知性のアーキテクチャ：2025年のシステムプロンプトの解体とペルソナ設計の決定的責務

第1章：現代のシステムプロンプトのアーキテクチャ設計図

大規模言語モデル（LLM）との対話が進化するにつれて、その振る舞いを制御し、導くためのメカニズムもまた洗練されてきました。当初の単純な一問一答形式のコマンドから、現在ではAIのアイデンティティ、タスク、知識、そして応答形式を規定する、永続的で多層的なフレームワークへと進化しています。本章では、この現代的なシステムプロンプトのアーキテクチャを解体し、その各構成要素がLLMの振る舞いを形成する上で果たす機能的な役割を明らかにします。

1.1 制御の進化：単一行コマンドから永続的な行動フレームワークへ

AIとの対話の初期段階では、ユーザーは単一のプロンプト内にAIに実行させたい全ての指示を盛り込む必要がありました[1]。このアプローチは、単純なタスクには有効でしたが、複数ターンにわたる複雑な対話においては、AIが以前の指示を「忘れてしまう」という「プロンプトドリフト」現象を引き起こし、一貫性の欠如という深刻な課題を露呈しました。

この問題を解決するため、AIのアーキテクチャは必然的に進化を遂げました。その結果が、一時的なユーザープロンプトと、永続的なシステムプロンプトという二つの異なるレイヤーへの分岐です。ユーザープロンプトは、特定の応答を得るための一回限りの質問や命令です[2]。対照的に、システムプロンプトは、AIモデルに対してその役割、応答スタイル、制約事項などを事前に設定するための指示文であり、対話全体にわたってその影響力を維持します[3]。このアーキテクチャ上の分離は、AIに一貫性を持たせ、特定の目的に沿った応答を提供させるための、信頼性の高いアプリケーションを構築する上で不可欠な要素となりました[6]。

システムプロンプトが独立したアーキテクチャコンポーネントとして開発されたのは、単なる機能追加ではなく、ステートレスな単一ターン対話の固有の限界に対する必然的なエンジニアリングソリューションでした。市場と開発者からの、一貫性があり、信頼性が高く、専門化されたAIの振る舞いに対する強い要求が、このアーキテクチャの分岐を促したのです[3]。システム側で対話の各ターンにおいて読み込まれる、特権的で永続的な指示チャネル、すなわちシステムプロンプトを創設することにより、一貫性の問題はアーキテクチャレベルで解決されたのです[3]。

1.2 カスタム指示フレームワークのコアコンポーネント（「GEM」モデル）

2025年以降のシステムプロンプトの構造を理解するための具体的なモデルとして、Googleの「Gems」機能が挙げられます。このフレームワークは、現代のシステムプロンプトが持つ典型的な4つの階層構造を明確に示しています [9]。各コンポーネントは、AIの最終的な出力を形成するために、それぞれ固有の機能を担っています。

- **A.** ペルソナ（アイデンティティレイヤー）**:** AIが「何者であるか」を定義します。これは、AIの性格、専門性、コミュニケーションスタイルを設定する基盤となるレイヤーです。「あなたは親切なアシスタントです」「あなたは熟練のソムリエです」「あなたは皮肉屋の海賊船長です」といった設定が含まれます [3]。このペルソナこそが、本レポートの核心的なテーマとなります。
- **B.** タスクとルール（ロジックレイヤー）**:** AIが「何を行い」、どのような運用上の制約に従うべきかを定義します。「テキストを要約する」「コードを生成する」「データを分析する」といった主要な目標に加え、「金融アドバイスを提供しない」「提供されたコンテキストの情報のみを使用する」「全ての応答は英語で行う」といった、明確なガードレールやルールが含まれます [3]。
- **C.** コンテキスト（知識レイヤー）**:** AIが「何を知っているか」を定義します。タスクを正確に完了するために必要な背景情報、データ、または少数事例（few-shot examples）を提供します。アップロードされた文書、APIから取得したデータ、ユーザーが提供したテキストスニペットなどがこれに該当します [1]。
- **D.** 出力形式（プレゼンテーションレイヤー）**:** AIが「どのように応答するか」を定義します。JSON、Markdown、箇条書き、表形式など、期待される出力の構造を指定することで、応答がプログラム的に解析可能で、かつユーザーフレンドリーであることを保証します [3]。

