deep Learning）はニューラルネットワークの隠れ層を数十層・数百層に積み上げたものだ。「ディープ（深い）」という名前はこの多層構造に由来する。

層を深くする理由は情報の抽象化にある。画像認識を例にすると、第1層は「エッジ（輪郭線）」を検出し、第2層は「形状（円・四角）」を組み合わせ、第3層は「部位（耳・目・鼻）」を認識し、最終層で「これは猫だ」という判断を下す。人間が「猫の特徴」を明示的に教えなくても、モデルが自ら特徴を発見していく。

ディープラーニングが一気に注目されたのは2012年のImageNetコンテストだ。トロント大学チームが提出したAlexNetはそれまでの最高精度を約10ポイント上回り、研究者たちを驚かせた。その後、画像認識・音声認識・自然言語処理のほぼすべての分野でディープラーニングが従来手法を置き換えた。

- 画像認識：スマートフォンの顔認証、医療画像での癌検出
- 自然言語処理：翻訳、テキスト生成、感情分析
- 音声認識：Siri、Googleアシスタントの音声→テキスト変換
- レコメンドシステム：YouTubeの次の動画、Amazonの「この商品はいかがですか？」
- ゲームAI：囲碁・チェス・StarCraftで人間世界チャンピオンを超越

1章まとめ

機械学習は「ルールを書く」のではなく「データからパターンを学ぶ」アプローチである。教師あり学習・教師なし学習・強化学習の3種類がある。ニューラルネットワークは脳のニューロン構造を数式で模倣したモデルであり、層を深く積んだディープラーニングが現代AIの中核技術となった。画像認識・音声認識・言語処理まで、私たちが日常で使うほぼすべてのAI機能の背後にディープラーニングがある。

02

LLMの原理

AIはなぜ「もっともらしい」文章を書けるのか？

LLM（大規模言語モデル）とは何か

ChatGPT、Claude、Gemini——これらはすべて「Large Language Model（大規模言語モデル）」、略してLLMという技術の上に作られている。LLMとはインターネットの膨大なテキストデータを学習したディープラーニングモデルで、文章の続きを予測することに特化している。

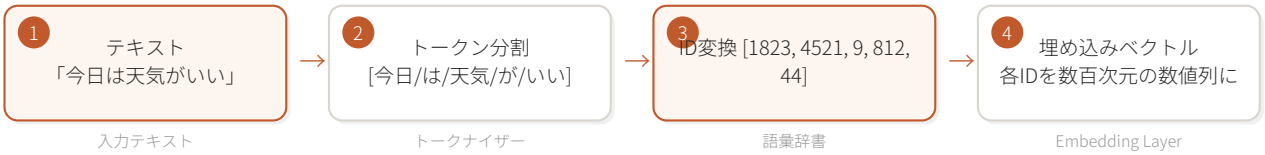
「大規模」という修飾語には二つの意味がある。一つはモデルのパラメータ数——GPT-4は数千億から1兆を超えるパラメータを持つとされる。もう一つは学習データの規模——数千億から数兆語分のテキストが学習に使われた。

> 非開発者へ
>
> パラメータとはモデルの「重み」の数だ。人間の脳のシナプス結合の強度に相当する。パラメータが多いほど細かなパターンを記憶できる。GPT-3の1,750億パラメータを1秒に1個ずつ書き出していったら全部書き終えるのに5,500年以上かかる規模だ。

テキストはどうやって数値になるか ― トークナイゼーション

コンピュータは文字をそのまま処理できない。まず「トークナイゼーション（Tokenization、トークン化）」という過程でテキストを数値列に変換する必要がある。

トークンは文字でも単語でもなく、その中間のような単位だ。英語で「unhappy」は「un」と「happy」の2つのトークンに分割される。日本語では「機械学習」が「機械」と「学習」のように形態素単位で分かれる場合が多い。OpenAIのGPT-4では日本語1文字は1〜2トークン程度になる。



各トークンIDは「埋め込み（Embedding）」という処理で高次元ベクトルに変換される。意味が近い単語ほどベクトル空間での距離も近く学習される。例えば「王」から「男」を引いて「女」を足すと「女王」に近いベクトルになる——という有名な実験がこれを示している。

日本語はなぜトークンを多く消費するのか

トークンの効率は言語によって異なる。ほとんどのトークナイザー（BPEアルゴリズム）は英語中心のデータで学習されている。英語は約4文字が1トークンに相当するが、日本語は1〜2文字で1トークンが必要だ。同じ意味を伝える文章でも日本語は英語より1.5〜2倍多くのトークンを消費する。

文章	言語	トークン数（概算）
Hello, how are you?	英語	約6トークン
こんにちは、お元気ですか？	日本語	約12〜15トークン
The weather is nice today.	英語	約6トークン
今日は天気がいいです。	日本語	約10〜12トークン

これはコストに直結する。AIサービスのAPI料金は「トークン数」を基準に課金される。同じ内容でも日本語ユーザーは英語ユーザーより約1.5〜2倍多くの料金を支払うことになる。大学の課題でAIを多用するほど、この差を実感するだろう。

AIに質問するとき実際に送られるもの

ChatGPTやClaudeに質問を投げると、ユーザーが入力したテキストだけがAIに送られるわけではない。目に見えない要素が一緒に送信される。AIが毎回のリクエストで実際に受け取る入力は3つの層で構成されている。

