



プロンプトエンジニアリング基礎

～AIと対話する技術の基本～

このスライドでは生成AIを使いこなすための「指示（プロンプト）」
の考え方・実践例を解説します。



| プロンプトエンジニアリングとは？

AIへ適切に指示を出して、望む出力を得る技術です。

- AIに質問する力＝AIを使いこなす力
- 生成AI時代の新たなスキルとして注目されています
- 言語を操る技術が、AIの能力を最大限に引き出します



良いプロンプト→良い結果
的確な指示がAIパフォーマンスの鍵です



なぜ重要なのか？

同じAIでも、指示の仕方で結果が大きく変わります

- 曖昧な指示 → バラバラな答え
- 明確な指示 → 期待通りの答え
- 作業効率や成果物の質が大幅に向上します

✕ 曖昧な指示の例

"何か良いアイデアを出して"

→ 目的や範囲が不明瞭で活用しにくい結果に

✓ 明確な指示の例

"小学生向けの環境教育イベントのアイデアを3つ提案して"

→ 具体的で実用的な提案が得られる



| プロンプト作成の基本

AIは「人間の意図を推測」して動きます

- 誰に向けた内容か（対象）
- 何をしてほしいか（目的）
- どんな形で出力するか（形式）

この3つを明確にすることが良いプロンプトの基本です



明確な指示 = 明確な結果

AIが意図を正確に理解できるように指示しましょう

| 良いプロンプトの条件

効果的なプロンプトを作るための4つのポイント

- **具体的**であること（あいまいな言葉を避ける）
- **前提や背景**を伝える
- **出力形式**を指定する（箇条書き、表、コードなど）
- 必要に応じて**例を提示**する



良いプロンプトは**明確な意図**と**具体的な指示**で構成されています

AIが人間の意図を正確に理解するための「翻訳作業」だと考えましょう

悪いプロンプト例

“ 「プログラミングについて教えて」

なぜ悪い例なのか：

- × 何を？どの言語？どのレベル？が不明確
- × 出力形式も指定なし
- × AIが推測しなければならない部分が多すぎる



AIに適切な情報を与えることで、より具体的で役立つ回答が得られます

良いプロンプト例

“「Python初心者向けに、変数の使い方を3つの例とコード付きで説明して」

なぜ良い例なのか：

- ✓ 言語：Python
- ✓ 対象：初心者
- ✓ 内容：変数の使い方
- ✓ 形式：3つの例＋コード



具体的な要素を指定することで、AIは期待通りの回答を提供できます



| プロンプト活用の場面

プロンプトエンジニアリングは様々な場面で効果を発揮します

 **資料作成**（企画書、提案書、レポート）

 **学習支援**（問題集作成、要約、解説）

 **コード生成**や修正、デバッグ支援

 **文章の翻訳**や文法・表現の添削

 **アイデア出し**（キャッチコピー、企画立案）

 **データ分析**や解釈のサポート

「同じAIでも**指示の質**で**出力の質**が変わります」



プロンプト改善の流れ

繰り返し改善することで、AIからより良い結果を引き出せます

- 1 まず初回指示を出す
- 2 出力結果を確認、修正点を見つける
- 3 指示や条件を追加・具体化して再実行
- 4 必要に応じて繰り返し改善！



継続的な改善サイクル

最初は完璧でなくても、対話を通じて洗練されていきます



| 上達のコツ

プロンプトスキルは練習と経験で着実に向上します。

- 短く簡単な指示でまず試す
- 結果を見て徐々に追加・調整する
- 他者のプロンプト例を参考にする
- 自分用のパターンをストックする



試行錯誤と改善のサイクル
繰り返しの実践が上達への近道です

まとめ

- プロンプトエンジニアリングはAI活用の必須スキル
- 明確・具体・構造化が成功のカギ
- 練習と改善でAIを自分の思い通りに動かせる



効果的なプロンプトで、AIの可能性を最大限に引き出しましょう