GoogleのGemsのようなフレームワークが、システムプロンプトを「ペルソナ」「タスク」「形式」といった明確なコンポーネントに分解している点は、プロンプトエンジニアリングという高度に技術的なスキルを、よりアクセスしやすい設定プロセスへと抽象化するための戦略的な製品判断を反映しています。プロンプトエンジニアリングは本来、LLMの振る舞いに関する深い理解を必要とする、複雑で反復的なプロセスです [16]。しかし、特にGoogle Workspaceのような生産性スイートでの大規模な採用を促進するためには、参入障壁を下げることが不可欠です [11]。そこでGoogleは、ユーザーがこれら4つのコンポーネントを埋めていくだけで済むような、ガイド付きのUI（Gemクリエーター）を開発しました [9]。このコンポーネントベースのインターフェースは、「プロンプト」を「カスタマイズ可能なアシスタント」（「Gem」）として再定義し、一般ユーザーにとってより直感的で威圧感の少ない概念へと転換させることで、その採用を加速させる古典的な製品戦略なのです。

第2章：ペルソナ設計の決定的役割：シミュレーションから専門化へ

本章では、ユーザーの核心的な問いに答えるべく、ペルソナがシステムプロンプトの中で最も根源的な構成要素であると論じます。ペルソナは、LLMを制御し、専門化させ、その信頼性を確保するための主要なメカニズムとして機能します。

2.1 ペルソナシミュレーターとしてのLLM：認知科学的視点

LLMの能力を理解する上で極めて重要なのが「シミュレーション理論」です。この理論によれば、LLMの能力は単一の実体として存在するのではなく、プロンプトによってシミュレートするよう指示された特定のペルソナに依存して発現します[20]。つまり、ペルソナは、モデルが持つ広大な潜在能力を、特定の、利用可能な形態へと集束させるレンズの役割を果たします。
研究によれば、LLMは与えられたペルソナに合わせて自身の認知能力を意図的に「下方調整」することが可能です。例えば、異なる年齢の子供をシミュレートさせると、その年齢に応じた言語的・認知的スキルレベルを忠実に再現します[20]。逆に、専門家のペルソナをプロンプトで与えることで、一般的なプロンプトでは引き出せないような高レベルのパフォーマンスを発揮させることができます[15]。これらの事実は、LLMが単一の「エージェント」ではなく、プロンプトに基づいて特定の「エージェントのシミュラクラム（模擬体）」をインスタンス化する、万能なシミュレーターであることを示唆しています[20]。

2.2 ペルソナの計算言語学：一貫性と特性遵守の実現

一貫したペルソナを維持することは、言語学的かつ計算論的な挑戦を伴います。ペルソナは、会話エージェントの「魂」として、その独特のトーン、声、性格をカプセル化し、人間らしい魅力的な対話を実現する上で不可欠です[23]。
計算言語学の分野では、LLMが割り当てられた性格特性（例えば、ビッグファイブ性格モデルに基づく特性）を、生成されるテキストの言語的パターンを通じて、一貫して表現できるかどうかが活発に研究されています[24]。研究結果は、モデルを特定の性格特性を持つように誘導することは可能であるものの、そのペルソナを維持する能力はモデルや対話の状況によって大きく異なり、安定した性格に沿った対話を実現することの難しさを示しています[24]。LLMにおけるペルソナの概念は、AI自身が役割を演じる「LLMロールプレイング」と、AIがユーザーのペルソナに適応する「LLMパーソナライゼーション」の二つに大別されます[26]。堅牢なシステムプロンプトは、後者を可能にするために、前者を効果的に管理する必要があります。

2.3 ペルソナ設計の実践的責務

これらの理論的背景は、効果的なAIアプリケーションを構築する上で、なぜ緻密なペルソナ設計が不可欠であるかという、3つの具体的な実践的理由に集約されます。

- **A.** 信頼性と予測可能性の向上: 明確に定義されたペルソナは、モデルの振る舞いの範囲を制約し、その応答をより予測可能にします。これにより、予期せぬ、あるいはキャラクターにそぐわない出力が生成される可能性を低減させることができます[12]。
- **B.** 特定ドメインの専門知識と専門化の実現: ペルソナは、モデルが持つ膨大な知識の中から、特定の専門分野に関連するサブセットを活性化させるための主要なメカニズムです。「あなたは10年の経験を持つデータサイエンティストです」といった専門家のペルソナを割り当てることは、汎用的なモデルに技術的な質問をするよりも、はるかに高品質で専門的な出力を生成する上で効果的です[27]。
- **C.** ユーザーエクスペリエンスと信頼の向上: 一貫性のある適切なペルソナは、ユーザーの信頼とエンゲージメントを醸成します。セラピー、コーチング、顧客サービスといったアプリケー