1 システムプロンプト（AIの行動ルール・人格設定） 毎リクエストごとに繰り返し送信

2 過去の会話履歴（これまでの全質問＋全回答） ターンが積み重なるほど急激に増加

3 現在のユーザー入力（今送ったメッセージ） 実際にユーザーが書いた部分

システムプロンプトは「あなたは親切なAIアシスタントです」のような指示文で、ユーザーには見えないが毎回のリクエストと一緒に送信される。ChatGPTのシステムプロンプトは数千トークンに達すると言われている。

重要なのは「過去の会話履歴」だ。AIは以前の会話を「記憶」しているのではない。毎回、過去のすべての質問とすべての回答を丸ごと再入力される。会話1ターン目はユーザー入力だけが送信されるが、10ターン目には前の9ターン全体（質問9個＋回答9個）が入力トークンに含まれる。

会話が長くなるほどトークンコストは急増する

これはコスト構造を完全に変える事実だ。1ターンに平均500トークンを使うと仮定すると、10ターンの会話の総入力トークンは約27,500トークンになる（1ターン目500＋2ターン目1,000＋…＋10ターン目5,000）。単純に10倍ではなく約55倍のコストがかかるのだ。

会話ターン	該当ターンの入力トークン	累積総トークン	コスト感覚（GPT-5.4基準）
1ターン	500	500	約$0.001
5ターン	2,500	7,500	約$0.019
10ターン	5,000	27,500	約$0.069
20ターン	10,000	105,000	約$0.263
50ターン	25,000	637,500	約$1.594

大学生向け実践ティップ

長い会話を続けるより「新しい会話を始めて、必要な文脈だけ貼り付ける」方式がコストと品質の両面で有利だ。会話が長くなるとAIが「前半を忘れる」現象も発生する——これがまさに次に説明する「コンテキストウィンドウ超過」問題だ。

コンテキストウィンドウを超えると何が起こるか

すべてのLLMには「コンテキストウィンドウ」という処理上限がある。システムプロンプト＋過去の会話履歴＋現在の入力＋AI応答を合わせた総トークン数がこの上限を超えると問題が発生する。

APIを直接使う場合、「This model's maximum context length is X tokens」というエラーメッセージが返され、リクエスト自体が失敗する。一方、ChatGPTやClaudeのようなWebアプリではより静かな方法で処理される：古い会話内容が自動的にカットされる。

この自動カットはFIFO（先入先出）方式で動作する。新しいトークンが追加されるたびに最も古いトークンが押し出される。1時間前の会話内容がひっそりと消え、AIはその会話が存在したという事実自体を知らなくなる。
さらに深刻な場合、システムプロンプトまでカットされるとAIが設定された役割やルールを忘れ、突然別の口調で答えたり、精度が急激に低下する現象が発生する。これはハルシネーションを引き起こす主要な原因の一つだ。

自動カットの兆候

会話の途中でAIが突然「何のことかわかりません」と言ったり、以前合意した内容を完全に無視したりしたら、コンテキストウィンドウが超過した可能性がある。その場合は新しい会話を始めて、核心の文脈だけ再度伝えるのが最も効果的だ。

100万トークンだと論文何本分？

100万トークンはおよそA4で750ページ、論文にすると約30〜50本分に相当する。理論的には論文のPDFをまるごとアップロードしても構わない。しかし実戦では注意点がある。コンテキストが長くなるほどAIの「集中力」が落ちる。100万トークンを全部使うより、核心部分だけ抽出してアップロードする方が回答品質は高い。「論文全体をアップして要約して」より「この論文の3章と結論部分を分析して」の方がはるかによい結果になる。

コンテキスト超過に対応する最新技術

2026年現在、この問題に対応する技術が発展している。Claudeの「コンテキスト圧縮（Compaction）」は会話が上限に近づくと古い内容を自動要約して60〜70%の容量を確保する。要約された内容は完全ではないが、単純なカットよりはるかに多くの文脈を保持する。
また「プロンプトキャッシング」技術はシステムプロンプトのように毎回繰り返される部分をサーバーにキャッシュしてコストを最大90%まで削減する。しかし根本的な解決ではなく、トークン構造を理解して効率的に会話することが依然として重要だ。

TransformerとSelf-Attention — AIが文脈を読む方法

2017年にGoogleが発表した論文「Attention Is All You Need」で提案されたTransformer（トランスフォーマー）アーキテクチャが現代LLMの基盤だ。最大の革新は「Self-Attention（自己注意）」だった。

Self-Attentionとは、文中の各トークンが他のすべてのトークンとの「関連度」を計算する仕組みだ。「彼女はバッグを拾った。それが彼女のものだったから」という文で「それ」が「バッグ」を指していると理解するには文中で離れた単語同士の関係を把握する必要がある。Self-Attentionはこれを並列に処理して解決する。

Transformerは複数の「アテンションヘッド」を同時に走らせる。あるヘッドは文法的な依存関係に注目し、別のヘッドは意味的な関連性に注目する形で多角的な視点から文脈を分析する。GPT-4は96個のアテンションヘッドを持つとされている。

アーキテクチャ	特徴	主な用途
RNN（旧型）	順番に処理、長距離依存に弱い	2017年以前の翻訳・音声認識
Transformer（エンコーダー）	文全体を同時に分析	文分類、質疑応答（BERT系）
Transformer（デコーダー）	左から右へ順次生成	テキスト生成（GPT系）
エンコーダー-デコーダー	両方の組み合わせ	翻訳、要約（T5系）