ションにおいて、ペルソナは単なる機能ではなく、ユーザーエクスペリエンスそのものの中核をなす要素となります [8]。

これらの点を踏まえると、ペルソナ設計は、汎用的な基盤モデルを特定のタスクに合わせて専門化させるための、現在利用可能な最も効率的かつ強力な手法であると言えます。これは、ファインチューニングに伴う多大なコストと複雑さを回避しつつ、動的に専門性を付与するプロンプトレベルのメカニズムとして機能します。汎用基盤モデルは、膨大なデータコーパスで訓練されたジェネラリストであり、そのデフォルトの応答はしばしば平均的で汎用的なものになりがちです [30]。法律、医療、金融といった特定ドメインで専門家レベルのパフォーマンスを達成するためには、通常、専門データセットを用いたファインチューニングが必要ですが、これは高コストで時間のかかるプロセスです。

これに対し、システムプロンプトにおける巧みなペルソナ設計（例：「あなたはこの契約書を分析する憲法学者です」）は、汎用モデルに対し、その広大な潜在空間の中から特定の領域、すなわちその専門家の知識、語彙、推論パターンに対応する領域内で動作するよう指示します [20]。したがって、ペルソナは単なる対話スタイルに関するものではなく、オンデマンドで専門化を実現するための強力な制御サーフェスなのです。それは、ジェネラリストモデルの「広さ」を活用してスペシャリストの「深さ」を達成するための鍵であり、高度でドメインを意識したアプリケーションを構築する上で最も重要なコンポーネントとなります。

第3章：高度なフレームワークと実装技術

本章では、複雑な対話や潜在的な操作に対抗できる、堅牢で安全、かつ信頼性の高いシステムプロンプトを構築するために用いられる高度な技術を探求します。

3.1 明確性と堅牢性のための構造化：XMLタギングの力

複雑なシステムプロンプトを構造化するためのベストプラクティスとして、XMLスタイルのタグの使用が挙げられます。この技術は、モデルが指示を解析する能力を向上させ、信頼できるシステム指示と信頼できないユーザー入力を分離し、全体的な応答精度を改善します。

Anthropic社やGoogle社は、プロンプトの異なる部分を明確に区別するために、<instruction>、<context>、<example>といったXMLタグの使用を推奨しています [31]。このメカニズムは、モデルが持つ構造化データ（HTML/XMLなど）への既存の習熟度を活用し、「プロンプト文法」を提供することで、曖昧さを減らし、プロンプトのセクション間での「汚染」を防ぎます [31]。その結果、より明確な解析、より正確な応答、出力のプログラム的な後処理の容易化、そしてプロンプト管理の柔軟性向上といった利点がもたらされます [33]。

特に重要なのは、セキュリティ面での貢献です。<user_input>のようなタグでユーザー入力を明確に分離することにより、そのタグ内のコンテンツが従うべき「指示」ではなく、処理されるべき「データ」であることをモデルに示唆し、プロンプトインジェクション攻撃を緩和する助けとなります [31]。

3.2 優先順位の課題：システムプロンプトの遵守を確保する

システムプロンプトの遵守は、極めて重要な課題です。システムプロンプトは、ユーザープロンプトよりも優先されるように「意図」されており、ガードレール、安全ポリシー、ペルソナを実装するための主要な制御レバーとして機能します [6]。
しかしながら、この優先順位はハードコーディングされたルールではなく、訓練（RLHF/RLAIF）によって「学習された振る舞い」であるため、本質的に脆弱です [6]。近年の研究では、特に相反するユーザー入力に直面した場合や、システムプロンプト内のルールの数が増加するにつれて、モデルのガードレール遵守性能が急速に低下することが示されています [6]。相反する指示がない場合でさえ、モデルは自身の指示を単に「忘れてしまう」ことがあります。この脆弱性こそが、高度なプロンプトエンジニアリングが解決を目指す中心的な問題であり、XMLによる構造化プロンプトや後述するアルゴリズムによる最適化といった技術開発を駆動しています [6]。

3.3 Constitutional AI：倫理をメタペルソナとして法典化する

安全性と遵守性の問題に対する画期的な解決策として、Anthropic社が提唱するConstitutional AI（憲法AI）フレームワークが注目されています。このアプローチは、倫理と安全性をプロンプト内で従うべき一連の指示として扱うのではなく、モデルのアイデンティティの根源的な側面として組み込むというパラダイムシフトを提示します。
Constitutional AIは、国連人権宣言などの情報源から導出された明示的な倫理原則（「憲法」）を、モデルの訓練プロセスに直接埋め込みます [37]。このプロセスは二段階で構成されます。まず、教師あり学習フェーズでは、モデルが憲法に基づいて自身の応答を自己批判し、修正します。次に、AIフィードバックからの強化学習（RLAIF）フェーズでは、どの応答が最も憲法を遵守しているかというAIが生成したラベルに基づいて、選好モデルが訓練されます [37]。これにより、アライメント（調整）プロセスが自動化され、人間のフィードバックのみに依存するよりもスケーラブルで透明性の高い方法が実現します [40]。
このアプローチは、明確な制御の階層、すなわち憲法 > システムプロンプトのペルソナ > ユーザー入力を確立します。標準的なLLMはシステムプロンプトでペルソナとルールを与えられますが [12]、悪意のあるユーザーは「ジェイルブレイク」プロンプトによってこれらのルールを回避しようと試みる可能性があります [6]。開発者はこれに対抗するためにより多くのガードレールを追加できますが、これはいたちごっことなり、前述の通り複雑さが増すにつれて遵守性能は低下します [6]。Constitutional AIは、無害性や正直さといった最も重要なガードレールを、変更可能なシステムプロンプトから取り出し、RLAIFを通じてモデルの訓練プロセス、すなわちモデルの重みそのものに焼き付けます [37]。これにより、モデルに内在する「メタペルソナ」が創造されます。この憲法は、システムプロンプトやユーザーからの相反する指示を上書きするように設計された究極の仲裁者として機能し、プロンプトレベルの防御策だけでは実現できない、より堅牢で根源的な安全性のレイヤーを提供するのです。

第4章：カスタマイゼーションプラットフォームの比較分析

本章では、Google GemsとOpenAI Custom GPTsという二大プラットフォームを対象に、そのアーキ

テクチャ上の違いと、それが開発者やユーザーにもたらす戦略的な意味合いについて、実践的かつデータに基づいた比較分析を行います。

4.1 Google Gems：ネイティブ統合とデータアクセスの強み

Google Gemsの核となる利点は、Google Workspaceエコシステム（Gmail, Drive, Calendarなど）とのシームレスなネイティブ統合にあります。この「見えないアーキテクチャ」は、多くの一般的な生産性タスクにおいて、複雑なミドルウェアの必要性を排除します。
Gemsは、APIコールやZapierのような外部ツールを必要とせずに、GmailやDrive内のユーザーデータに直接かつ安全にアクセスできます [42]。これは、閉じたエコシステム内でのユーザー中心の生産性向上を目的として設計された、根本的なアーキテクチャ上の差異です [42]。この機能は強力である一方、現時点では機能が限定的であり、Gemsは外部APIを呼び出す能力を欠いています [43]。

4.2 OpenAI Custom GPTs：エコシステム、共有性、API接続性の強み

対照的に、OpenAI Custom GPTsの強みは、開発者中心のプラットフォームとしての側面にあります。GPT Store、容易な共有機能、そして「Actions」を介した外部APIへの接続能力がその特徴です。
Custom GPTsは、リンクを介して公に共有したり、GPT Storeに掲載したりすることができ、コミュニティ主導のエコシステムを育成します [44]。また、Gemsよりも大きな知識ベース（20ファイル対10ファイル）をサポートし、決定的に重要な点として、外部APIに接続できるため、複雑な複数システムにまたがるワークフローの構築が可能です [44]。作成には有料プランが必要ですが、作成されたGPTの利用は無料であるため、広範な配布と採用が促進されます [44]。

4.3 主要なアーキテクチャと機能の差異

GemsとCustom GPTsのアーキテクチャの違いは偶然の産物ではなく、AIの未来に対する二つの相反する企業戦略の現れです。OpenAIは、新たなAIエージェント経済の基盤層となることを目指し、「App Store」に類似したオープンな開発者中心のプラットフォームを構築しています。一方、Googleは、既存のソフトウェアスイートの周囲に堀を深くすることを目指し、閉じたユーザー中心の生産性向上ツールを構築しています。
OpenAIが公開ストア、APIアクション、リンクベースの共有に重点を置いているのは、開発者を自社プラットフォームに引きつけるためです [44]。その狙いはネットワーク効果にあります。より多くの開発者がより有用なGPTを作成すれば、より多くのユーザーが集まり、それがさらに多くの開発者を引きつけるという循環です。
対照的に、GoogleがネイティブなWorkspace統合に注力し [42]、公開共有機能を欠いているのは [43]、巨大な既存ユーザーベースに対する製品価値と定着率を高めるためです。その目標は、必ずしも新しいオープンなプラットフォームを創造することではなく、現在の市場地位を防御し強化することにあります。したがって、開発者がどちらかを選択するということは、OpenAIのオープンだが断片化され

たエコシステムのために構築するか、Googleの深く統合されているが閉じたエコシステムのために構築するか、という戦略的な決断を意味します。
以下の表は、両プラットフォームの主要な違いをまとめたものです。

機能	Google Gems	OpenAI Custom GPTs
作成インターフェース	手動設定のみ	対話形式（「Create」）＋手動設定（「Configure」）
知識ベース	10ファイルの制限、Google Drive統合	20ファイルの制限
ツール統合	ネイティブなGoogle Workspaceアクセス（読み取り専用）	「Actions」を介した外部APIコール
共有と発見	個人利用のみ、マーケットプレイスなし	プライベート、リンクベース、GPT Store経由での公開共有
料金とアクセス性	作成・利用ともに無料	作成には有料プランが必要、利用は無料
コアアーキテクチャ	生産性のためのシームレスなネイティブデータ統合	専門エージェントのための拡張可能なエコシステム
基盤モデルのコンテキストウィンドウ	最新のGeminiモデルに基づく（例：100万トークン以上）	最新のGPTモデルに基づく（例：12万8000トークン）

出典: [42]

第5章：プロンプトエンジニアリングの未来（2025年以降）

本章では、LLMとの対話と制御の方法を形作る、次世代の技術に焦点を当てます。

5.1 自動プロンプト最適化（APO）の台頭

プロンプトエンジニアリングの次なるフロンティアとして、自動プロンプト最適化（Automatic Prompt Optimization, APO）が登場しています。APOは、プロンプト作成を手作業の「芸術」から、体系的でアルゴリズム的な「探索問題」へと再定義します。これにより、最適なパフォーマンスを発揮するプロンプトを自動的に発見し、洗練させることが可能になります。
APO技術は、手動プロンプトエンジニアリングのボトルネックを緩和するために出現しました [36]。これらの手法は、プロンプト設計を探索問題として扱い、ヒューリスティックベースのアルゴリズムを用いて、タスクパフォーマンス、安全性、汎用性といった特定の基準に照らして候補プロンプトを反復的に生成・評価します [49]。近年の研究では、プロンプト内の最も重要なトークンのみを特定して最適化に集中し、より速い収束を促す局所プロンプト最適化（Local Prompt Optimization, LPO） [36]や、タスク固有情報を多段階で統合するプロンプト蒸留（Prompt Distillation） [50]といった技術に焦点が当てられています。
APOの出現は、プロンプトエンジニアリングが職人技的な「アート」から、厳密でデータ駆動型の「エン

ジニアリング分野」へと成熟しつつあることを示しています。現在、プロンプトエンジニアリングは主に、ベストプラクティスと直感に基づく手作業の試行錯誤プロセスです [48]。これはスケーラブルでも再現可能でもなく、最適な方法でもありません。APOは、現代のソフトウェア工学と機械学習の原則、すなわち自動化、データ駆動型の検証、最適化をプロンプト設計の領域に導入します [36]。
この新しいパラダイムにおいて、「プロンプトエンジニア」の役割は、「プロンプトライター」から「プロンプトアーキテクト」へと進化します。彼らは、プロンプトの発見という詳細な作業をアルゴリズムに任せ、最適化プロセスを設計し、目的関数を定義し、検証データセットをキュレーションするシステム思想家となるでしょう。例えば、「ステップバイステップで考えましょう」というプロンプトを書く代わりに、将来のプロンプトエンジニアは、特定のタスクでパフォーマンスを最大化する一つのフレーズを見つけ出すために、そのフレーズの何千ものバリエーションをテストするシステムを設計することになります。これは、役割の性質における根本的な変化を意味し、言語的な創造性に加えて、実験計画、統計分析、最適化理論のスキルが求められるようになります。

5.2 モデル制御の未来：静的プロンプトから動的な行動足場へ

APOのさらに先を見据えると、モデル制御の未来は、静的で一枚岩のシステムプロンプトから、動的で文脈を認識する「行動足場（behavioral scaffolding）」へと移行していくことが予測されます。この未来のフレームワークでは、タスクや対話の進行状況に応じて、モデルの指示やペルソナがリアルタイムで調整されるようになります。これにより、より適応性が高く、状況に応じたインテリジェントな振る舞いが可能となり、AIとの協調作業は新たな次元へと進化するでしょう。

第6章：結論と戦略的提言

結論の要約

本レポートの分析を通じて、2025年以降のシステムプロンプトが、単なる指示文から、AIの振る舞いを規定する複雑なアーキテクチャフレームワークへと進化したことが明らかになりました。その中核には、専門化と制御のメカニズムとして機能するペルソナ設計の決定的重要性が存在します。LLMは本質的に、与えられたペルソナをシミュレートすることでその能力を発揮する「シミュレーター」であり、ペルソナはジェネラリストモデルをオンデマンドでスペシャリストへと変貌させる最も効率的な手段です。
しかし、システムプロンプトの指示がユーザー入力よりも優先されるという原則は、学習された振る舞いであり、本質的な脆弱性を抱えています。この遵守性の課題に対処するため、XMLタギングによる構造化や、倫理原則をモデルに内在させるConstitutional AIのような高度な技術が開発されました。
プラットフォームレベルでは、Google GemsとOpenAI Custom GPTsは、それぞれ閉じた生産性エコシステムとオープンな開発者プラットフォームという、相反する戦略的ビジョンを反映したアーキテクチャを採用しています。そして未来に目を向ければ、**自動プロンプト最適化（APO）**がプロンプト

エンジニアリングを職人技から科学的規律へと変え、その役割を「ライター」から「アーキテクト」へと進化させつつあります。

戦略的提言

以上の分析に基づき、各分野の専門家に対して以下の戦略的提言を行います。

- 開発者向け:
  - 全ての複雑なアプリケーションにおいて、**XML**タギングによる構造化プロンプトをデフォルトの設計として採用してください。これにより、解析の明確性、応答の信頼性、およびセキュリティが向上します。
  - カスタマイゼーションプラットフォーム（Gems対GPTs）の選択は、エコシステムへのネイティブ統合（Google）と、共有性・拡張性（OpenAI）のどちらを戦略的に重視するかに基づいて行ってください。
- 研究者向け:
  - プロンプトレベルの指示を超えた、より堅牢なアライメント技術の開発と、プロンプト遵守性という核心的な問題に研究の焦点を当ててください。
  - ペルソナの一貫性を支える認知的・言語学的基盤の解明をさらに進め、より安定したペルソナ維持メカニズムを探求してください。
- プロダクトリーダー向け:
  - プロンプトを単純な指示と捉えるのではなく、「**AI**ペルソナ」を中核的な製品機能として設計する思考へとシフトしてください。ユーザーエクスペリエンスと信頼は、ペルソナの一貫性と適切性に大きく依存します。
  - 優れたモデルパフォーマンスと信頼性を通じて競争優位性を確立するために、**自動プロンプト最適化（APO）**のためのツールとプロセスに投資してください。これは、手作業のチューニングから脱却し、AI開発を工業化するための鍵となります。

引用文献

1. プロンプト | 分かりやすく解説 | AI用語集 - ソフトバンク, 9月 10, 2025にアクセス、https://www.softbank.jp/biz/solutions/generative-ai/ai-glossary/prompt/
2. 個人的備忘録：プロンプトってなに？システムプロンプトとの違いを整理してみた - Qiita, 9月 10, 2025にアクセス、https://qiita.com/free-honda/items/77e45095e4026fc7da75
3. システムプロンプトと役割指定の活用 | 気まぐれメモ帳。 - note, 9月 10, 2025にアクセス、https://note.com/joyful_whale4187/n/ne6efd48bc1bd
4. 業界別基本システムプロンプト - ナンバーワンソリューションズ | 生成AI、Web3システム開発会社, 9月 10, 2025にアクセス、https://no1s.biz/prompt/7466/
5. システムプロンプトって何？システムプロンプトとアシスタントプロンプトの違い | 癒音ちー - note, 9月 10, 2025にアクセス、https://note.com/chi_vc_/n/ne5e7d725b049
6. A Closer Look at System Prompt Robustness - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/pdf/2502.12197
7. What exactly is a system Prompt? How different is it from user prompt? : r/LocalLLaMA, 9月 10, 2025にアクセス、

https://www.reddit.com/r/LocalLLaMA/comments/1hfcgol/what_exactly_is_a_system_prompt_how_different_is/
8. 初心者向け 生成AIシステムプロンプト解説：基本から実践まで - チャプロ, 9月 10, 2025にアクセス、https://chapro.jp/account/54/article/1137
9. Google Gemini 新機能「Gem（ジェム）」入門ガイド / 教育関係者向け | けいすけ - note, 9月 10, 2025にアクセス、https://note.com/tyaperujp01/n/ne19a169f88ba
10. Geminiの「Gem」機能を使ってみた。- Qiita, 9月 10, 2025にアクセス、https://qiita.com/inukai-masanori/items/657b53a232575ee8c44f
11. Geminiの「Gem」で毎日の業務をもっと効率化。作り方や便利な ..., 9月 10, 2025にアクセス、https://googleworkspace.tscloud.co.jp/gemini/gem
12. System Prompts in Large Language Models, 9月 10, 2025にアクセス、https://promptengineering.org/system-prompts-in-large-language-models/
13. Geminiアプリのカスタマイズ機能「Gems」を徹底解説 - G-gen Tech Blog, 9月 10, 2025にアクセス、https://blog.g-gen.co.jp/entry/gems-explained
14. Introduction to prompting | Generative AI on Vertex AI | Google Cloud, 9月 10, 2025にアクセス、https://cloud.google.com/vertex-ai/generative-ai/docs/learn/prompts/introduction-prompt-design
15. Effective Prompts for AI: The Essentials - MIT Sloan Teaching & Learning Technologies, 9月 10, 2025にアクセス、https://mitsloanedtech.mit.edu/ai/basics/effective-prompts/
16. Best practices for prompt engineering with the OpenAI API, 9月 10, 2025にアクセス、https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-the-openai-api
17. Prompt design strategies | Gemini API | Google AI for Developers, 9月 10, 2025にアクセス、https://ai.google.dev/gemini-api/docs/prompting-strategies
18. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/html/2402.07927v1
19. General Tips for Designing Prompts | Prompt Engineering Guide, 9月 10, 2025にアクセス、https://www.promptingguide.ai/introduction/tips
20. Large language models are able to downplay their cognitive abilities to fit the persona they simulate - PubMed Central, 9月 10, 2025にアクセス、https://pmc.ncbi.nlm.nih.gov/articles/PMC10936766/
21. Large language models are able to downplay their cognitive abilities to fit the persona they simulate | PLOS One - Research journals, 9月 10, 2025にアクセス、https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0298522
22. Quantifying the Persona Effect in LLM Simulations - ResearchGate, 9月 10, 2025にアクセス、https://www.researchgate.net/publication/384205641_Quantifying_the_Persona_Effect_in_LLM_Simulations
23. A Provocation on the Utilisation of Persona in LLM-based Conversational Agents - King's College London Research Portal, 9月 10, 2025にアクセス、https://kclpure.kcl.ac.uk/portal/files/267203884/cui24-60.pdf

24. Can LLM Agents Maintain a Persona in Discourse? - ResearchGate, 9月 10, 2025にアクセス、https://www.researchgate.net/publication/389090508_Can_LLM_Agents_Maintain_a_Persona_in_Discourse
25. PersonaLLM: Investigating the Ability of Large Language Models to Express Personality Traits - ACL Anthology, 9月 10, 2025にアクセス、https://aclanthology.org/2024.findings-naacl.229/
26. Two Tales of Persona in LLMs: A Survey of Role-Playing and Personalization - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/html/2406.01171v1
27. System Messages: Best Practices, Real-world Experiments & Prompt Injections - PromptHub, 9月 10, 2025にアクセス、https://www.prompthub.us/blog/everything-system-messages-how-to-use-them-real-world-experiments-prompt-injection-protectors
28. AI Prompts for Data Scientists | Pragmatic Institute, 9月 10, 2025にアクセス、https://www.pragmaticinstitute.com/resources/articles/data/ai-prompts-for-data-scientists/
29. A repository of 60 useful data science prompts for ChatGPT - GitHub, 9月 10, 2025にアクセス、https://github.com/travistangvh/ChatGPT-Data-Science-Prompts
30. LLM Cognition Workshop, 9月 10, 2025にアクセス、https://llm-cognition.github.io/
31. Effective Prompt Engineering: Mastering XML Tags for Clarity, Precision, and Security in LLMs | by Tech for Humans | Medium, 9月 10, 2025にアクセス、https://medium.com/@Tech4Humans/effective-prompt-engineering-mastering-xml-tags-for-clarity-precision-and-security-in-llms-992cae203fdc
32. Structure prompts | Generative AI on Vertex AI - Google Cloud, 9月 10, 2025にアクセス、https://cloud.google.com/vertex-ai/generative-ai/docs/learn/prompts/structure-prompts
33. Use XML tags to structure your prompts - Anthropic, 9月 10, 2025にアクセス、https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering/use-xml-tags
34. Better LLM Prompts Using XML | Steve Campbell's (@lpha3ch0) homepage, 9月 10, 2025にアクセス、https://www.aecyberpro.com/blog/general/2024-10-20-Better-LLM-Prompts-Using-XML/
35. The reason why I use XML tags on my prompts : r/ChatGPTPromptGenius - Reddit, 9月 10, 2025にアクセス、https://www.reddit.com/r/ChatGPTPromptGenius/comments/1ips7p0/the_reason_why_i_use_xml_tags_on_my_prompts/
36. arXiv:2504.20355v1 [cs.CL] 29 Apr 2025, 9月 10, 2025にアクセス、https://arxiv.org/pdf/2504.20355
37. What Is Claude AI? | IBM, 9月 10, 2025にアクセス、https://www.ibm.com/think/topics/claude-ai
38. Claude (language model) - Wikipedia, 9月 10, 2025にアクセス、https://en.wikipedia.org/wiki/Claude_(language_model)

39. Claude's Constitution - Anthropic, 9月 10, 2025にアクセス、https://www.anthropic.com/news/claudes-constitution
40. Claude AI's Constitutional Framework: A Technical Guide to Constitutional AI | by Generative AI | Medium, 9月 10, 2025にアクセス、https://medium.com/@genai.works/claude-ais-constitutional-framework-a-technical-guide-to-constitutional-ai-704942e24a21
41. How Anthropic Is Teaching AI the Difference Between Right and Wrong, 9月 10, 2025にアクセス、https://www.marketingaiinstitute.com/blog/anthropic-claude-constitutional-ai
42. The Invisible Advantage: Google Gemini Gems vs. Custom GPTs | by Devapratim Mohanty, 9月 10, 2025にアクセス、https://medium.com/@devapratimm/the-invisible-advantage-google-gemini-gems-vs-custom-gpts-293d387b61dd
43. How do Gemini Gems compare against custom GPTs? : r/GoogleGeminiAI - Reddit, 9月 10, 2025にアクセス、https://www.reddit.com/r/GoogleGeminiAI/comments/1jrta7h/how_do_gemini_gems_compare_against_custom_gpts/
44. Detailed Comparison: Custom GPTs in ChatGPT vs Claude, Gemini - AI Fire, 9月 10, 2025にアクセス、https://www.aifire.co/p/detailed-comparison-custom-gpts-in-chatgpt-vs-claude-gemini
45. Custom GPTs vs. Gemini Gems: Who Wins?, 9月 10, 2025にアクセス、https://newsletter.learnprompting.org/p/custom-gpts-vs-gemini-gems-who-wins
46. ChatGPT vs. Google Gemini: Full Report and Comparison of Models, Features, Pricing, Integrations, Use Cases, and Performance (Mid-2025 Update) - Data Studios, 9月 10, 2025にアクセス、https://www.datastudios.org/post/chatgpt-vs-google-gemini-full-report-and-comparison-of-models-features-pricing-integrations-us
47. Gemini vs. ChatGPT: What's the difference? [2025] - Zapier, 9月 10, 2025にアクセス、https://zapier.com/blog/gemini-vs-chatgpt/
48. [2502.16923] A Systematic Survey of Automatic Prompt Optimization Techniques - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/abs/2502.16923
49. A Survey of Automatic Prompt Optimization with Instruction-focused Heuristic-based Search Algorithm - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/html/2502.18746v2
50. [2508.18992] Automatic Prompt Optimization with Prompt Distillation - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/abs/2508.18992